**Rules**

1. Submission Steps:
   1. In your local machine, create a new folder; the name of the folder should be your 7 digit roll number.
   2. For each problem that you choose to solve, code your solution in a .c file. The name of the file should be exactly as is mentioned in problem description. Put this C file in the folder created in step 1(a).
   3. Finally, zip the folder created in step 1(a) to produce a .zip file. The name of the .zip file should be your 7 digit roll number.
   4. Submit the .zip file created in step 1(c).
2. You must submit your solution package (.zip file) within the given deadline through Moodle.
3. You must bring your solution package in the next lab class.
4. You must be able to explain your code properly in the next lab class.
5. Do not copy code. You will be caught and given -20.

**Problem #1**

**Triangle or not**

**File name: triangle.c**

Write a C function that takes as parameters three floating point numbers which indicate the lengths of a triangle. The purpose of this function is to

* Determine if such a triangle is possible
* Determine if the triangle is right-angle or not

As we very well know, a function CAN NOT return more than one values. Therefore, to make this function perform both its purposes, take a fourth parameter, which is a pointer to an array of appropriate type. The function will perform these two checks and populate the entries of the array accordingly. In main function, after the call to the above function returns, you have to check these values in the array to finally understand if the triangle is possible, and/or it is right angle.

You are required to

* Write a main function that takes these three float numbers as input
* Perform ONLY ONE call to this function that you have written
* After the call, you have to print if the triangle is possible or not
* Next, without any further call to any function, you have to print if this triangle is right angle or not

**Problem #2**

**Substring**

***File name: substring.c***

Write a C program that will take two strings S1 and S2 as 2 lines of input. The program will print how many times S2 appears in S1.

|  |  |
| --- | --- |
| **Sample Input(s)** | **Corresponding Output(s)** |
| I am a boy. Very good boy.  boy | 2 |

**Problem #3**

**Symmetry**

***File name: symmetry.c***

Write a C program that will take an n x n matrix as input and determine if this matrix is symmetric or not. A matrix is symmetric when it the same as its transpose. (Check Wikipedia if you don’t know already what a transpose matrix is)

|  |  |
| --- | --- |
| **Sample Input(s)** | **Corresponding Output(s)** |
| 4  2 3 0 7  3 5 1 1  0 1 0 2  7 1 2 3 | YES |
| 3  1 2 3  4 5 6  7 8 9 | NO |