

Judging Notes

1. Judge System

When you submit a program, the judge system determines a verdict by compiling and running it against undisclosed test input.

1.1. Input and Output

- Your program must read all input from **standard input** (stdin, e.g., `std::cin` in C++, `System.in` in Java, `sys.stdin` in Python).
- Your program must write all its output to **standard output** (stdout, e.g., `std::cout` in C++, `System.out` in Java, `sys.stdout` in Python).
- Output written to **standard error** (stderr, e.g., `std::stderr` in C++, `System.error` in Java, `sys.stderr` in Python) is ignored when judging.
- Your program is not allowed to open, create, or write to files. These file operations will result in a **RUN-ERROR**.
- Your program must exit with status code 0 upon successful completion. Exiting with a non-zero status code will result in a **RUN-ERROR**, even if the output is correct.

1.2. Execution Environment (Sandbox)

Your program will be run inside a *sandboxed environment*. This environment isolates the execution of your program to prevent system damage and to ensure fair and equal conditions for all contestants. This means several restrictions apply:

- **Memory Usage:** Total memory usage is limited to **2 GB**. This limit applies to the entire processes, including your program's code, static variables, stack, heap, and any virtual machine overhead (like for Java or Python). Exceeding this limit will typically result in a **RUN-ERROR**.
- **Stack Size:** In C/C++ and Java/Kotlin, the stack size is not explicitly limited, but is limited by the total memory limit. In Python, the stack size is limited by the Python runtime's default limit. Your program can attempt to change it by calling `sys.setrecursionlimit()`, but the stack size is still ultimately limited by the total memory limit.
- **Processes/Threads:**
 - Your program will be run on a **single processor core**. Using multiple threads or processes is allowed but generally discouraged as it will not likely improve performance.
 - The total number of processes allowed for a run is limited to **64**. This includes the main process of your program and any child processes, as well as processes the system might use internally. Exceeding this limit will result in a **RUN-ERROR**.
- **External Commands:** Running external commands (e.g., via `system()` in C/C++) is restricted and will not likely work as expected.
- **File System Access:** As mentioned, opening, creating or writing to files is not allowed.

1.3. Submission and Compilation

Your submission must meet these requirements:

- **Submission Files:** A single submission can contain single or multiple files. Each filename must start with a half-width alphanumerical character and may contain only half-width alphanumerical characters and `+. _ -`.
- **Source Code Size:** The total size of all source code files in a single submission must not exceed **256 KB**. Submissions larger than this will be rejected.
- **Compilation Time:** Your submission must compile successfully within **30 seconds**. A compilation taking longer than this will be aborted, resulting in a **COMPILER-ERROR**.
- **Compilation Memory Limit:** Total memory usage during compilation is limited to **2 GB**. A compilation exceeding this will be aborted, resulting in a **COMPILER-ERROR**.

1.4. About the Platform

The judge system runs your program on Google Compute Engine, C4 machine type (`c4-highcpu-2`, with `turboMode` set as `ALL_CORE_MAX`). For more information about Google Compute Engine, please visit the official website^{*1}.

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

2. 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 name(s); "\$DEST" is the name of the binary (which is `./a.out` by default) and is chosen arbitrarily by the system.

C	
Version	gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Compile	<code>gcc -x c -g -O2 -std=gnull -static -o "\$DEST" "\$@" -lm</code>
Run	<code>"\$DEST" < infile > outfile</code>
C++	
Version	g++ (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Compile	<code>g++ -x c++ -g -O2 -std=gnu++20 -static -o "\$DEST" "\$@"</code>
Run	<code>"\$DEST" < infile > outfile</code>
Java	
Version	OpenJDK 21.0.7 (build 21.0.7+6-Ubuntu-0ubuntu122.04)
Compile	<code>javac -encoding UTF-8 -sourcepath . -d . "\$@"</code>
Run	<code>java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m <i>MainClass</i> < infile > outfile</code>
Python 3 (PyPy)	
Version	Python 3.9.18 (PyPy 7.3.15 with GCC 10.2.1 20210130 (Red Hat 10.2.1-11))
Compile	<code>pypy3 -m py_compile "\$@"</code>
Run	<code>pypy3 "\$@" < infile > outfile</code>
Kotlin	
Version	1.7.21 (JRE 21.0.7+6-Ubuntu-0ubuntu122.04)
Compile	<code>kotlinc -d . "\$@"</code>
Run	<code>kotlin -Dfile.encoding=UTF-8 -J-XX:+UseSerialGC -J-Xss64m -J-Xms1920m -J-Xmx1920m <i>MainClass</i> < infile > outfile</code>

For Java and Kotlin submissions, you do not need to name your main class `Main`. When you submit, the web interface tries to auto-detect the main class file. Double-check that it is correct before submitting multiple files.

In Python, **Compile** commands only verify the syntax using the `py_compile` module. `*.pyc` files will not be used in the actual run.

3. Verdicts of Submissions

The judge system runs your program against multiple inputs prepared by the judges.

The system first decides the verdicts of your submission for each input. Then, based on those verdicts, the system decides an overall single verdict of your submission. You will only see this overall verdict, not the verdicts for each input.

3.5. Possible Verdicts on Each Input

For each input, the judge system decides one of the following verdicts:

- **CORRECT:** Your program completed the run successfully within the time limit, and produced the correct output.
 - The correct output is prepared by the judges.
 - Minor differences in whitespaces are ignored on checking your output. For example, an extra whitespace at the end of a line or an extra blank line at the end of the output will be ignored.
 - In some problems, the correct output may not be unique. The specific checking method will be documented in the problem statement (e.g. with floating point answers).
- **TIMELIMIT (Time Limit Exceeded):** Your program did not complete the run within the per-problem time limit and was aborted.
- **RUN-ERROR (Runtime Error):** Your program terminated abnormally during the run. Common causes include:
 - Attempting illegal operations (e.g., division by zero).
 - Accessing invalid memory (e.g., array index out of bounds, null pointer dereferencing).
 - Exceeding the memory limit.
 - Aborting by `assert()`.
 - Exiting with a non-zero status code (e.g., forgetting `return 0;` at the end of `main` in C). The exit status code is checked *before* the output is checked.
 - Crashing for any other reason.
- **WRONG-ANSWER:** Your program completed the run within the time limit without crashing, but produced an output different from the correct output.
- **OUTPUT-LIMIT (Output Limit Exceeded):** Your program produced an output that is larger than **8 MB**. This limit does not consider output written to standard error.
- **NO-OUTPUT:** Your program completed the run successfully within the time limit without crashing but did not produce any output on standard output.

3.6. Overall Verdict of a Submission

The judge system decides one overall verdict of a submission as follows:

3.6.1. Accepted

- **CORRECT**: Your program resulted in **CORRECT** for all input.

3.6.2. Rejected (with 20 Minute Penalty)

If the system decides any five verdicts below for an input, the decided verdict will be the overall verdict of the submission, and it will incur a **20-minute time penalty** per-submission if the problem is eventually solved.

- **TIMELIMIT**
- **RUN-ERROR**
- **WRONG-ANSWER**
- **OUTPUT-LIMIT**
- **NO-OUTPUT**

If multiple inputs fail with different verdicts (e.g., one **TIMELIMIT** and one **WRONG-ANSWER**), the system decides one of these verdicts as the overall verdict of the submission. In this case, there's no guarantee which verdict is selected.

3.6.3. Rejected (with No Penalty)

Two verdicts below mean that the system did not run your program against the inputs. They **do not incur any time penalty**.

- **COMPILER-ERROR**: The judge system failed to compile your source code. You can see the compiler's error messages on the submission details page.
- **TOO-LATE**: The judge system received your submission after the contest ends. Note that submissions received before the contest ends will be judged even after the contest ends, and the system decides a non-**TOO-LATE** verdict.

4. 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).

5. Note to Python Users

Only syntax errors will be reported as **COMPILER-ERROR**. Other 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 a shebang).^{*2}

2. Some past versions of DOMjudge refused scripts that contain a shebang.

5.7. Available Python Modules

<code>__decimal</code>	<code>_rawffi</code>	<code>email</code>	<code>pyexpat</code>
<code>__exceptions__</code>	<code>_resource_build</code>	<code>encodings</code>	<code>pypy_tools</code>
<code>__future__</code>	<code>_resource_cffi</code>	<code>ensurepip</code>	<code>pypyjit</code>
<code>__pypy__</code>	<code>_scproxy</code>	<code>enum</code>	<code>pyrepl</code>
<code>_abc</code>	<code>_sha1</code>	<code>errno</code>	<code>queue</code>
<code>_aix_support</code>	<code>_sha256</code>	<code>faulthandler</code>	<code>quopri</code>
<code>_ast</code>	<code>_sha3</code>	<code>fcntl</code>	<code>random</code>
<code>_audioop_build</code>	<code>_sha512</code>	<code>filecmp</code>	<code>re</code>
<code>_audioop_cffi</code>	<code>_signal</code>	<code>fileinput</code>	<code>readline</code>
<code>_blake2</code>	<code>_sitebuiltins</code>	<code>fnmatch</code>	<code>reprlib</code>
<code>_bootlocale</code>	<code>_socket</code>	<code>formatter</code>	<code>resource</code>
<code>_bootsubprocess</code>	<code>_sqlite3</code>	<code>fractions</code>	<code>rlcompleter</code>
<code>_bz2</code>	<code>_sqlite3_build</code>	<code>ftplib</code>	<code>runpy</code>
<code>_cffi_backend</code>	<code>_sqlite3_cffi</code>	<code>functools</code>	<code>sched</code>
<code>_cffi_ssl</code>	<code>_sre</code>	<code>future_builtins</code>	<code>secrets</code>
<code>_codecs</code>	<code>_ssl</code>	<code>gc</code>	<code>select</code>
<code>_codecs_cn</code>	<code>_ssl_build</code>	<code>genericpath</code>	<code>selectors</code>
<code>_codecs_hk</code>	<code>_string</code>	<code>getopt</code>	<code>shelve</code>
<code>_codecs_iso2022</code>	<code>_strptime</code>	<code>getpass</code>	<code>shlex</code>
<code>_codecs_jp</code>	<code>_struct</code>	<code>gettext</code>	<code>shutil</code>
<code>_codecs_kr</code>	<code>_structseq</code>	<code>glob</code>	<code>signal</code>
<code>_codecs_tw</code>	<code>_sysconfigdata</code>	<code>graphlib</code>	<code>site</code>
<code>_collections</code>	<code>_sysconfigdata__linux_x86_64-linux-gnu</code>	<code>greenlet</code>	<code>smtpd</code>
<code>_collections_abc</code>	<code>_syslog_build</code>	<code>grp</code>	<code>smtplib</code>
<code>_compat_pickle</code>	<code>_syslog_cffi</code>	<code>gzip</code>	<code>sndhdr</code>
<code>_compression</code>	<code>_testcapi</code>	<code>hashlib</code>	<code>socket</code>
<code>_contextvars</code>	<code>_testing</code>	<code>heapq</code>	<code>socketserver</code>
<code>_continuation</code>	<code>_testmultiphase</code>	<code>hmac</code>	<code>sqlite3</code>
<code>_cppyy</code>	<code>_testmultiphase_build</code>	<code>html</code>	<code>sre_compile</code>
<code>_crypt</code>	<code>_thread</code>	<code>http</code>	<code>sre_constants</code>
<code>_csv</code>	<code>_threading_local</code>	<code>identity_dict</code>	<code>sre_parse</code>
<code>_ctypes</code>	<code>_tkinter</code>	<code>idlelib</code>	<code>ssl</code>
<code>_ctypes_test</code>	<code>_vmprof</code>	<code>imaplib</code>	<code>stackless</code>
<code>_ctypes_test_build</code>	<code>_warnings</code>	<code>imghdr</code>	<code>stat</code>
<code>_curses</code>	<code>_weakref</code>	<code>imp</code>	<code>statistics</code>
<code>_curses_build</code>	<code>_weakrefset</code>	<code>importlib</code>	<code>string</code>
<code>_curses_cffi</code>	<code>_winapi</code>	<code>inspect</code>	<code>stringprep</code>
<code>_curses_panel</code>	<code>abc</code>	<code>io</code>	<code>struct</code>
<code>_dbm</code>	<code>aifc</code>	<code>ipaddress</code>	<code>subprocess</code>
<code>_decimal_build</code>	<code>antigravity</code>	<code>itertools</code>	<code>sunau</code>
<code>_ffi</code>	<code>argparse</code>	<code>json</code>	<code>symbol</code>
<code>_gdbm</code>	<code>array</code>	<code>keyword</code>	<code>symtable</code>
<code>_gdbm_build</code>	<code>ast</code>	<code>lib2to3</code>	<code>sys</code>
<code>_gdbm_cffi</code>	<code>asynchat</code>	<code>linecache</code>	<code>sysconfig</code>
<code>_hashlib</code>	<code>asyncio</code>	<code>locale</code>	<code>syslog</code>
<code>_hpy_universal</code>	<code>asyncore</code>	<code>logging</code>	<code>tabnanny</code>
<code>_immutable_map</code>	<code>atexit</code>	<code>lzma</code>	<code>tarfile</code>
<code>_imp</code>	<code>audioop</code>	<code>macpath</code>	<code>telnetlib</code>
<code>_io</code>	<code>base64</code>	<code>macurl2path</code>	<code>tempfile</code>
<code>_jitlog</code>	<code>bdb</code>	<code>mailbox</code>	<code>termios</code>
<code>_locale</code>	<code>binascii</code>	<code>mailcap</code>	<code>test</code>
<code>_lsprof</code>	<code>binhex</code>	<code>marshal</code>	<code>textwrap</code>
<code>_lzma</code>	<code>bisect</code>	<code>math</code>	<code>this</code>
<code>_lzma_build</code>	<code>builtins</code>	<code>mimetypes</code>	<code>threading</code>
<code>_lzma_cffi</code>	<code>bz2</code>	<code>mmap</code>	<code>time</code>
<code>_markupbase</code>	<code>cProfile</code>	<code>modulefinder</code>	<code>timeit</code>
<code>_marshal</code>	<code>calendar</code>	<code>msilib</code>	<code>tkinter</code>
<code>_md5</code>	<code>cffi</code>	<code>msvcrt</code>	<code>token</code>
<code>_minimal_curses</code>	<code>cgi</code>	<code>multiprocessing</code>	<code>tokenize</code>
<code>_multibytecodec</code>	<code>gitb</code>	<code>netrc</code>	<code>tputil</code>

_multiprocessing	chunk	nntplib	trace
_opcode	cmath	ntpath	traceback
_operator	cmd	nturl2path	tracemalloc
_osx_support	code	numbers	tty
_overlapped	codecs	opcode	turtle
_pickle_support	codeop	operator	turtledemo
_posixshm	collections	optparse	types
_posixshm_build	colorsys	os	typing
_posixshm_cffi	compileall	parser	unicodedata
_posixsubprocess	concurrent	pathlib	unittest
_pwdgrp_build	configparser	pdb	urllib
_pwdgrp_cffi	contextlib	pickle	uu
_py_abc	contextvars	pickletools	uuid
_pydecimal	copy	pipes	venv
_pyio	copyreg	pkgutil	warnings
_pypy_generic_alias	cpyext	platform	wave
_pypy_interact	crypt	plistlib	weakref
_pypy_irc_topic	csv	poplib	webbrowser
_pypy_openssl	ctypes	posix	wsgiref
_pypy_testcapi	ctypes_support	posixpath	xdrlib
_pypy_util_build	curses	pprint	xml
_pypy_util_cffi	dataclasses	profile	xmlrpc
_pypy_util_cffi_inner	datetime	pstats	zipapp
_pypy_wait	dbm	pty	zipfile
_pypy_winbase_build	decimal	pwd	zipimport
_pypy_winbase_cffi	difflib	py_compile	zlib
_pypy_winbase_cffi64	dis	pyclbr	zoneinfo
_pypyjson	distutils	pydoc	
_random	doctest	pydoc_data	