

CSci 1113: Introduction to C/C++
Programming for Scientists and Engineers
Homework 3
Spring 2020

Due Date: Friday, February 28, 2020 before 11:55pm.

Instructions: This is an individual homework assignment. There are two problem worth 20 points each. Solve the problem below by yourself (unlike the labs, where you work collaboratively), and submit the solution as a C++ source code file. Here are a few more important details:

1. Unlike the computer lab exercises, this **is not** a collaborative assignment. You must design, implement, and test the solution to each problem on your own without the assistance of anyone other than the course instructor or TAs. In addition, you may not include solutions or portions of solutions obtained from any source other than those provided in class: examples from the textbook, lectures, or code you and your partner write to solve lab problems. Otherwise obtaining or providing solutions to any homework problems for this class is considered academic misconduct. See the “collaboration rules” file on the class website page for more details, and ask the instructor if you have questions.
2. Because all homework assignments are submitted and tested electronically, the following are important:
 - You follow the naming conventions mentioned at the end of the problems.
 - You submit the correct file(s) on gradescope (<https://www.gradescope.com/>) by the due deadline.
 - You follow the example input and output formats exactly given in each problem description.
 - **Regardless of how or where you develop your solutions, your programs compile and execute on gradescope computers (which run Linux/Ubuntu operating system like the cselabs machines).**
3. The problem descriptions will usually show at least one test case and the resulting correct output. However, you should test your program on other test cases (that you make up) as well. Making up good test cases is a valuable programming skill, and is part of ensuring your code solution is correct.

Problem A: Random vertical print (20 points)

Write a program that asks the user to input some letters to be vertically displayed. Put a zero at the end to signify the end of the letters to be printed (much like the '#' in HW2B). You should skip white space, such as tabs, spaces or enters (see example 2).

You should then print the input text vertically on the 10th character (index 9 or in the middle) of each row, and the rest of the row should be filled with a random character chosen from {'0', '1', '.' or ' ' (space)}. There should be a total of 19 characters on each line (including the actual words, so it is in the center). You may assume they user always puts a zero at the end of their desired text.

I suggest you break this program up into various parts (i.e. functions). To make my code answer to this problem, I made these functions:

```
char randomChar(); // returns either '0', '1' '.' or ' ' at random
void printLine(char c, int spot, int len); //prints a line of random characters of length='len',
// except for 'c' is always at 'spot'.
```

(Hint: Use the buffer and only cin a char variable.)

(Reminder: this line should be run once and only once in your code):

```
srand(time(0)); // you may need to say: srand(time(NULL));
```

Example 1 (user input is underlined):

What do you want printed vertically?

hello0

```
00 0110 1h0.00 1101
00  .1  e0 .111000
0.00 .10011 10001.0
.1 1 0011 .110.101
.00 .1. .o0.0 10 11
1000 . 101 0 111
```

Example 2 (user input is underlined):

What do you want printed vertically?

LIKE A BOSS0

```
0.. 0. .0L 1100
 1 0000 I101111 0
100 1 .0.K0. . 0101
01.1010 1E.000 1. 1
. 0000 A11 1.1110
0.0. 1.B0 101 1
00.011 .00.10. 001.
1100 0.00S1 110 1 0
00 .110..S 00011.0
. .011 0 01 .0100.
```

Example 3 (user input is underlined):

What do you want printed vertically?

0

```
000.01 0100 0. 1
```

When you are done, name the source code file hw3A.cpp. Then log into gradescope and upload your file for the “Homework 3A” submission. **If you name your file incorrectly it will be unable to compile and run your code, so you will fail all test cases.** You may submit cpp files as many times as you want until the deadline to try and fix the code if you fail a test case. Following rigorous naming conventions and using test cases are something computer programmers often must do in “real life” programming, and so submitting your program with the correct name and functionality is part of doing this assignment correctly.

Problem B: Days until summer (20 points)

Write a program that lets you enter two dates: a start date and end date. Then find the number of days between them. You may assume only valid dates are entered, but you should account for the case when the start date is after the end date.

(Hint: I'd suggest continuing what you have from lab, rather than trying a mathematical approach. Then adding either 3 new functions or 1 call-by-reference function if you did the last part in main().)

Example 1 (user input underlined):

```
Enter start date (no spaces):  
2/27/2020  
Enter end date (no spaces):  
5/12/2020  
You have to wait 75 days
```

Example 2 (user input underlined):

```
Enter start date (no spaces):  
2/28/2020  
Enter end date (no spaces):  
3/8/2020  
You have to wait 9 days
```

Example 3 (user input underlined):

```
Enter start date (no spaces):  
1/2/3  
Enter end date (no spaces):  
3/21/12345  
You have to wait 4507901 days
```

Example 4 (user input underlined):

```
Enter start date (no spaces):  
1/2/3  
Enter end date (no spaces):  
3/2/1  
Never...
```

When you are done, name the source code file hw3B.cpp. Then log into gradescope and upload your file for the "Homework 3B" submission. **If you name your file incorrectly it will be unable to compile and run your code, so you will fail all test cases.** You may submit cpp files as many times as you want until the deadline to try and fix the code if you fail a test case. Following rigorous naming conventions and using test cases are something computer programmers often must do in "real life" programming, and so submitting your program with the correct name and functionality is part of doing this assignment correctly.