FAQ

1. For the format of the .tst file we will have the input on one line separated by spaces like so " 1 2 3". Is the next line the expected output in the same format or is it different.

=> My notes say we agreed to have a blank line between the input and the output but upon further reflection and consultation with product owners, I would like the following structure.

case1.tst // sample file name

1 2 3

case1.ans // answer for case1.tst

First quadrant

Rationale: It will be less work for you to do a diff of the resulting file if you don’t have to parse the input out of the .tst file. You will thank me for this change.

Other than that, I don’t think there is anything we need to specify about the program input format. So if the program reads all on one line or reads one number per line it should still work.

case1.tst // sample file name

1

2

3

case1.ans // answer for case1.tst

First quadrant

1. Is the student source code guaranteed to be a valid .cpp program file?

* Yes, in that it will compile and run without causing a runtime error. But the answers may be incorrect. That’s what you are looking for.

1. All input can be redirected. So, in Linux, the program can be tested by:

quadratic < case1.tst > case1.out

The results can be compared with case1.ans to see if it worked or not and the result logged in quadratic.log. A summary of results can be stored in the log file as well.

There is no user interaction in this system (you don’t type anything in from the keyboard. Side note: I have changed all the programs to put prompts to standard error so they don’t appear in the output file. Again, you will thank me for that.

1. From my understanding, there will be no menus or in-program command line input, just arguments passed to start the program. Is this correct?

* Correct – no user input from the keyboard. Just values on a file.

1. Is it true that if a program needs an input from a file, the .tst file will specify that? If so, what would the syntax of that be? If not, how should we know?

* All test files are called case#.tst and the accompanying answer for that test case is case#.ans (for answer).

1. Could you review the user story for this product (from your perspective)?

* A student submits a program. That program is placed into a directory that forms the root of the directory tree related to that program. All the instructors and TAs will have the ability to write test cases (called case#.tst and the accompanying file case#.ans). There are no restrictions on where those files can be located except that they will be at the level of the .cpp file or below. For example. Manes might have a subdirectory where he puts his test cases. His TAs might have subdirectories under the Manes subdirectory where they put their test cases. Manes might just make the directory but not put any test cases in it just so his TAs have a place to put theirs (in subdirectories). I want to say

test quadratic

and have your program find all the applicable test cases (with the accompanying answer files), run the tests, log the results, and provide a summary. I want to be able to fix problems and rerun the test without losing the original log file, by the way (a question no one has asked yet). Append the date so I can tell them apart.