CSCI 1300 - Intro to Computer Programming
Instructor: Fleming
Recitation 4

This assignment is due **Friday, September 28th, by 11:55 pm**

- ***All components*** **(Cloud9 workspace and moodle quiz attempts) must be completed and submitted by Friday, September 28th 11:55 pm for your solution to receive points.**
- **Recitation attendance is required to receive credit.**

Please follow the same submission guidelines outlined in Homework 3 description regarding Style, Comments, and Test Cases. Here's a review below on what you need to submit for Recitation 4.

Develop in Cloud9: For this recitation assignment, write and test your solution using Cloud9.
Submission: All three steps must be fully completed by the submission deadline for your homework to receive points. Partial submissions will not be graded.

1. ***Make sure your Cloud 9 workspace is shared with your TA:*** Your recitation TA will review your code by going to your Cloud9 workspace. *TAs will check the last version that was saved before the submission deadline*.
    - Create a directory called **Rec4** and place all your file(s) for this assignment in this directory.
    - Make sure to *save* the final version of your code (File > Save). Verify that this version displays correctly by going to File > File Version History.
    - The file(s) should have all of your functions, test cases for the functions in main function(s), and adhere to the style guide. Please read the **Test Cases** and **Style and Comments** sections included in the **Homework 3** write up for more details.
2. ***Submit to the Moodle Autograder:*** Head over to Moodle to the link **Rec 4**. You will find one programming quiz question for each problem in the assignment. Submit your solution for the first problem and press the Check button. You will see a report on how your solution passed the tests, and the resulting score for the first problem. Continue with the rest of the problems.

New this week: You can modify your code and re-submit (press *Check* again) **only 15 times** for each problem, up until the assignment due date.

CSCI 1300 - Intro to Computer Programming
Instructor: Fleming
Recitation 4

While Loops

Loops allow us to run a section of code multiple times. They will repeat execution of a single statement or group of statements as long as a specified condition continues to be satisfied. If the condition is not true, then the statement will not be executed.

Syntax and Form:

```
while (condition)
{
 //statement to do something;
}
```

where *while* is a C++ reserved word, *condition* is a boolean-expression that will evaluate to a true or false statement, and *statement to do something* is enclosed by curly brackets. If the condition is true, the specified statement within the loop is executed. After running once, the boolean-expression is then re-evaluated. If the statement is true, then it is executed again. This process of evaluation and execution is repeated until the condition becomes false.

Example 1:

```
int userChoice = 1;
while (userChoice != 0)
{
    cout << "Do you want to see this question again? " << endl;
    cout << "Press 0 for no, any other number for yes." << endl;
    cin >> userChoice;
}
```

Entering ‘0’ will terminate the loop, but any other number will cause the loop to run again. Note how we have to initialize the condition before the loop starts. Setting *userChoice* equal to 1 ensures that the while loop will run at least once.

Example 2:

```
int i = 0;
while (i < 5)
{
    cout<< i << endl;
    i = i + 2;
}
```

Notice how you must manually initialize *i* to equal 0 and then manually increment i by 2.

CSCI 1300 - Intro to Computer Programming
Instructor: Fleming
Recitation 4

Inserting print statements into your loops is a quick way to debug your code if something isn't working.

Problems Set:

Write a **function** for each of the following problems. You should first write your solution in Cloud9. Then copy and paste your function into the moodle quiz questions.

Problem 1

Write a function to print all positive even integers less than or equal to a max value.

- Your function MUST be named **printEvenNums**
- Your function should take one parameter:
  - an integer parameter representing the max value.
- Your function should not return anything.
- Your function should print the positive even integers less than or equal to the max value (see expected output format below)

If a user calls printEvenNums with the max=10, the expected output would be:

```
2
4
6
8
10
```

Problem 2

Write a function to print all positive multiples of an integer number less than or equal to a max value.

- Your function MUST be named **printMultiples**
- Your function should take two parameters in this order:
  - an integer parameter
  - an integer parameter representing the max value
- Your function should not return anything
- Your function should print all the positive multiples of the first integer parameter less than or equal to the max value (see expected output format below)

If a user calls printMultiples with 3 and max=10 the expected output would be:

```
3
6
9
```

CSCI 1300 - Intro to Computer Programming
Instructor: Fleming
Recitation 4

Problem 3

Write a function to print a filled and hollow square of given width side by side to the screen. The two squares should be separated by a single space.

- Your function MUST be named **printSquares**
- Your function should take a single parameter:
    - an integer representing the width of a single square
- Your function should not return anything
- Your function should print two squares, one filled and one hollow, side by side separated by a single space

A call to printSquares with the input 5 should result in the following output:

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

Note that the whitespace between "*" for the hollow square is the width of 3 "*" characters (3 spaces).

Problem 4

Write a function to print a diamond with given side length to the screen.

- Your function MUST be named **printDiamond**
- Your function should take a single parameter:
    - an integer representing the side length of the diamond
- Your function should not return anything
- Your function should print a diamond with given side length to the screen

A call to printDiamond with the input 4 should result in the following output:

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