Contest environment instructions

Host Scientific Committee IOI 2012

In your computer you will find a directory named after each of the tasks in the competition. That directory will contain development material that you may find useful in working on that task. You are not required to use it, thus it is there only for your convenience. You can also find a copy of the same files distributed in compressed form by the contest Web interface.

You are free to modify the files that you will find in those directories: anyway, the only files that will be considered for evaluation will be those submitted via the contest Web interface (CMS).

1 Programming tasks

Programming tasks require you to submit one or more source code files in your preferred language among those available for the competition (`C`, `C++` and `Pascal`). If one tasks requires you to submit two or more source code files, they should all be written in the same language.

For programming tasks, you can find at least the following files in the task director.

Template source files. For each file that you have to submit, you find a template written in each competition language. Of course, these templates do not implement a correct solution and thus submitting them do not earn you any point in general. However, you are encouraged to write your solution by modifying the template files, so to be sure to correctly implement the interface required by the task.

Compilation script. For each language, you find a script that compiles your program (or programs) using the official compilation command. You can tweak this script to change the flags passed to the compiler (for example, to enable debugging symbols).

Graders, headers, ... These files are used by the compilation scripts to produce a fully working program from your source code. They are provided to you for testing your solution, but they may be different from those actually used in the grading system: your solution should be conforming to the interface specified in the task description, without relying on tricks or shortcomings possibly contained in these files.

1.1 Usage

In order to work on a programming task, you should edit the relevant files with your favorite editing program and implement the required routines, as indicated in the "implementation details" part of the task statement. To compile a program, you can run the script `compile_c.sh`, `compile_cpp.sh` or `compile_pas.sh` or, when applicable, configure your IDE (Integrated Development Environment) to perform a similar operation.

If the compilation terminates correctly, one or more executable files will be generated: you can run them, feeding them with a test case on the standard input and collecting their answer on the standard output. The "implementation details" section for each task will document the format to use for input and output.

Some programming tasks may deviate from these procedures because of their peculiar structure: the "implementation details" section will contain all due explanation.

1.2 Caveats

File access. Under no condition your submitted code should try to access file streams, like the standard input/output, or other files. These actions will probably interfere with the grading process, possibly leading a correct solution to be considered incorrect. They are banned anyway, and leaving these commands in your source code is interpreted as an attack towards the system and used as evidence for disqualification.

Beside that, writing on the standard error in your code could significantly disrupt the performance of your submitted solution. You are advised to check you submissions and get rid of debugging instructions before posting them to the system.

Input/output buffering. Since each system call invoked by a solution during evaluation causes a slight time overhead in the execution of the program, the grading programs are instructed to use a comparatively large buffer for reading input files and writing output files. Contestants should not further modify these settings. Time and memory limits are already tuned to take into consideration the overhead introduced by processing input/output files.

2 Output-only tasks

Output-only tasks will require you to submit, instead of a program source code, the output files originated by processing some input files according to the rules prescribed in the task statement.

The task directory will contain a copy of the input files, for which you are expected to produce the output files to be submitted via the Web interface. You can work on them using any of the tools available in the contest environment.

Some output-only tasks may deviate from these procedures because of their peculiar structure: the "implementation details" section will contain all due explanation.