

CSI 1420 Introduction to C Programming & Unix

Fall 2018, CRN 44882, Oakland University

Homework Assignment #5 - Using Repetition Statements in C

Due: Sunday, November 18, 2018

Total Grade Points: 100

Description

This assignment consist of writing a program that will demonstrate your ability to write repetition/loop structures (while/do-while/for) in a C program. Successful completion of the homework will require use of both counter-controlled as well as sentinel-controlled iteration structures.

You will write a menu-driven program that can draw a triangle shape using character symbols. Details on the structure of the triangle pattern and the program menu is provided in the following:

The Triangle shape

- Each triangle will be drawn using any one of these eight character symbols ! @ # \$ % ^ & *
- A triangle of size **N** would be printed using exactly **N** number of lines. The first line will consist of an appropriate number of blank/space characters, followed by one character symbol. Similarly, the second line will consist of two character symbols and appropriate number of blank characters; and so on. The base of the triangle will contain exactly **N** symbols.
- Any two consecutive symbols on the same line will be separated by a single space/blank character.
- The minimum size of a triangle is 2. The maximum size of a triangle is 40.
- The base of the triangle will start with a symbol at the beginning of that line. There will be no space 'offset' to the left of the first symbol of the base.
- Some example triangle patterns are given below:

(Triangle of size 2, drawn using '*')

```
*  
* *
```

(Triangle of size 10, drawn using '^')

```
  ^  
  ^ ^  
 ^ ^ ^  
^ ^ ^ ^  
^ ^ ^ ^ ^  
^ ^ ^ ^ ^ ^  
^ ^ ^ ^ ^ ^ ^  
^ ^ ^ ^ ^ ^ ^ ^  
^ ^ ^ ^ ^ ^ ^ ^ ^  
^ ^ ^ ^ ^ ^ ^ ^ ^ ^
```

(Triangle of size 4, drawn using '&')

```
  &
 & &
& & &
& & & &
```

The Program Menu

1. At the beginning of the program the user will be asked to enter the size and the character symbol to draw a triangle. If the user enters an incorrect numeric value as the size, or an incorrect character value as the drawing symbol, Your program will print an error message, and then, draw a 'default' triangle of size 5 using the character '*'. Otherwise, your program will draw a triangle of the user-input size and symbol
2. After drawing the first triangle, the user is provided with the following menu with four options:
 - G: Grow the current triangle
 - S: Shrink the current triangle
 - N: Draw a new triangle
 - Q: Quit
3. Upon entering the character 'G' or 'g' at the menu prompt, your program will increment the current size of the triangle by 1 and redraw it. If the current size is already at the maximum of 40, then choosing this option will have no effect and the current rectangle will be redrawn.
4. Upon entering the character 'S' or 's' at the menu prompt, your program will decrement the current size of the triangle by 1 and redraw it. If the current size is already at the minimum of 2, then choosing this option will have no effect and the current rectangle will be redrawn.
5. Upon entering 'N' or 'n' at the menu prompt, the program will repeat the operations at step 1 and draw a new triangle based on the user-input size and symbol (or a default triangle for incorrect input)
6. Each time after re-drawing a triangle, the user will be presented with the same menu shown in Step 2 and re-prompted for input of the user's choice.
7. Upon entering 'Q' or 'q' at the menu prompt, your program will terminate.
8. For any other character input, your program will print an error message and keep re-prompting user with the four options until a valid option is entered.
9. You may apply any kind of output formatting of your choice while printing the menu, prompts and error messages. However, the user input must be processed in the exact way as described above.

Assumptions:

- We assume that the user may enter an input of incorrect value, but will not enter an input of incorrect type.

Submission instructions:

You will only turn in **ONE** file: that is, the source code of your program. DO NOT submit your program executable file. Your source code will have the name "**hw5_triangle.c**". Make sure that the file containing your source code compiles successfully at the SECS Linux servers without any compilation errors or warnings. Be sure to test the correctness of your program by running the program several times with different user inputs and see if the output matches your hand-calculated results.

At the beginning of your source code, you will need to provide a comment block of the following format

```
/* Name: <Your Name>
Assignment: <This homework assignment #>
Due Date: <Due Date>
Last Modified: <date >
```

```
About this project: <In your own words, write a small paragraph about what
this project's goals are/what tasks it accomplishes>
```

```
Known Issues: <Mention if there is any incompleteness or unfixed bug in
your program>
```

```
*/
```

You must use the same above format in **ALL** subsequent programming assignment submissions. In addition to the comment header block as shown above, your source code must contain sufficient amount of comments, explaining each non-trivial step of your source code.

Rules/Guidelines:

- You may not use global variables to complete this assignment
- Each compilation warning in your program will cause 5% penalty to your total grade; and may be accumulated up to 20% of your grade.
- A submitted program that does not compile will incur heavy penalty worth up to 50% of your total grade
- Throughout your program, you should follow appropriate good programming practices. Failing to maintain consistency and recommended style in coding will result in penalty to your grade.
 - Identifiers(a.k.a variable names) should be meaningful and descriptive, but not too long
 - Comments in the same section or on consecutive lines should be aligned with each other(start from the same cursor position)
- Separate different "sections" of code with blank lines
- All content inside a **scope**, bounded by a pair of curly braces, { and }, should always be indented by a few spaces to the right. This maintains the structure and readability of your program. It also helps to find program bugs caused by mismatched/missing braces