CS215: Introduction to Program Design, Abstraction and Problem Solving
(Spring, 2023)
Lab Assignment 4
(20 points)

Today's Date: Sunday, February 5

Demonstration Due Date: the end of Lab5 class

Submission Due Date: Friday, February 17

The purpose of this lab assignment is

- to practice using nested loop structure
- to practice creating functions
- to practice the user input validation

Problem Statement

Write a program that asks the user to input an integer in the range [1, 30] as the size n of a triangle, which is the number of lines in the triangle. Based on the size, you will create a function named **drawTri()** to print a triangle which contains one little star (asterisk symbol) at the first line, three little stars at the second line, and so on till **2*n-1** little stars at the nth line, and it is symmetric. For example, the triangle with size 5 are shown in the following:

```
    *
   ***
  *****
 *******
*********
```

Create two more functions named **drawTriR90()**, and **drawTriR180()** that will draw TWO more triangles obtained after rotating the original ones 90 degrees and 180 degrees to the right, respectively. When size = 5, for instance, functions drawTri(), drawTriR90(), drawTriR180(), respectively, will print:

//original:

```
    *
   ***
  *****
 *******
*********
```

//after 90 degree rotation to the right

```
*
**
***
****
*****
****
***
**
*
```

//after 180 degree rotation to the right

```
*********
 *******
  *****
   ***
    *
```

Your program will

- repeatedly read the size of a triangle, which is in the range of [1, 30], till the user input Q (or q) to stop your program
- print the original triangle based on the user input size, and rotate it clockwise for 90 and 180 degrees, and print the rotated triangles respectively.

You need to define the following three functions in your program:

```
/*
 * print a triangle of asterisk, center alignment
 * starting with one asterisk at the first row, three at the second
row...
 *@param n int: representing how many rows of the triangle
 *        (The triangle should contain 2*n-1 asterisks at the bottom
row of the triangle)
 *@return: void function
*/
void drawTri(int n)
```

```
/*
 * print a triangle of asterisk
 * It rotates the pattern from drawTri function, 90-degree clockwise
 *@param n int: representing how many rows of the original triangle
before rotation
 *@return: void function
 */
void drawTriR90(int n)
```

```
/*
 * print a triangle of asterisk
 * It rotates the pattern from drawTri function, 180-degree clockwise
   or it rotates the pattern from drawTriR90 function, 90-degree
clockwise
 *@param n int: representing how many rows of the original triangle
before rotation
 *@return: void function
 */
void drawTriR180(int n)
```

The following shows TWO sample outputs of running your program:

Sample output 1:

```
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: size 5 is my choice↵
Invalid size!
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: Q↵
Thank you, have a great day!
```

Sample output 2:

```
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: five as the size↵
Invalid size!
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: 5 as the size↵
The triangle with size 5 is:
    *
   ***
  *****
 *******
*********
The rotation for 90 degrees clockwise:
*
**
***
****
*****
****
***
**
*
```

```
The rotation for 180 degrees clockwise:
*********
 *******
  *****
   ***
    *
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: 31 is that ok?↵
The size is not in the correct range!
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: -5 is my guess↵
The size is not in the correct range!
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: 0↵
The size is not in the correct range!
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: 1↵
The triangle with size 1 is:
*
The rotation for 90 degrees clockwise:
*

The rotation for 180 degrees clockwise:
*
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: 7.5 is the size↵
The triangle with size 7 is:
      *
     ***
    *****
   *******
  *********
 ***********
*************
The rotation for 90 degrees clockwise:
*
**
***
****
*****
******
*******
******
*****
****
***
**
```

```
*

The rotation for 180 degrees clockwise:
*************
 ***********
  *********
   *******
    *****
     ***
      *
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: 13 thirteen↵
The triangle with size 13 is:
            *
           ***
          *****
         *******
        *********
       ***********
      *************
     ***************
    *****************
   *******************
  *********************
 ***********************
*************************
The rotation for 90 degrees clockwise:
*
**
***
****
*****
******
*******
********
*********
**********
***********
************
*************
************
***********
**********
*********
********
*******
******
```

```
*****
****
***
**
*

The rotation for 180 degrees clockwise:
*************************
 ***********************
  *********************
   *******************
    *****************
     ***************
      *************
       ***********
        *********
         *******
          *****
           ***
            *
Enter the size of your triangle (integer in [1, 30])
Type Q to quit the program: q↵
Thank you, have a great day!
```

Note that the blue part represents the user input, and “↵” represents the return key from user input.

Demonstration and Submission

1. Each Lab assignment needs to demonstrate to your TA to be graded. You can demonstrate Lab4 during Lab4 class (with possible bonus 3 points) or no later than the end of Lab5 class (this is **the demonstration deadline** for Lab4).
If you finish Lab4 assignment during Lab4 class, you may demonstrate your program to your TA and answer your TA's questions, you can get up to 3 extra points for this lab assignment. (Note you can also demonstrate your program to your TA during Lab5 class. However, any demonstration later than the end of the Lab4 class cannot get bonus 3 points.)
If you need extra time, you can continue working on Lab4 assignment after the Lab class, and try to finish it before the next Lab class. Then demonstrate your Lab4 during Lab5 class.
If you do not demonstrate your code, even if you submit it in Canvas, you will receive a grade of 0!! *The TA may ask you to make some corrections. If so, make the corrections and demonstrate again…repeat until you have 100%!*
2. After the successful demonstration, submit the code in Canvas. Open the link to Course Canvas page (https://www.uky.edu/canvas), and log in to your account using your LinkBlue ID and password. Please submit your **source code in a .cpp file** through link “**Lab 4**”.

Even if you successfully demonstrated it to the TA, if you do not submit in Canvas by the submission deadline, you will receive a grade of 0!

Grading (20 points + Bonus 3 points)

1. Attend the lab session or have a documented excused absence. (5 points)
2. Demonstrate your program to your TA and submit it in Canvas. (15 points)
   - Include comments as specified in the lecture notes. (3 points)
   - Provide the correct definition of three functions. (each 3 points, total 9 points)
   - User-friendly Input/Output design, and user input validation. (3 points)

Demonstrate your program to your TA and answer TA's questions during Lab class when the same Lab assignment is given. (Bonus 3 points)