



Judging Notes

Revised: Dec 17, 2024

1. The following guidelines apply to clarification requests:
 - If a team feels that a problem is stated ambiguously, they may submit a clarification request.
 - The judges may refuse to answer if they believe that the problem is stated clearly enough.
 - The judges will not answer questions about algorithms, test data, etc.
2. The following guidelines apply to judge responses:
 - A judge response of "**CORRECT**" means that your submission was correct!
 - A judge response of "**WRONG-ANSWER**" means that your program output a wrong answer (or did not match the judges' output within the specified tolerance).
 - A judge response of "**TIMELIMIT**" means that your program ran for too long on at least one test case.
 - A judge response of "**RUN-ERROR**" means that your program crashed before completing. There may be many causes for this, including but not limited to division by zero, subscript out of bounds, out of memory, etc.
 - A judge response of "**COMPILER-ERROR**" means that your submission was never run because it did not compile.
 - The time limits for each problem will be specified in the problem statement. If your program runs longer than the specified time limit on ANY test case, your submission will be judged "**TIMELIMIT**", unless it receives another incorrect judgment.
 - There is no such thing as "Presentation Error" or "Format Error." If you misspell the word "*infinity*", for example, and the problem requires that word as output, then your submission will be judged as "**WRONG-ANSWER**".
 - If you submit a solution that receives a judgment of "**COMPILER-ERROR**", then you will be notified of it (just as any other error). However, compiler errors do not count toward penalty time.
 - If a submission is "**CORRECT**" then no subsequent submissions on that problem will count against you, even if they are not correct.
3. The problem set may include one or more *interactive problems*. See the next page for additional notes on interactive problems.



Additional Judging Notes on Interactive Problems

- For interactive problems, you usually need to flush the output buffer after outputting each line (that is, after each newline character), as shown below:
 - In C (or C++ using `cstdio`), you can use `fflush(stdout)`. A C++ output stream is flushed automatically each time you write the `endl` manipulator. When using other means or if you want to be sure, call `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()`.
- Interactive problems are judged in a way similar to other problems, but there are some differences:
 - When your program attempts to read data, it will wait until more data are available or until the judge program terminates the input (unless you read in a non-blocking way, which is beyond the scope of these notes). Thus, if your program attempts to read more input than can currently be provided (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**".
 - As usual, the judgement given to an incorrect submission is the first error discovered, but this does not always mean exactly the same thing as for traditional problems. For instance, if your submission to a traditional problem is too slow on a particular test case, you would get "**TIMELIMIT**" on that test case. With an interactive problem, the judgement may be "**WRONG-ANSWER**", if the solution exhibits an incorrect answer before it runs out of time.
 - 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.
 - The judge program may behave in an adversarial way and adapt the input provided to your program based on your previous output, with the intent to discover errors in your algorithm.
 - Because of timing between the interactive judge program and your submission, judgements for incorrect submissions to interactive problems are not necessarily deterministic. We can guarantee the following: a judgement of "**WRONG-ANSWER**" means that your submission produced incorrect output, and a judgement of "**RUN-ERROR**" means that your submission returned a non-zero exit code. If your submission does both, you may receive either judgement.