**Data Structures Lab 2**

**Course:** Data Structures (CL2001) **Semester:** Fall 2023

**Instructor: Taha Ahnmed**  **T.A:** N/A

**Note:**

* + - * Lab manual cover following below elementary sorting algorithms

**{Bubble, insertion, selection}**

* Maintain discipline during the lab.
* Just raise your hand if you have any problem.
* Completing all tasks of each lab is compulsory.
* Get your lab checked at the end of the session.

**Task-1:**

Given an array of strings arr[]. Sort given strings using Bubble Sort and display the sorted array.

**Key Points**:

1. Bubble Sort, the two successive strings arr[i] and arr[i+1] are exchanged whenever arr[i]> arr[i+1]. The larger values sink to the bottom and hence called sinking sort. At the end of each pass, smaller values gradually “bubble” their way upward to the top and hence called bubble sort.



**Task-2:**

Develop C++ solution such that day month and year are taken as input for 5 records and perform Sorting Dates based on year, month, day using Selection Sort.

**Key Points**:

void selectionSort(int \*array, int size) {

Find the smallest element in the array and exchange it with the element in the first position.

Find the second smallest element in the array and exchange it with the element in the second position.

Continue this process until done.

}

Diagram

Description automatically generated

**Task-3:**

Develop an implementation of insertion sort that runs in reverse order i.e. right to left.

**Key Points**:

void insertionSort (int \*array, int size) {

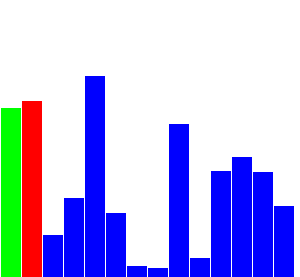
Choose the second element in the array and place it in order with respect to the first element.

Choose the third element in the array and place it in order with respect to the first two elements.

Continue this process until done.

Insertion of an element among those previously considered consists of moving larger elements one position to the right and then inserting the element into the vacated position

}



**Task 4:**

Given an unsorted array arr[0..n-1] of size n, find the minimum length subarray arr[…] such that sorting this subarray makes the whole array sorted.   
**Examples:**   
1) If the input array is [10, 12, 20, 30, 25, 40, 32, 31, 35, 50, 60], your program should be able to find that the subarray lies between the indexes 3 and 8.  
2) If the input array is [0, 1, 15, 25, 6, 7, 30, 40, 50], your program should be able to find that the subarray lies between the indexes 2 and 5.

**Key Points (Hint)**:

**Traverse the array from left to right tracking the max, saving the last found index value of ‘j’ which will be less than max**

**Traverse the array from right to left tracking the min, saving the last found index value of ‘i’ which will be greater than max**

**Sort the segment array from index i to j**

**Task 5:**

A clerk at a shipping company is charged with the task of rearranging a number of large crates in order of the time they are to be shipped out. Thus, the cost of compares is very low relative to the cost of exchanges (move the crates). The warehouse is nearly full: there is extra space sufficient to hold any one of the crates, but not two. Which sorting method should the clerk use? Implement this question via a user generated array?