|  |  |
| --- | --- |
| **Notes to Teams** | *Last modified November 4, 2012* |

1. Programs must be in a single source file with the name specified in the problem description; failure to meet this requirement is a [*Submission Error*](file:///\\errors.html). Java programmers, note that you can include additional top-level classes in a single source file as long as they are unqualified (just "class", without "public"). Java programmers, note as well that your classes must be defined within the default package.
2. All output will be judged using a file comparison utility, so output must be *exactly* as shown in the examples. This applies to all problems, whether explicitly stated in the problem description or not. Spelling, punctuation, spacing, and case (uppercase/lowercase) are all significant.
3. Programs must write their results to standard output (usually *stdout* in C, *cout* in C++, and *System.out* in Java). The judges will ignore all output to standard error (usually *stderr* in C, *cerr/clog* in C++, and *System.err* in Java), so you can write as much debugging information to standard error as you want, subject to the one-minute time limit.
4. Your program cannot require any intervention by the user. For example, if you pause the program and ask the user to press a key to continue, you will be flagged with a [*Submission Error*](file:///\\errors.html). If you pause the program without any prompting at all, you may be flagged with a [*Time Limit Exceeded*](file:///\\errors.html) error.
5. Do *not* use drive and/or path specifications when naming input files. If a problem indicates that the input file is named file.in, then you must open file.in and not a:file.in or c:\stuff\file.in or anything else. Violating this rule will result in a [*Submission Error*](file:///\\errors.html).
6. All test cases used in judging will conform to the input specifications. It is not necessary for you to detect invalid input.
7. Input data and correct output data will obey the following rules.
   * Other than end-of-line characters, spaces are the only whitespace that appear.
   * Two or more consecutive spaces do not appear, unless specifically mentioned in the problem statement.
   * Spaces do not appear at the end of lines.
   * Spaces do not appear at the beginning of lines, unless specifically mentioned in the problem statement.
   * Blank lines do not appear.
   * All lines, including the last line, end with the standard end-of-line marker.
8. This applies only to Java programmers. Counter to Java conventions, the name of your source file and main class must be in ***lowercase*** for this competition. For example, if a problem states that your program must be called compute, then you would create a file called compute.java that begins like this:

public class compute {  
 public static void main(String args[]) {  
 ...  
 }  
 ....  
}

1. Teams are ranked by the number of problems solved, with teams solving the same number of problems ranked by least total time (see [Regional Scoring](http://cm2prod.baylor.edu/ICPCWiki/Wiki.jsp?page=Regional%20Rules#section-Regional+Rules-ScoringOfARegionalContest)). Teams solving the same number of problems with the same total time are ranked by the smallest *elapsed* time of their last accepted solution (not counting penalties for rejected runs). Any remaining ties are left unbroken unless they affect the regional winners, in which case they are broken by a coin flip.
2. Any team that jeopardizes the integrity of the contest or violates the rules of the contest will be disqualified and the team members may be banned permanently from competing in the Mid-Central Region. Some examples of such actions are:
   * accessing the Internet in ***any*** way,
   * disrupting power to computers,
   * corrupting judging materials or the judging process,
   * collaborating with anyone not on the team (this includes using a portable phone),
   * disobeying site officials' instructions regarding appropriate conduct.

|  |  |
| --- | --- |
| **Error Messages** | *Last modified November 6, 2010* |

If a submission is not correct it will be returned with the *first* applicable error message from the following list. A 20-minute time penalty will be assessed for each incorrect submission. The error messages are described below.

1. Submission Error
2. Run-time Error
3. Time Limit Exceeded
4. Wrong Answer
5. Presentation Error

**Submission Error.** This indicates a violation of the basic contest rules. Teams should never get a *Submission Error*, but in the heat of competition they sometimes make silly mistakes. Submission errors include (but are not limited to)

* compilation errors (warnings are OK)
* submitting the wrong problem
* submitting the executable instead of the source code
* submitting a program consisting of more than one source file
* using the wrong source or input file name
* specifying a java class as part of a package
* including a drive and/or path with a file name
* prompting and waiting for keyboard input (see also *Time Limit Exceeded*).

**Run-time Error.** The program crashed when run. Examples of run-time errors include (but are not limited to)

* stack overflows
* I/O errors
* heap overflows
* invalid memory references (*e.g.*, dereferencing a null pointer)
* bizarre behavior (garbage on screen, beeping, keyboard lockup, ...).

**Time Limit Exceeded.** The program ran (without crashing) for more than one minute. It may or may not have produced any output. This error may indicate that the program simply runs too slowly or that it was silently waiting for keyboard input, but it usually means that it got stuck in an infinite loop.

**Wrong Answer.** The program ran to completion in one minute or less but produced one or more incorrect answers. A program that produces unsorted output when the problem calls for sorted output will receive this error.

**Presentation Error.** The program ran to completion in one minute or less, produced answers that appear to contain essentially the same information as the correct answers, but formatted the output incorrectly in one or more cases. Examples of incorrect formatting include (but are not limited to)

* misspelled words or omitted non-essential words
* wrong case (*e.g.*, uppercase instead of lowercase)
* too few or too many spaces
* wrong number of significant digits in numeric output
* incorrect or missing punctuation
* extra blank lines.

**Note:** Receiving a *Presentation Error* guarantees that there are one or more formatting errors, and indicates that the judges did not see any wrong answers based on a brief visual inspection. However, it is not a *guarantee* that the program produced no incorrect answers.