CL-210

Data Structures

Lab # 2

|  |
| --- |
| Objectives:  1. Arrays    1. Insertion    2. Searching    3. Sorting 2. Pointers 3. Pointer to Function 4. DMA 5. Singly Linked List |

**Note: Carefully read the following instructions (*Each instruction contains a weightage*)**

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function about its functionality.
3. Use understandable name of variables.
4. Proper indentation of code is essential.
5. Make separate .cpp files for all tasks and use this format **22F-1234\_Task1.cpp.**
6. First think about statement problems and then write/draw your logic on copy.
7. After copy pencil work, code the problem statement on C++ compiler.
8. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every **task output in MS word and submit .cpp file with word file**.
9. Please submit your file in this format **22F-1234\_L1**.
10. Do not submit your assignment **after the deadline**.
11. **Do not copy code from any source otherwise you will be penalized with negative marks.**

|  |
| --- |
| **Problem 1:** |

Develop a program to create three 3x3 matrices using DMA and populate those two matrices using pointers now perform addition and multiplication using pointers and store the result into third matrix and print them also using pointers.

|  |
| --- |
| **Problem 2:** |

Populate an array of your size choice and write a program to find the largest and smallest element in that array.

|  |
| --- |
| **Problem 3:** |

Write a program that places a developer check on arrays index out of bound i.e. it checks during insertion that the index is not less than zero or greater than the mentioned size.

**Problem 4:**

Write a code to take input in 2 dynamic arrays as long as the user doesn't enter sentinel value. After you have finished taking input for one dynamic array start for the second. Their size depends on after entering *how many elements* the user enters the sentinel value and their size may be different. Now completely swap the contents of the two dynamic arrays.



**Problem 5:**

Implement another function; pass the array (again using a pointer) to it. The function should then sort the array using the Bubble Sort algorithm. The function should use pointers for all computations (counter variables, traversal, swap).



**Problem 6:**

We need to implement Array data structure that can store real numbers and have the following public functionalities:

* Constructor: with the size of the array as the parameter
* Destructor
* Size: returning the current size of the array
* Insert: input the content of the array from the user
* Delete: remove the element specified by the user
* Print: printing the content of the array
* Avg: returning the arithmetic mean of the values of the array
* Min: returning the smallest element present in the array
* Max: returning the largest element present in the array
* Reverse: reversing the order of the values in the array

Note: Implement this data structure using classes.