

## Data Structures Lab 2

**Course:** Data Structures (CL2001)

**Instructor:** Sameer Faisal

**Semester:** Fall 2023

**T.A:** N/A

---

### Note:

- Lab manual cover following below elementary sorting algorithms  
**{Bubble, insertion, selection sort algorithms}**
  - Maintain discipline during the lab.
  - Just raise 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.

Take array[10]= {5,1,3,6,2,9,7,4,8,10}

### 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 using Selection Sort. Note: Input must be taken from user

[Hint: Struct can be used]

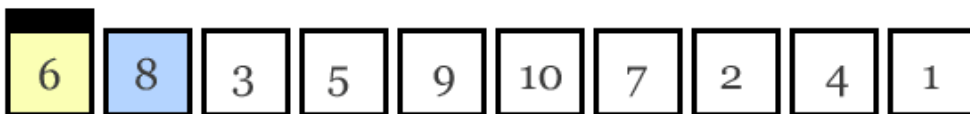
It's not strictly necessary to take inputs in the format as shown in example, but, the output should be in the given format.

Example Input:	Example Output:
01/02/2022	4/07/2015
5/01/2018	5/01/2018
4/07/2015	12/10/2021
12/10/2021	01/02/2022
11/12/2023	11/12/2023

**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.  
}
```

Note: Use a function to pass an array do not do this directly in the main function.



**Task-3:**

In a bustling corporate office, the facilities management team is tasked with organizing the seating arrangements for employees based on their designations. The office layout consists of rows of computer desks, and each desk has a designated employee.

The priority is to sort out the computer desks for employees using the Insertion Sort algorithm, with the designation determining the sorting order. The higher the designation, the closer the employee should be seated to the corner office. The designations and their corresponding priorities are as follows:

1. CEO (Chief Executive Officer) - Highest Priority
2. CTO (Chief Technology Officer)
3. CFO (Chief Financial Officer)
4. VP (Vice President)
5. MGR (Manager)
6. EMP (Employee) - Lowest Priority

Here's the initial arrangement of employees' desks from left to right:

1. Employee (EMP)
2. CFO (Chief Financial Officer)
3. Manager (MGR)
4. Employee (EMP)
5. VP (Vice President)
6. CTO (Chief Technology Officer)
7. Manager (MGR)
8. CEO (Chief Executive Officer)

**Key Points:**

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

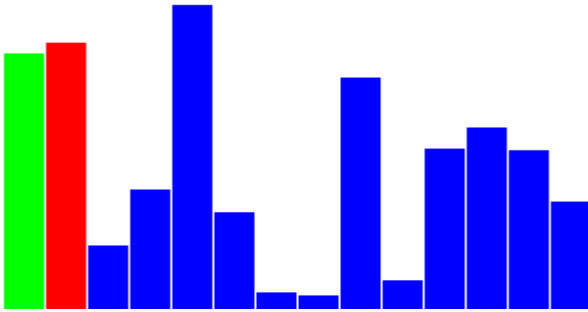
Choose the second element in the array and place it in order with respect to the first element.

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:

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?