National University of Computer and Emerging Sciences

FAST School of Computing **Fall-2022** **Islamabad Campus**

CL1002: Programming Fundamentals Lab

BS(CS) (A, B, D, K)

Friday 6th January 2023

Course Instructors

Muhammad Toqeer, Riva Malik, Sher Bano, Muhammad Usman

Serial No:

Final Exam

Total Time: 3 Hour

Paper attempt: 2:40 min

Submission: 20 min

Total Marks: 100

_____

Invigilator Signature

_____ _____

Student Name Roll No. Section Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

This is a Close Book Paper; you are not allowed to use Internet or any Gadgets including USB device.

Instructions:

1. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. Calculator sharing is strictly prohibited.
4. If found any kind of plagiarism in your code **F grade** will be awarded.
5. Submission will be opened before 20 minutes' end of the exam.
6. Put your all **.cpp** files in single folder name it Roll_no_Section_Final**(22i_6069_A_Final)** and zip that folder and name of the folder is same as mentioned.
7. Write anything you learnt in this lab (in Q1.cpp on top as comment not on paper) apart from studies at the end to get 3 bonus marks.
8. You have **7** pages in total and **4** question in total.

	Q-1	Q-2	Q-3	Q-4	Total
Marks Obtained					
Total Marks	**30**	**30**	**20**	**20**	**100**

Question 1 [30 Marks]

Image is stored as matrix having a value of 0 to 256. For further processing we must reduce the image size by the given formula considering the image and filter are square matrix: -

$$O = \frac{Image - filter}{2} + 1$$

O x O is the output image

Example input:
Input image size: 6 x 6
Filter size: 2 x 2

$$O = \frac{6-2}{2} + 1 = 3$$

Output image size: 3 x 3

Given a 6 x 6 image matrix you have iterate over the image using the 2 x 2 filter matrix. Consider the following rule for applying filter.

1. Place the filter matrix over image matrix.
2. Calculate the maximum element of the filter as **filter_max**
3. If **filter_max** is greater than the maximum value of the image matrix **img_max** in that filter space, then filter_max otherwise img_max will place in the 1st index of the resultant image

For example: -

33	**45**
11	**110**

Filter (2x2)

128	**197**	**80**	**88**	**134**	**0**
31	**176**	**62**	**76**	**14**	**60**
11	**145**	**32**	**89**	**11**	**07**
12	**167**	**48**	**89**	**31**	**08**
51	**18**	**255**	**67**	**13**	**60**
161	**19**	**76**	**67**	**41**	**40**

Image (6x6)

Output image (3x3)

Applying filter over image. Maximum element value will be placed in output image 1st index.

Image Matrix

Filter Matrix →

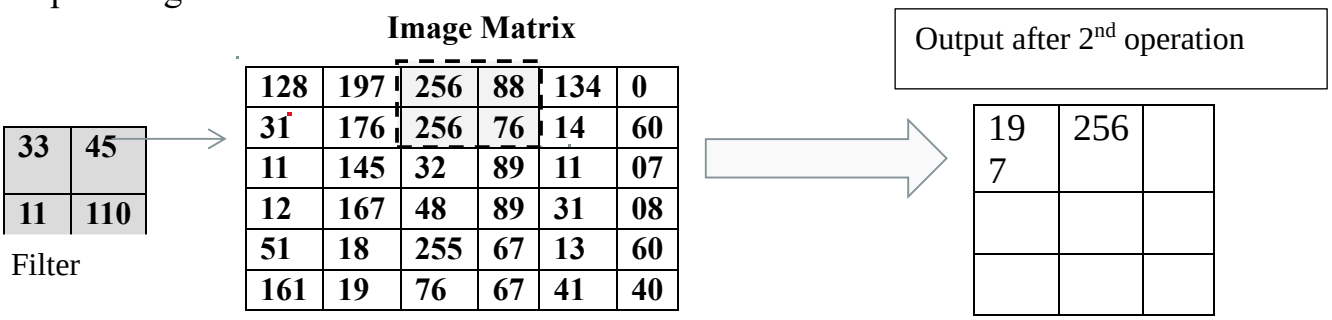
128	**197**	**256**	**88**	**134**	**0**
31	**176**	**256**	**76**	**14**	**60**
11	**145**	**32**	**89**	**11**	**07**
12	**167**	**48**	**89**	**31**	**08**
51	**18**	**255**	**67**	**13**	**60**
161	**19**	**76**	**67**	**41**	**40**

Output after one element operation

197		

The maximum value from 1st iteration of the filter is 197.

Moving filter to horizontally to jump to 3rd column of the image.
Applying filter over image. Maximum element value will be placed in the 2nd index of the output image



The maximum value from 2nd iteration of the filter is 256.

And so on.

Final output is: -

197	256	134
167	110	110
161	255	110

Rubrics	Marks
If program fully working with functions	30 marks(full)
If program fully working without functions	20 marks
If partial implementation is found (max filter and max image)	10 marks
If program not working but highlighted logic is there	20 marks

Question 2 [30 Marks]

You are given a list of words (a char array of words), that has multiple words. You must perform the following list of tasks: -

1. Remove the following words from that character array .
   string remove_words[] = {"is", "am", "of", "are", "the"};

   Example: -

   Input :- char corpus[ ]="hello i am toqeer how is the life? what is going in the life? life is very short.";
   Output:- char corpus[ ]="hello i toqeer how

Function Prototype: void remove_words (char arr[ ], string remove_words **[ ], int size=5)**

Size = 5 is for remove words array size.

2. Display the frequency of each word
   Example: -

   Input :- char corpus[ ]="hello i am toqeer how is the life? what is going in the life? life is very short.";

   Output:- hello: 1, i:2, am:1, toqeer:1, how:1,

   Function Prototype: void display_word_count(char words [ ])

3. Count the alphabets in the given corpus

Example: -

Input :- char corpus[ ]="hello i am toqeer how is the life? what is going in the life? life is very short.";

Output:- a: 2, b: 0, c: 0, d: 0, e: 9, f: 3, g: 2, h: 6,

Function Prototype: int *alphabets_count(char words [ ])

Hint:- use integer array of size 26, where zero index will count char a and 25th for letter z count (only consider small alphabets)

Rubrics	Marks
If program fully working with 3 functions	30 marks(full)
If program fully working without functions	20
If only one function	10 marks
If **display_word_count minor logic** and other complete functions	25 marks
If alphabets count is using if else and very naïve logic and other functions complete	20 marks
If logic is present but not workings	15 marks

Question 3 [20 Marks]

You have an array of int type. There is method call **mostFrequnentElementsInArray**(int *array, int size, int frequency). This method will take three parameters size, frequency and array size is the array of size and frequency is count of element. Your task is to find those elements in array which occur frequency time or more than frequency time.

Example

Arr[]=[2,2,4,2,4,5,6,l,1]

We call mostFrequnentElementsInArray(Arr,0,9,2) it will, return the following elements [2,4,1].

You have to use only pointers and dynamic memory allocation.

Rubrics	Marks
If program fully working with function with redundant elements checking and frequency match	20 marks(full)
If only duplicates finding and printing	5 marks
If only duplicates finding and storing without frequency count	10 marks
If program not working but highlighted logic is there	10 marks

Question 4[20 Marks]

Write a recursive function [int count(int n)] that takes in a non-negative integer 'n' and returns the number of ways to write 'n' as the sum of 1, 3, and 4.

For example, count(5) should return 3, because the possible ways to write 5 as the sum of 1, 3, and 4 are:

1 + 1 + 1 + 1 + 1
1 + 4
4 + 1

You may assume that the function will only be called with non-negative integers.

Rubrics	Marks
If program fully working with function with redundant elements checking and frequency match	20 marks(full)
If using loop	5 marks
If answer is not their but logic is their somehow	10 marks
If program not working but highlighted logic is there	10 marks