JUDGING NOTES

Updated: March 20, 2025

1. If a team feels that a problem is stated ambiguously or that there is an error in a problem statement, in test data, or with the judging system, they may submit a clarification request. The judges will investigate, but may refuse to answer if they believe that the problem is stated clearly enough or that the concerns are unfounded.

2. The following guidelines apply to handling input in programs:
   - All input comes from standard input.
   - The problem statement will specify the format and bounds for all input data. Test cases are guaranteed to respect this specification.
   - Unless the problem explicitly states otherwise, the input for a problem consists of data for a single test case. The judge system will execute a new instance of your program for each test case. If the input contains multiple test cases, then the problem will also state an upper bound on the number of test cases.

3. The following guidelines apply to handling output in programs:
   - Print all output to standard output.
   - Format your output as specified in the problem statement and as illustrated in the sample output. Follow all instructions in the problem statement regarding problem output (the order in which to print multiple pieces of output; relative or absolute tolerances allowed for floating-point output; etc.)
   - Unless the problem explicitly states otherwise, and with the exceptions below regarding whitespace, your output must exactly match the problem statement specification to be judged correct. Misspellings, incorrect case, and extraneous standard output (such as debugging prints) are not accepted.
   - Unless the problem explicitly states otherwise, the judge system will treat whitespace leniently: leading and trailing whitespace is allowed, and any sequence of one or more consecutive whitespace characters is treated as equivalent to any other such sequence.

4. The problem set may include one or more interactive problems. In many respects, interactive problems are like any other problem – your program will read from standard input and print results to standard output. However, with an interactive problem, standard input and output of your program are connected to another (judge) program, with which your program must communicate back and forth.

5. The following guidelines apply to Judge responses:
   - A judge response of “Accepted” means that your submission was correct!

- A judge response of “Wrong Answer” means that your program produced incorrect output, or output not formatted according to the problem statement specifications.
- A judge response of “Time Limit Exceeded” means that your program ran for too long on at least one test case.
- A judge response of “Runtime Error” means that your program crashed before completing. There may be many causes for this error, 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.

6. The following guidelines apply to contest scoring:
   - The problems are all of equal value.
   - A team can be credited with solving a given problem at most once.
   - The scoreboard is sorted by the greatest number of problems solved. The team with the most solved problems wins.
   - Ties in number of solutions are broken by fewest penalty minutes.
   - Teams accrue one penalty minute from the beginning of the contest per problem, until they solve that problem. For example, if a team solves a problem after one hour from the start of the contest, that’s 60 penalty minutes. If they solve another problem after two hours from the start of the contest, that’s an additional 120 penalty minutes, for a total of 180.
   - Twenty (20) penalty minutes are assessed to a team for each of their incorrect submissions of a problem up until a submission of theirs for that problem is accepted.
   - No penalty minutes are assessed to a team for problems that they never solve.
   - No additional penalty minutes are assessed to a team for their submissions of a problem that they have already had accepted, regardless of the judgment.
   - No penalty minutes are assessed to a team for a submission that receives a judgment of “Compiler Error”.

Additional Judging Notes on Interactive Problems

1. 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 line of output (that is, after each *newline* character) as follows:
   - 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()`**.

2. 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 Time Limit Exceeded.
   - 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 Time Limit Exceeded 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 Time Error means that your submission returned a non-zero exit code. If your submission does both, you may receive either judgement.

3. For some problems, the judges may provide a simple testing tool simulating the judging process. If this is the case, you will find such a tool where you normally see the sample data. Executing the tool with a "-h" option should describe how to use the tool. Of course, the testing tool will only implement some test scenarios and only some functionality of the real judge program.