

Lab 6

Loop practice

3 October 2019

Lab this week is getting more practice with loops, as we get ready for the exam.

For the string-related ones, remember that the book mentions a variety of useful string operators and functions.

In all the problems on this lab, don't prompt the user (it'll break my test cases).

- a. Reads in two positive integers (the first less than the second) and prints out all the even numbers between them (including the numbers themselves, if they're even), each on their own line. So if the inputs were 6 and 10, the program would print 6, 8, and 10. (Hint: what's the remainder when you divide an even number by 2?)

This problem is on repl.it as Lab 6 a.

- b. Reads numbers into a `vector<double>` and then prints out all the numbers that are within 0.5 of the largest number, in the order they were inputted. (So if the inputs were 2.0, 1.5, 2.6, 2.2, 1.9, 0.8, it would print out 2.6 and 2.2.) Note that this requires *at least two* loops—one to read in the numbers and keep track of the highest, and a second to go through and print the ones you want. You might even prefer three (one to read, one to find the highest, and one to filter which ones you want). (For this problem, do not use built-in functions to compute the largest value.)

This problem is on repl.it as Lab 6 b. You can assume that all the inputs end with a non-numeric line (which will cause your input loop to properly terminate).

For the rest of the lab period, work on writing programs that accomplish the following tasks. Don't forget to test that they work correctly.

- c. Reads a single "word", and examines each character of the input one-by-one and prints them back out unchanged except that if it's an underscore, your program prints a space instead.

This problem is on repl.it as Lab 6 c. Because of the nature of the problem, the test cases are much stricter than usual about whitespace!

- d. Reads in “words” (as delimited by whitespace, until it runs out of words), and prints how many of them are 5 characters long or longer.

This problem is *not* on repl.it, so you’ll have to work on it on the server. Think about how you might test it. (Remember that if you run it without using the .in and .expect files, pressing Ctrl-D indicates end-of-input.)

- e. Reads in non-negative integers (until it runs out), and then prints the proportion of the total that each input represents, one per line. (For instance, if one of the inputs were 5 and the total of all the inputs were 20, that line would print 0.25.)

This problem is on repl.it as Lab 6 e. You can assume that all the inputs end with a non-numeric line (which will cause your input loop to properly terminate).

- f. Reads in “words”, and prints them back out each separated by a space EXCEPT that if one word is the string “/*”, stop printing out words until you read a “*/”. For instance, if the input were

```
testing the output /* and what shouldn't print */ today
```

the expected output would be

```
testing the output today
```

This is a bit like what goes on in the compiler when it ignores your comments.

This problem is *not* on repl.it, so you’ll have to work on it on the server. It’s also a little tricky—how do you know whether you’re “still inside” a non-printing area or not?—but I’ll let you think about how to solve that.

No handin

I don’t need you to run the `handin` command this week; the purpose of this was to give you extra practice with loops before the exam.