Judging Notes

1. Judge System

When you submit source code, the judge system decides a verdict by compiling and running it against undisclosed test inputs.

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::cerr` in C++, `System.err` 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. 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 your program to prevent system damage and to ensure 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 whole 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++, the stack size is not explicitly limited, but is limited by the total memory limit. In Java/Kotlin, the stack size is limited by the `-Xss` command line parameter. In Python, the stack size is limited by the Python runtime's default limit. Your program can attempt to change this limit 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 is unlikely to 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 is unlikely to 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 one or more files. Each file name must start with an ASCII alphanumeric character (a "half-width" `A`-`Z`, `a`-`z`, or `0`-`9`) and may contain only ASCII alphanumeric characters and `+.\_-`.
- **Source Code Size:** The total size of all source code files in a single submission must not exceed **256 KB**.
- **Source Code Encoding:** All source code files must be encoded in UTF-8. Submitting files with other encodings (e.g., Shift_JIS) may result in a **COMPILER-ERROR** or unexpected behavior.
- **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 Usage:** 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 to `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 and is chosen by the system.

C	
Version	gcc 13
Compile	`gcc -x c -g -O2 -std=gnu11 -static -o "$DEST" "$@" -lm`
Run	`"$DEST" < infile > outfile`
C++	
Version	g++ 13
Compile	`g++ -x c++ -g -O2 -std=gnu++20 -static -o "$DEST" "$@"`
Run	`"$DEST" < infile > outfile`
Java	
Version	OpenJDK 21
Compile	`javac -encoding UTF-8 -sourcepath . -d . "$@"`
Run	`java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m MainClass < infile > outfile`
Python 3 (PyPy)	
Version	Python 3.11.11 (PyPy 7.3.18)
Compile	`pypy3 -m py_compile "$@"`
Run	`pypy3 "$@" < infile > outfile`
Kotlin	
Version	1.9.24
Compile	`kotlinc -d . "$@"`
Run	`kotlin -Dfile.encoding=UTF-8 -J-XX:+UseSerialGC -J-Xss64m -J-Xms1920m -J-Xmx1920m MainClass < infile > outfile`

For Java and Kotlin submissions, the system automatically detects the main class. You do not need to name your main class `Main`.

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 verdict for each input against your submission. Then, based on those verdicts, the system decides the submission's overall verdict. You will only see this overall verdict, not the verdict for each input.

3.1. Possible Verdicts on Each Input

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

- **CORRECT:** The judge system determined that your program completed its run within the time limit and wrote output that was validated as correct.
  - The validation method is prepared by the judges. This may involve comparing the output to a specific answer or using a custom program for problems where the correct output is not unique.
  - Minor differences in whitespace are ignored in common validation. For example, an extra whitespace at the end of a line or an extra blank line at the end of the output can be ignored.
  - Differences between uppercase and lowercase letters are ignored. For example, `Yes` and `yes` are treated the same.
  - For problems where the correct output may not be unique (e.g., with floating-point answers), the specific validation method will be documented in the problem statement.
- **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 invalid operations (e.g., division by zero).
  - Accessing invalid memory (e.g., array index out of bounds, null pointer dereferencing).
  - Exceeding the memory limit (2 GB).
  - 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 wrote output that was not validated as correct.
- **OUTPUT-LIMIT (Output Limit Exceeded):** Your program wrote output exceeding **8 MB**. This limit does not apply to output written to standard error.
- **NO-OUTPUT:** Your program completed the run successfully within the time limit without crashing but wrote nothing to standard output.

3.2. Overall Verdict of a Submission

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

3.2.1. Accepted

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

3.2.2. Rejected (with 20 Minute Penalty)

If the system decides any of the five verdicts below for an input, the system decides that verdict as 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 result in different verdicts including ones from this list (e.g., one **TIMELIMIT** and one **WRONG-ANSWER**), the system decides one of the verdicts from the list as the overall verdict of the submission. In this case, there is no guarantee which one will be selected.

3.2.3. 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 those submissions are judged normally and will not receive a **TOO-LATE** verdict.

4. Note on Languages

For every problem, the judges have verified solutions 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 a **COMPILER-ERROR**. Other errors, such as `NameError` or `ModuleNotFoundError`, will result in a **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.

5.1. Available Python Modules

```
__decimal           _pypy_winbase_cffi  dis                 pyexpat
__exceptions__      _pypy_winbase_cffi64 distutils           pypy_tools
__future__          _pypyjson           doctest             pypyjit
__hello__           _random             email               pyrepl
__phello__          _rawffi             encodings           queue
__pypy__            _resource_build     ensurepip           quopri
_abc                _resource_cffi      enum                random
_aix_support        _scproxy            errno               re
_ast                _sha1               faulthandler        readline
_audioop_build      _sha256             fcntl               reprlib
_audioop_cffi       _sha3               filecmp             resource
_blake2             _sha512             fileinput           rlcompleter
_bootsubprocess     _signal             fnmatch             runpy
_bz2                _sitebuiltins       fractions           sched
_cffi_backend       _socket             ftplib              secrets
_cffi_ssl           _sqlite3            functools           select
_codecs             _sqlite3_build      future_builtins     selectors
_codecs_cn          _sqlite3_cffi       gc                  shelve
_codecs_hk          _sre                genericpath         shlex
_codecs_iso2022     _ssl                getopt              shutil
_codecs_jp          _ssl_build          getpass             signal
_codecs_kr          _string             gettext             site
_codecs_tw          _strptime           glob                smtpd
_collections        _struct             graphlib            smtplib
_collections_abc    _structseq          greenlet            sndhdr
_colorize           _sysconfigdata      grp                 socket
_compat_pickle      _syslog_build       gzip                socketserver
_compression        _syslog_cffi        hashlib             sqlite3
_contextvars        _testcapi           heapq               sre_compile
_continuation       _testing            hmac                sre_constants
_cppyy              _testmultiphase     html                sre_parse
_crypt              _testmultiphase_build http                ssl
_csv                _thread             identity_dict       stackless
_ctypes             _threading_local    idlelib             stat
_ctypes_test        _tkinter            imaplib             statistics
_ctypes_test_build  _vmprof             imghdr              string
_curses             _warnings           imp                 stringprep
_curses_build       _weakref            importlib           struct
_curses_cffi        _weakrefset         inspect             subprocess
_curses_panel       _winapi             io                  sunau
_dbm                abc                 ipaddress           symtable
_decimal_build      aifc                itertools           sys
_ffi                antigravity         json                sysconfig
_gdbm               argparse            keyword             syslog
_gdbm_build         array               lib2to3             tabnanny
_gdbm_cffi          ast                 linecache           tarfile
_hashlib            asynchat            locale              telnetlib
_hpy_universal      asyncio             logging             tempfile
_immutables_map     asyncore            lzma                termios
_imp                atexit              mailbox             test
_io                 audioop             mailcap             textwrap
_jitlog             base64              marshal             this
_locale             bdb                 math                threading
_lsprof             binascii            mimetypes           time
_lzma               bisect              mmap                timeit
_lzma_build         builtins            modulefinder        tkinter
_lzma_cffi          bz2                 msilib              token
_markupbase         cProfile            msvcrt              tokenize
_marshal            calendar            multiprocessing     tomllib
_md5                cffi                netrc               tputil
```

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