ESS 112: C Programming

Instructors: Profs. Yashavant Kanetkar and Srinivas Vivek

8th March, 2021

End-Sem Exam

Time: 2:30 PM – 4:00 PM

Instructions:

- You are not allowed to refer to any printed or written material nor any electronic media or the internet. Any interaction with any other person is prohibited. Any malpractice will be dealt with severely.
- You need to submit to BOTH **domjudge** AND **EXAM LMS**.
- Answer all the questions, there are a total of 5 questions.
- The *maximum marks* for this test is 80.
- Make sure to compile and run your program successfully for at least the sample inputs given in the question paper before submitting to domjudge. After submission to domjudge, it will show as pending evaluation. Evaluation will be done after the exam is completed.
- Name the programs as 1.c 2.c 3.c 4.c and 5.c for questions 1 through 5, respectively. Zip the files and name the zipped file IMT2020XYZ.zip where IMT2020XYZ is your roll number. Do **not** put the files inside a folder.
- Upload the zip file in EXAM LMS (https://exam.iiitb.net/) into the **End SEM EXAM** folder.
- Submit your answers into EXAM LMS at least 15 minutes before the stipulated deadline to avoid last minute bandwidth or other network issues. Late submissions will be **rejected**.
- **Note:** No clarifications will be entertained during the test.

Questions:

(10 marks)

1. Write a (C) program to display the `most` and the `least` frequent characters in the given input string.
Note: The maximum length of the input string will be 30 characters and will contain only lower case characters. The value of `least` ≤ `most`.
If there are multiple characters having the same frequency, print the character that appears first in the given input string.

Sample Input 1:
occurrence
Output 1:
c
o

Sample Input 2:
aaabbb
Output 2:
a
a

(10 marks)

2. Write a single `Calculator` function which takes two floating point numbers x, y and performs an arithmetic operation on them. It should be able to perform addition, subtraction, multiplication, division of two numbers, and you need to also pass another argument to the function which lets you choose which operation to perform on the two numbers (implementation of this is left to you). To perform any further arithmetic operations, you should use this calculator function only. So to add two numbers, you need to call your calculator function with suitable arguments to give you the result instead of directly using addition operator of C.
Note - If you use (C) arithmetic operators instead of calling calculator function, you will be awarded 0 marks. However you can use the `sqrt()` function of math.h with no penalty.

Create a Triangle structure which has the sides of the triangle as its members. Write a `Compute` function which takes a Triangle structure as argument and prints "Area" if the area of triangle is greater than its perimeter or else prints "Perimeter". This function should use the `Calculator` function as described above. Write a (C) program to input three real numbers and calls the `Compute` function appropriately.

Sample Input:
3.1 4.1 5.1
Sample Output:
Perimeter

Sample Input:
6.1 8.1 10.1
Sample Output:
Area

(15 marks)

3. A string is said to be a palindrome if it reads the same backwards as forward.
Write a (C) program that takes string s as input. Print the count of all possible substrings of s which are palindromes.
Note : You should only consider substrings of length greater than 1. Also a substring that reads the same may be counted multiple times if the same substring occurs in different positions in s.
For example, if input given is "IIITB" then output i.e., count will be 3 which includes the strings - "II", "III", "II")
Sample input 1:
MALAYALAM
Output:
6
Sample input 2:
MISSISSIPPI
Output:
9

(20 marks)

4. You are given description of two pentagons, write a (C) program to output whether they intersect or not. A pentagon is defined by the two integer arrays X, Y of length 5 denoting the X-coordinates and Y-coordinates of the vertices of the polygon. The i^{th} index entry in the arrays X, Y forms the point i^{th} vertex (x_i, y_i). The sides of the polygon are formed by consecutive vertices i.e. the i^{th} side is formed by the pair of vertices (x_i, y_i) and (x_{i+1}, y_{i+1}). The fifth side is formed from the vertices (x_5, y_5) and (x_1, y_1).
The first 2 lines of the input correspond to the X and Y-coordinates of the vertices of the first pentagon. The next 2 lines of input correspond to the second pentagon's vertices in similar format. Output `Yes` if the given pentagons intersect else, output `No`.

`Note:` Here two pentagons are said to intersect if there exists at least one pair of line segments (l, m) such that l is the side of first pentagon and m is the side of second pentagon and l, m intersect at one point.
It is always guaranteed that the given vertices form a pentagon. Each x_i and y_i is guaranteed to be in the range [-1000,1000]

Sample Input:

```
0 2 2 -2 -3
4 2 -3 -2 0
-6 -4 -4 -6 -7
3 2 -1 -2 1
```

Sample Output:

```
No
```

Sample Input:

```
3 5 7 6 2
4 6 2 -1 -2
0 2 2 -2 -3
4 2 -3 -2 0
```

Sample Output:

```
Yes
```

(25 marks)

5. You are given a map of islands and surrounding water bodies. Write a (C) program to count the total number of islands visible in the map. Map is a 2-D character array of n rows ($1 \leq n \leq 500$) and m columns ($1 \leq m \leq 500$), where each element of the array represents either Land element ('L') or Water element ('W'). All surrounding elements are considered to be the neighboring elements. For example an element at array index (i, j) has neighbors in the North (i-1, j), South (i+1, j), East (i, j+1), West (i, j-1), North-West (i-1, j-1), North-East (i-1, j+1), South-West (i+1, j-1), South-East (i+1, j+1). For corner elements there are only 3 neighbors and for edge elements there are only 5 neighbors. An Island is a set of land elements surrounded by water bodies and/or no elements. The first line of the input will be n and m (single space separated). Each of the following n lines will be a string of m characters ('L' or 'W'). Your program should print the total number of islands present in the map.
Hint: Think if there is a recursive way to traverse all the Land elements of an island.

Sample Input 1:

```
4 5
LLLWW
WWWWW
WWWLL
WWLLL
```

Output 1:

```
2
```

Sample Input 2:

```
8 8
WWLLLLWW
WLLLLLLW
LWWWWWWL
WWLLLWWW
WWWWWWWW
WWWWWLLL
WLLLLLLL
WWWWWWWL
```

Output 2:

```
3
```

Sample Input 3:

```
2 2
WW
WW
```

Output 3:

```
0
```

Sample Input 4:

```
3 5
LWWWL
WWLWW
LWWWL
```

Output 4:

```
5
```

Good luck!