CS-1004 Object Oriented Programming Fall-2022

ASSIGNMENT-01

Sections (All), TOTAL MARKS = 170

Submission Deadline: SUNDAY (18-SEPT-2022) 11:59PM

Instructions:

1. Assignments are to be done individually. You must complete this assignment by yourself.You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own and you must understand each part of your code. You are encouraged to get help from the instructional staff through google classroom.
2. The AIM of this assignment is to practice with Pointers, Dynamic Memory Allocation (DMA) in C++.
3. No late assignments will be accepted.
4. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.
5. Do not use any String or math libraries (such as cmath, cstring, string etc) and also do notuse built-in function (such as strlen, strcmp etc). **Caution**: zero marks will be awarded.
6. Marks distribution and test Cases are provided for each question. Your code will be evaluated with **similar test cases**. If the required output is generated, you will be awardedfull marks. Failing to generate the correct output will result in zero marks.
7. Do not edit **Function Prototypes**. **Caution**: zero marks will be awarded.
8. For **PrintPattern** questions, the output should be properly displayed and well presented. There will be no gtests for **PrintPattern** questions.
9. The usage of string is strictly prohibited.
10. Your code must be **generic**.
11. Marks distribution is provided for each question.
12. Static and global variables not allowed forrecursive functions.
13. **Plagiarism**: Plagiarism of any kind (copying from others, copying from the internet, etc) is not allowed. If found plagiarized, you will be awarded zero marks in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in the course.
14. **Please start early otherwise you will struggle with the assignment.**
15. **Submission Guidelines**:

Dear students we will be using auto-grading tools, so failure to submit according to the belowformat would result in zero marks in the relevant evaluation instrument.

i. For each question in your assignment, make a separate .cpp file e.g. for question 1, makeq1.cpp and so on. Each file that you submit must contain your name, student-id, and assignment # on top of the file in the comments.

ii. Combine all your work in one folder. The folder must contain only .cpp files (no binaries,no exe files etc.,). And don't remove **test cases** (in testcases.cpp) or **function prototypes** (in submission.cpp).

iii. Run and test your program on a lab machine before submission.

iv. Rename the folder as ROLL-NUM_SECTION (e.g. 19i-0001_B) and compress the folderas a zip file. (e.g. 19i-0001_B.zip).

v. Submit the .zip file on Google Classroom within the deadline.

vi. Submission other than Google Classroom (e.g. email etc.) will not be accepted.

vii. The student is solely responsible to check the final zip files for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroomdue to any reason, it will lead to zero marks in the assignment.

Q1: [Total 15 marks]

Write a C++ program using a dynamic array (or arrays) to assign passengers seats in a Bus and your program will ask the user how many rows the Bus has and will handle that many rows(Assume the Bus does not always have the same rows). **(5 points)**

Expected output: Assume a small Bus with seat numbering as follows:

```
 1 A B C D
 2 A B C D
 3 A B C D
 4 A B C D
 5 A B C D
 6 A B C D
 7 A B C D
 8 A B C D
 9 A B C D
10 A B C D
```

The program should display the seat pattern, with an X marking the seats already assigned. For example, after seats 1A, 2B, and 4C are taken, the display should look like this: **(5 points)**

```
1 X B C D
2 A X C D
3 A B C D
4 A B X D
5 A B C D
6 A B C D
7 A B C D
8 A B C D
9 A B C D
10A B C D
```

After displaying the seats available, the program prompts for the seat desired, the user types in a seat, and then the display of available seats is updated. This continues until all seats are filled or until the user signals that the program should end. If the user types in a seat that is already assigned, the program should say that the seat is occupied and ask for another choice. **(10 points)**

Q2: Part A [Total 20 marks]

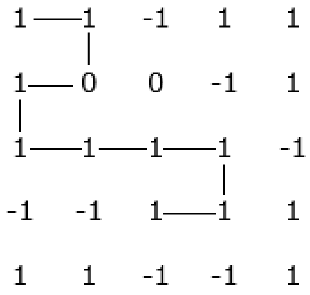
Given an M × N matrix where each cell can have a value of 1, 0, or -1, where -1 denotes an unsafecell, collect the maximum number of ones starting from the first cell and by visiting only safe cells(i.e., 0 or 1). We can only go left or down if the row is odd; otherwise, we can only go right or down from the current cell. Write a program to count the calculate the maximum score collected.

<u>Your Tasks:</u>

1. Your program should ask the user total number of rows and columns of the 2D array (use only dynamic arrays). (3 points)
2. Fill the array with random values ranging from -1 to 1. (2 points)
3. Pass the array as a reference to the function. (3 points)
4. Function should return the total maximum score. Array should be accessed using pointer notation only. (12 points)

Example:

1	1	-1	1	1
1	0	0	-1	1
1	1	1	1	-1
-1	-1	1	1	1
1	1	-1	-1	1



The Maximum Score Collected is 9.

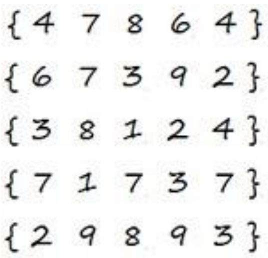
Q2: Part B [Total 20 marks]

Given an `M × N` matrix of integers where each cell has a cost associated with it, find the minimum cost to reach the last cell `(M-1, N-1)` of the matrix from its first cell `(0, 0)`. We can only move one unit right or one unit down from any cell, i.e., from cell `(i, j)`, we can move to `(i,j+1)` or `(i+1, j)`.

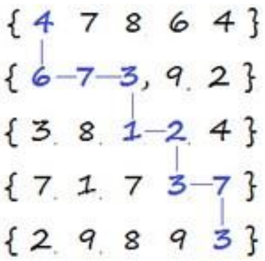
<u>Your Tasks:</u>

1. Your program should ask the user total number of rows and columns of the 2D array (use only dynamic arrays). **(1.5 points)**
2. Fill the array with random values ranging from 1 to 9. **(2 points)**
3. Pass the array as a reference to the function. **(1.5 points)**
4. Function should return the total minimum score. Array should be accessed using pointer notation only. **(15 points)**

Example:



```
{4 7 8 6 4}
{6 7 3 9 2}
{3 8 1 2 4}
{7 1 7 3 7}
{2 9 8 9 3}
```



```
{4 7 8 6 4}
{6-7-3, 9. 2}
{3. 8. 1-2. 4}
{7. 1. 7 3-7}
{2. 9. 8 9 3}
```

The Minimum Cost is 36.

Q3: [Total 20 marks]

Write a C++ program using dynamic arrays that allows the user to enter the last names ofthe candidates in a local election and the number of votes received by each candidate. The program must ask the user for the number of candidates and then create the appropriate arraysto hold the data. The program should then output each candidate's name, the number of votesreceived, and the percentage of the total votes received by the candidate. Your program should also output the winner of the election.

Name of Candidate	Votes Received	% of Total Votes
Ali	5000	25.91
Imran	4000	20.73
Ahmad	6000	31.09
Ijaz	2500	12.95
Khan	1800	9.33
Total	19300	

The winner of the local election is Ahmad.

Q4: [Total 20 marks]

Write a C++ program that outputs a histogram of student Marks for a Mid-term-1 Examination. The program should input each student's Marks as an integer and store the Marks in a dynamic array. Marks should be entered until the user enters -1 for marks. The program should then scan through the Dynamic array and compute the histogram. In computing the histogram, the minimum value of a marks is 0 but your program should determine the maximum value entered by the user. Then use a dynamic array to store and output the histogram.

For example, if the input is:

```
80
60
80
70
60
50
50
50
-1
```

Then the output should be:

```
The frequency of 80's: 2
The frequency of 70's: 1
The frequency of 60's: 2
The frequency of 50's: 3
```

For Histogram: https://www.mathsisfun.com/data/histograms.html

Q5: [Total 20 marks]

Write a C++ program (using pointers and dynamic memory allocation only) to implement the following functions and call it from the main function.

i. Write a function which take an int array and the array' size as arguments. It should return maximum value of the array elements (use only pointers with no loop variable in the body of the loop). **(3 points)**

ii. Write a function to swap the values of two integer variables by using pointers (instead of reference variables) and return its swapped values to the main function.**(3 points)**

iii. Write a function that takes two int arrays and the arrays' sizes as arguments (e.g., four arguments). It should create a new array big enough to store both arrays. Then it should copy the contents of the first array to the new array, and then copy the contents of the second array to the new array in the remaining elements and return apointer to the new array. **(4 points)**

iv. Write a piece of code which prints the characters in a cstring in a reverse order. **(2 points)**

v. Create an array of Planets. Populate the array and print the contents of the array using the pointer notation instead of the subscripts. **(3 points)**

vi. Write a function countEven(int*, int) which receives an integer array and its size, and returns the number of even numbers in the array**. (3 points)**

vii. Write a function whose signature looks like (char*, char) which returns true if the 1st parameter cstring contains the 2nd parameter char, or false otherwise. **(2 points)**

Q6: Recursive Functions

Marks:10

Write a recursive function find that finds given target value in the array. If value is not found your function must return -1, otherwise it should return the index of array where the value was found. Your function prototype must be as follows:

int find(int array[], int length, int target);

Marks:15

A palindrome is any word, phrase, or sentence that reads the same forward and backward. Here are some well-known palindromes:

Able was I, I saw Elba

Desserts, stressed

Kayak
Write a bool function that uses recursion to determine if a string argument is a palindrome. The function should return true if the argument reads the same forward and backward.
Function Prototype:
bool isPalindrome(char str[], int size);

Marks:10
Write a recursive function PrintPattern1 to print pattern that takes two integer arguments **n** and **k**. **n** is the starting number while **k** is the ending limit.
Example:
PrintPattern1(1, 5,'@','+') would print the following pattern.
1@2@@3@@@4@@@@5@@@@@4++++3+++2++1+

Prototype:
void PrintPattern1(int n, int k, char ch1, char ch2);

Marks:10
Write a C++ recursive function PrintPattern to print following pattern using recursion. No loops allowed whatsoever, and you can write maximum two functions apart from main function. For example, calling your function with these argument PrintPattern(5,'*') should print following pattern. Your function prototype must be as follows recursive function.
void PrintPattern(int value, char ch);

```
        *

      * *

    *   *

  *     *

* * * * *
```

Marks:10

Write a recursive function replace that changes all the occurrences of character 'f' in String 's' to character 't' and then returns the changed string. For example,
replace (steve, ‘e’, ‘a’) // it will return stava

replace (radar,’a’,’o’) // it will return rodor

Function Prototype:
void replace(char* s1, char ch1, char ch2);

Happy Coding 😊