Judging Details

Judge System

Your programs will be judged on the system once it’s submitted.

- Your program must read input data from the standard input, and write its output to the standard output.
- Other outputs, e.g. writing to the standard error, will not be used for judging.
- You will never have to write to (open) a file, and are not allowed to do so.

Your programs will be run inside a *sandboxed environment*, i.e. with protections to prevent the system from being damaged. Specifically:

- Memory usage is limited to 2 GB in the environment. Note it is the total amount, not the amount you can use exclusively in your programs.
- The stack size is set unlimited (in C/C++), only capped by the total memory limit.
- Multi-processing or multi-threading is discouraged and unlikely beneficial, though not prohibited. Remember your programs will run on a single processor core. The total number of processes is limited to 64, including ones the system may create outside your programs.
- It is *never* recommended to run external commands. It is technically possible but probably does not work as you expect.

If you have no idea about what these mean — no worries. Just remember your programs should use the standard input and output, not files.

There are a couple more restrictions that apply:

- The total amount of source code must not exceed 256 KB in each submission.
- Your program must compile within 30 seconds.

See the DOMjudge team manual for more details about these restrictions.

Note about Platform

The judge system is running on Google Compute Engine, C2 machine type (`c2-standard-4`). For more information about Google Compute Engine, please visit

> https://cloud.google.com/compute/docs/cpu-platforms

Compilers & Options

The judge system uses the following compilers and execution environments (e.g., interpreters) with the following options. `"$@"` is substituted with your source file(s); `"$DEST"` is the name of the binary (which is "`./a.out`" by default) and is chosen arbitrarily by the system.

The "Run" commands indicated in the following table are for non-interactive problems. For interactive problems, standard input and output are connected to a judge program. See the "Note on Interactive Problems" section below for the details.

C	
Version	gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Compile	gcc -g -O2 -std=gnu11 -static -o "$DEST" "$@" -lm
Run	"$DEST" < *infile* > *outfile*
C++	
Version	g++ (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Compile	g++ -g -O2 -std=gnu++17 -static -o "$DEST" "$@"
Run	"$DEST" < *infile* > *outfile*
Java	
Version	openjdk version "11.0.17" 2022-10-18 OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-1ubuntu220.04)
Compile	javac -encoding UTF-8 -sourcepath . -d . "$@"
Run[1]	java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m *MainClass* < *infile* > *outfile*
Python 3 (PyPy)	
Version	Python 3.8.13 (7.3.9+dfsg-1~ppa1~ubuntu20.04, Apr 02 2022, 00:30:03) [PyPy 7.3.9 with GCC 9.4.0]
Compile[2]	pypy3 -m py_compile "$@"
Run	pypy3 "$@" < *infile* > *outfile*
Kotlin	
Version	kotlinc-jvm 1.7.10 (JRE 11.0.17+8-post-Ubuntu-1ubuntu220.04)
Compile	kotlinc -d . "$@"
Run[1]	kotlin -Dfile.encoding=UTF-8 -J-XX:+UseSerialGC -J-Xss64m -J-Xms1920m -J-Xmx1920m *MainClass* < *infile* > *outfile*

[1] DOMjudge will detect the main class automatically; you do not have to name it `Main`. See the DOMjudge team manual for details.
[2] Python's "Compile" commands only verify the syntax. `*.pyc` files will *not* be used in the real run.

The compilers and the execution environments are also available on your workstation as the following commands:

C	`compilegcc` (no "run" command)
C++	`compileg++` (no "run" command)
Java	`compilejava` / `runjava`
Python 3	`compilepython3` / `runpython3`
Kotlin	`compilekotlin` / `runkotlin`

Submission Results

Your submissions will eventually be responded with one of the following results:

Accepted

- **CORRECT** — Your program ran successfully and passed all test cases.

Rejected with 20-minute penalty

- **TIMELIMIT** — Your program did not finish within the time limit for some test case.
- **RUN-ERROR** — Your program crashed with some test case or otherwise exited with a non-zero exit status (e.g. because of missing "`return 0;`" in C/C++).
- **OUTPUT-LIMIT** — Your program produced excessive output (> 8 MB) for some test case.
- **WRONG-ANSWER** — Your program neither crashed nor exceeded the time limit, but produced incorrect output for some test case(s).
- **NO-OUTPUT** — Your program did not produce any output for some test case(s).

Priority of the rejected results

As mentioned in the DOMjudge Team Manual, the judges may have prepared multiple test files for each problem. DOMjudge will report back the results with the following priority:

- if one of the test cases causes **TIMELIMIT**, **RUN-ERROR**, or **OUTPUT-LIMIT**, it will be returned immediately.
- if not, and some test cases cause **WRONG-ANSWER**, it will be returned.
- if not, and some test cases cause **NO-OUTPUT**, it will be returned.

Rejected with no penalty

- **COMPILE-ERROR** — Your program did not compile on the judging environment. You can consult the error message(s) on the submission details page.
- **TOO-LATE** — Your program was submitted after the contest was over.[3]

[3] Note it does not mean your programs need to be *judged* before the end of the contest. Your programs will be judged as long as submitted ("*queued*") within the contest time.

Note on Interactive Problems

You may meet "interactive problems" in the contest. They are the same as other problems in a way that your program will read from standard input and print results to standard output. The difference is, the standard input and output are connected to a special program (judge program), with which you have to communicate back and forth. Unlike other problems where the input text is fixed for each test case, the input varies based on your previous outputs.

In most programming environments, program output is buffered to speed up I/O operations. With interactive problems, it is crucial to make sure the output is actually sent from your program and not simply stored in internal buffers. This typically means flushing the output buffers after each write.

- In C/C++ with `stdio.h` (`cstdio`), you can use `fflush(stdout)`. Writing "\n" does not mean it will get flushed.
- In C++ with `iostream`, an output stream is flushed automatically each time you write the `std::endl` manipulator. When using other means or if you want to be sure, call `std::cout.flush()`.
- In Java and Kotlin, the `System.out` stream has so-called "auto-flush" functionality and its buffer is therefore flushed automatically with each newline character. When using other streams or if you want to be sure, invoke the `flush()` method of the stream.
- In Python, you can use `sys.stdout.flush()`.

The time limit for an interactive problem is how much time your submission may spend; the time spent by the judge program is *not* counted towards this. Note that if your program attempts to read more input than can be provided currently (e.g., because you forgot to flush your previous output, or because of some other reason), then the program will stall indefinitely and your submission will get **TIMELIMIT**.

Note on Languages

The judges have solved all problems in languages from at least two of the three distinct language groups (Java/Kotlin, C/C++, and Python). They do not guarantee that all problems can be solved in any language.

Note to Python Users

Only syntax errors will be reported as **COMPILE-ERROR**. Other types of errors, such as `NameError` or `ModuleNotFoundError`, will result in **RUN-ERROR** and incur a 20-minute penalty.

It is fine, though not needed, to start your scripts with an interpreter directive (line starting with "`#!`", also known as *shebang*).[4]

The full list of modules available in the judge system can be found in the following section.

[4] Some past versions of DOMjudge refused scripts that contain a shebang.

Available Python Modules

```
__decimal
__exceptions__
__future__
__pypy__
_abc
_ast
_audioop_build
_audioop_cffi
_blake2
_bootlocale
_bz2
_cffi_backend
_cffi_ssl
_codecs
_codecs_cn
_codecs_hk
_codecs_iso2022
_codecs_jp
_codecs_kr
_codecs_tw
_collections
_collections_abc
_compat_pickle
_compression
_contextvars
_continuation
_cppyy
_crypt
_csv
_ctypes
_ctypes_test
_curses
_curses_build
_curses_cffi
_curses_panel
_dbm
_decimal_build
_distutils_system_mod
_dummy_thread
_ffi
_functools
_gdbm
_gdbm_build
_gdbm_cffi
_hashlib
_hpy_universal
_immutables_map
_imp
_io
_jitlog
_locale
_lsprof
_lzma
_lzma_build
_lzma_cffi
_markupbase
_marshal
_md5
_minimal_curses
_multibytecodec
_multiprocessing
_opcode
_operator
_osx_support
_overlapped
_pickle_support
_posixshmem
_posixshmem_build
_posixshmem_cffi
_posixsubprocess
_pwdgrp_build
_pwdgrp_cffi
_py_abc
_pydecimal
_pyio
_pypy_interact
_pypy_irc_topic
_pypy_openssl
_pypy_testcapi
_pypy_util_build
_pypy_util_cffi
_pypy_util_cffi_inner
_pypy_wait
_pypy_winbase_build
_pypy_winbase_cffi
_pypy_winbase_cffi64
_pypyjson
_random
_rawffi
_resource_build
_resource_cffi
_scproxy
_sha1
_sha256
_sha3
_sha512
_signal
_sitebuiltins
_socket
_sqlite3
_sqlite3_build
_sqlite3_cffi
_sre
_ssl
_ssl_build
_string
_strptime
_struct
_structseq
_sysconfigdata
_syslog_build
_syslog_cffi
_testcapi
_testing
_testmultiphase
_thread
_threading_local
_vmprof
_warnings
_weakref
_weakrefset
_winapi
abc
aifc
antigravity
appdirs
argparse
array
ast
asynchat
asyncio
asyncore
atexit
audioop
base64
bdb
binascii
binhex
bisect
builtins
bz2
cProfile
cachecontrol
calendar
certifi
cffi
cgi
cgitb
chardet
chunk
cmath
cmd
code
codecs
codeop
collections
colorama
colorsys
compileall
concurrent
configparser
contextlib
contextlib2
contextvars
copy
```

```
copyreg                       ipaddr                        pydoc
cpyext                        ipaddress                     pydoc_data
crypt                         itertools                     pyexpat
csv                           json                          pyparsing
ctypes                        keyword                       pypy_tools
ctypes_support                lib2to3                       pypyjit
curses                        linecache                     pyrepl
dataclasses                   locale                        queue
datetime                      lockfile                      quopri
dbm                           logging                       random
decimal                       lzma                          re
difflib                       macpath                       readline
dis                           macurl2path                   reprlib
distlib                       mailbox                       requests
distro                        mailcap                       resource
distutils                     marshal                       retrying
doctest                       math                          rlcompleter
dummy_threading               mimetypes                     runpy
easy_install                  mmap                          sched
email                         modulefinder                  secrets
encodings                     modules                       select
ensurepip                     msgpack                       selectors
enum                          msilib                        setuptools
errno                         msvcrt                        shelve
faulthandler                  multiprocessing               shlex
fcntl                         netrc                         shutil
filecmp                       nntplib                       signal
fileinput                     ntpath                        site
fnmatch                       nturl2path                    six
formatter                     numbers                       smtpd
fractions                     opcode                        smtplib
ftplib                        operator                      sndhdr
functools                     optparse                      socket
future_builtins               os                            socketserver
gc                            packaging                     sqlite3
genericpath                   parser                        sre_compile
getopt                        pathlib                       sre_constants
getpass                       pdb                           sre_parse
gettext                       pep517                        ssl
glob                          pickle                        stackless
greenlet                      pickletools                   stat
grp                           pip                           statistics
gzip                          pipes                         string
hashlib                       pkg_resources                 stringprep
heapq                         pkgutil                       struct
hmac                          platform                      subprocess
html                          plistlib                      sunau
html5lib                      poplib                        symbol
http                          posix                         symtable
identity_dict                 posixpath                     sys
idlelib                       pprint                        sysconfig
idna                          profile                       syslog
imaplib                       progress                      tabnanny
imghdr                        pstats                        tarfile
imp                           pty                           telnetlib
importlib                     pwd                           tempfile
inspect                       py_compile                    termios
io                            pyclbr                        test
```

```
textwrap                 tty                      wave
this                     turtle                   weakref
threading                turtledemo               webbrowser
time                     types                    webencodings
timeit                   typing                   wheel
tkinter                  unicodedata              wsgiref
token                    unittest                 xdrlib
tokenize                 urllib                   xml
toml                     urllib3                  xmlrpc
tputil                   uu                       zipapp
trace                    uuid                     zipfile
traceback                venv                     zipimport
tracemalloc              warnings                 zlib
```