Abacus contestant manual

Bruce Merry *bmerry@users.sourceforge.net*
Max Rabkin *max.rabkin@gmail.com*

21 October 2017

1 Introduction

Abacus is a software package used for the South African Regional of the ACM International Collegiate Programming Contest. You will use it to submit your solutions and get feedback from the judges.

2 Logging into Abacus

Abacus is accessed through the Firefox web browser. Depending on your local installation, there may be a direct link on the desktop. Other browsers may be available but are not supported.

Once the page has loaded, fill in the username and password that you have been given. If you do this correctly, you will be taken to the Manual tab.

When the contest is running, you will see a clock in the top-left corner counting down the remaining time.

3 Clarifications

If you are uncertain about some details in a task or the contest in general, you may submit a clarification request. On the Clarification requests tab, click the Request Clarification. Select the problem your query pertains to and type in your query.

When a judge responds to your query, the response will be available on the Clarifications tab (you can click on clarification requests and clarifications to see the full text).

Some things to take note of:

1. The newest information appears at the top.

2. In a distributed (multi-site) contest, the judges might not be on-site. Thus, if you have specific questions about the installation at your site (e.g. your IDE won't open) then speak to somebody at the site.

3. The judges do not know who is submitting queries. This means that if you want to refer to a previous query you made, you should include the Request ID that appears in the list box. Similarly, if your query is about a particular submission, you should include the Submission ID from the Submissions tab (or use the Request Clarification button on that tab – see Section 5, Submitting your solution). Even if you include a submission ID, you still need to ask a specific question.

4. You may receive clarifications for questions that you never asked. These are broadcast replies and indicate important information for all contestants.

4 Writing your solution

If you are using Java, you can call your main class whatever you like, but it must be the *first* class in the file. You can include support classes after it, but they should not be public.

Your solution will read from standard input (`stdin`, `cin`, `System.in`, `sys.stdin`, depending on the language) and write to standard output (`stdout`, `cout`, `System.out`, `sys.stdout` again depending on the language). Your solution is tested by an automated marker, so you must be careful to input only what is asked for, and output only what is asked for. Note, input is not from file and output is not to file.

Do not print prompts or messages like `The answer is:` that are not part of the problem's output specification.

In some questions, you may be asked to give output that is correct to a particular number of decimal places. It is recommended that you use the following code fragments to produce correct output for such questions (in all the cases below, we assume that the number of decimal places asked for is 6, and that the variable x is the value to be output).

Example 1. Floating-point output in Java

```
System.out.println(String.format("Case #%d: %.6f", caseNum, x));
/* or */
System.out.printf("Case #%d: %.6f", caseNum, x);
```

Example 2. Floating-point output in C or C++ with stdio

```
printf("Case #%d: %.6f\n", caseNum, x);
```

Example 3. Floating-point output in C++ with iostreams

```
cout.precision(6);
cout << "Case #" << caseNum << ": " << fixed << x << endl;
```

Example 4. Floating-point output in Python 2.x

```
print "Case #%d: %.6f" % (caseNum, x)
```

Example 5. Floating-point output in Python 3.x

```
print("Case #{}: {:.6f}".format(caseNum, x))
```

Warning: Your program will generally be judged against input that is different from the sample input shown in the problem statement. Your solution must be able to handle all valid inputs.

5 Submitting your solution

When you are ready to submit a solution, go to the Submissions tab and click Make Submission. Be very careful to select the right problem, source file and language in the dialog box, as there are penalties for incorrect submissions and there is no way to reverse an accidentally submitted solution.

Once you make a submission, it will appear in the list box, with newest submissions at the top. The status field will let you keep track of the submission status. The possible options are:

`Pending`	The submission is still in the automatic marker.
`Compilation failure`	Your submission did not compile. This can happen with C++ in particular, because the marker might not be using the same version of the compiler. To assist you, you can double-click on the submission to get the compiler output. In many cases you just need to add a `#include` statement. There is no time penalty for compilation failures.
`Time limit exceeded`	Your program exceeding the maximum time limit. This could mean that it is too slow or has an infinite loop.
`Abnormal termination of program`	Your program has crashed for some reason. Possible reasons include using too much memory, writing an excessive amount of output, throwing an uncaught Java exception, using illegal memory in C/C++, and not returning zero as the exit code from C/C++.
`Awaiting judge`	The program produced an output file that was judged wrong by the automatic marker. A judge will look at it, and if the only error is in formatting it may be judged correct. You should not rely on this.
`Wrong answer`	A judge has confirmed that the answer is wrong.
`Correct answer`	Well done!

Judges report the first condition they notice. In particular, `Format error` does not indicate whether the answers are correct or not, and `Wrong answer` does not indicate whether the format is correct or not.

If you wish to ask a question about a particular submission, you should submit a clarification request and mention the Submission ID in your request. A quick way to do this is to click the Request Clarification button next to the submission. This will open a clarification request window with the correct problem selected, and the Submission ID entered in the request text. Even if you include a submission ID, you still need to ask a specific question.

6   Standings

You can see the standings at any time (from the standings tab), but they will not be updated during the last hour of the contest. Only teams that have made at least one submission will appear on the standings.

7   Alerts

When a hidden tab has new information (such as a reply to a clarification, or the result of a submission), the tab title will light up red. Click on the tab to see the update.

Updated by Linda Marshall (lmarshall@cs.up.ac.za) on 18 October 2017