Homework - C Loops – Shapes

This is a programming assignment that will be graded and scored as homework since it is so simple.

Write two functions in C as described below. These functions should be in a file named ***hwcl.cc.*** Make sure the file compiles and runs on the student machine before submitting.

Do not create any helper functions. You must ONLY have the two functions listed below.

You are ONLY allowed to include stdio.h. All other libraries will cause the test to fail.

YOU MUST USE PRINTF. DO NOT USE ANY C++ ESPECIALLY COUT or BOOL.

1. Create a function named ***printTriangleWhile***. The function should take a single integer as a parameter and print a triangle consisting of pluses, dashes, and spaces as shown below to the screen using only W***HILE*** loops. You are not allowed to use any ***FOR*** loops for this portion of the assignment (you can use if statements). If a value less than 1 is passed as an argument, the function should not print anything.

   The triangle should be printed with only dashes (minus), plus signs, spaces, and newlines. Each line should end with a newline. **There should be no extra spaces or extra characters anywhere not specifically needed.**

   For example:

   ```
   printTriangleWhile(7);
   ```

   should print a seven-level triangle as shown below (note the height and width are both 7)

   ```
   +++++++
    +----+
     +---+
      +--+
       +-+
        ++
         +
   ```

   Note that **NONE** of the rows should have spaces or tabs to the right AND the top row should have **NO** spaces to the left.

2. Create a function named ***printTriangleFor***. The function should take an integer as a parameter for height and print a triangle of plusses and dashes as shown below to the screen using only ***FOR*** loops. You are not allowed to use any ***WHILE*** loops for this portion of the assignment (you can use if statements). If a value less than 1 is passed as an argument, the function should not print anything.

The triangle should be printed with only spaces, dashes (minus), plus signs, and newlines. Each line should end with a newline. There should be spaces before the first plus sign and no spaces after the last plus sign on each line. There should be no extra characters anywhere else.

For example:

```
printTriangleFor(7);
```

should print a seven-level triangle as shown below.

```
      +
     +-+
    +---+
   +-----+
  +-------+
 +---------+
+++++++++++++
```

Note that **NONE** of the rows should have spaces or tabs to the right AND the bottom row should have **NO** spaces to the left.

3. Compiling and testing
    a. Make a file named hwcl.cc with the functions listed above.
    b. **Make a NEW FILE named main.cc** with a main method which calls the above functions. You will need to put prototype functions into main.cc. A sample main.cc is given below. Use main for testing. Put negative numbers and zeros as arguments and make sure your program follows the instructions above exactly.
    c. Compile your program using the following command:
        g++ -Werror -Wall -o hwcl hwcl.cc main.cc
    d. To run your program, use the following command
        ./hwcl

4. Submitting to Web-CAT
    a. Make sure hwcl.cc DOES NOT CONTAIN A MAIN METHOD!
    b. Compile and run your program on the student machine first.
    c. You MUST remove all warnings.
    d. Submit **ONLY** hwcl.cc to Web-CAT. A link is provided on AsULearn.
    e. You can submit up to two days late. You will lose 10 points per day late. Days start at the original due date and time. After two days you will receive a zero for the homework.
    f. You will be penalized 1 point for every submission over 5. You should be testing your programs thoroughly before submitting.

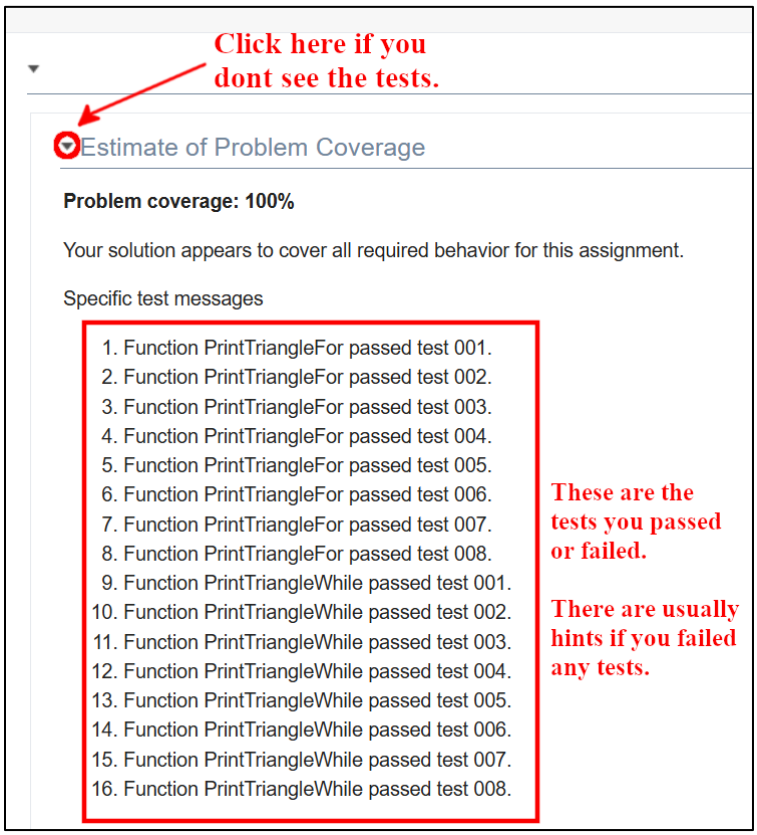
```
#include <stdio.h>


//Sample main.cc

//Prototypes from hwcl.cc are required so that
//the compiler knows parameters and return
//types for those functions at compile time.
void printTriangleWhile(int);
void printTriangleFor(int);


//USE OTHER VALUES OF WIDTH AND HEIGHT FOR TESTING.
//TEST YOUR FUNCTIONS THOROUGHLY.
//MAKE SURE YOUR FUNCTIONS WORK CORRECTLY
//FOR ODD VALUES (0, 1, -4, etc...)
int main()
{
      printTriangleWhile(7);
      printTriangleFor(7);
}
```

Running the above on the class server should look **EXACTLY** like the following. Note there are NO blank lines anywhere in the output.

```
swansonja@cs: ./hwcl
+++++++
 +-----+
  +---+
   +--+
    +-+
     ++
      +
      +
     +-+
    +---+
   +-----+
  +-------+
 +---------+
+++++++++++++
swansonja@cs:
```

A NOTE ON WEBCAT and TEST FEEDBACK.

When Web-CAT has finished grading your submission, you will get a screen that looks something the following:



If you click on the triangle pointed to by the arrow above, you will be shown the output message for each test which may help you troubleshoot your program.

IF YOU SEE A GRADE OF 0% even though all your tests appear to have passed, make sure all the tests have completed. There are 16 tests for this homework. If you see fewer than 16 test results in Web-CAT, it means one of the tests encountered a runtime error with your code. Your code caused a fault in the testing system that was so bad it caused the whole testing system to quit working. And since the system exited prematurely, it was unable to assign a grade. Make sure to thoroughly test your program with all types of input. Make sure you pass your functions values that could cause issues like negative numbers, 0, 1, and 2. Often a value of 0 or 1 sends student code into an infinite loop. Which will break the testing system.

Some Tips on Troubleshooting with Whitespace (tabs, spaces, and newlines)

Printing invisible characters, like spaces, tabs, and newlines, in the wrong place can cause problems with the tests.

But you can't see invisible characters so how do you know if you are printing them in the wrong place?

A tip for troubleshooting these invisible characters is to print something at the same time as a newline or replace tabs and spaces with a character so you can see where you are printing. This is most important to check and see if you are printing extra non-visible characters at the end of the line.

For newlines, print something just before the newline. Leave the newline in place. It it will get very confusing and impossible to decipher if everything prints on a single line.

Replace `printf("%d\n", i);` with `printf("%d**X**\n", i);`

And you will see the X character at the end of a line just before a newline.

For spaces, replace the space with a O to see where spaces are being printed.

Replace `printf(" ");` with `printf("O");`

***** !!!MAKE SURE TO CHANGE THESE BACK BEFORE SUBMITTING!!! *****

Running the main given above should result in the following output where you can see exactly where the spaces and newlines will appear.

```
[swansonja@cs ~]: ./hwcl
+++++++X
O+----+X
OO+---+X
OOO+--+X
OOOO+-+X
OOOOO++X
OOOOOO+X
OOOOOO+X
OOOOO+-+X
OOOO+---+X
OOO+-----+X
OO+-------+X
O+---------+X
++++++++++++X
[swansonja@cs ~]:
```

***** !!!MAKE SURE TO CHANGE THESE BACK BEFORE SUBMITTING!!! *****