**Logo, company name

Description automatically generated**

**Department of (Computer Science)**

**Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan**

**COMP-112L Data Structure** **& Algorithm Lab**

**Lab Journal**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 4th**

**Submission Date: 26th April 2022**

**Submitted to: Engr. Rafi-Ullah**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Signature**

**Lab No. 04**

**Sorting Algorithms Insertion Sort and Bubble Sort**

**Objectives:**

In this lab we will be discussing about Sorting Algorithms like Insertion Sort & Bubble Sort in detail. This is one of the most important concepts in C++ language.

The Objective of Lab is:

* To understand & implement the working of sorting algorithms using arrays in C++.

**Tools/Software Required:**

* All the tasks are implemented on DEV C++.

**Introduction:**

**SORTING:**

It is often necessary to arrange the elements in an array in numerical order from highest to lowest values i.e., from ascending to descending and vice versa. If an array contains string values or alphabetical order, then arrays need to be sorted. The process of sorting an array requires the exchanging of values. While this seems to be a simple process, a computer must be careful that no values are lost during this exchange.

**BUBBLE SORT:**

* Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared, and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of Ο (n2) where n is the number of items.

**Lab Tasks:**

**Task 1:**

Write a code to sort the following arrays using selection sort method. [10, 34, 2, 56,7,67, 88, 42]

**Code:**

**#include<iostream>**

**using namespace std;**

**void display(int arr[],int size);**

**void selectionSort(int arr[],int size);**

**int main()**

**{**

**int size=8;**

**int arr[size] = {10,34,2,56,7,67,88,42};**

**cout<<"Input array is: ";**

**display(arr,size);**

**selectionSort(arr,size);**

**return 0;**

**}**

**void display(int arr[],int size)**

**{**

**cout<<"\n[";**

**for(int i=0; i<size; i++)**

**{**

**cout<<arr[i]<<" ";**

**}**

**cout<<"]";**

**}**

**void selectionSort(int arr[],int size)**

**{**

**int Min,i,j;**

**for(i=0; i<size-1; i++)**

**{**

**Min=i;**

**for(j=i+1; j<size; j++)**

**{**

**if(arr[j]<arr[Min])**

**{**

**Min=j;**

**}**

**}**

**if(Min!=i)**

**{**

**cout<<"\nSwapping the number "<<arr[i]<<" and "<<arr[Min]<<" :";**

**int temp=arr[Min];**

**arr[Min]=arr[i];**

**arr[i]=temp;**

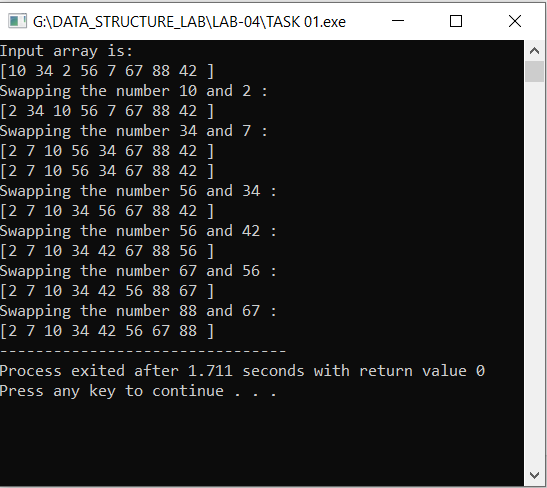
**}**

**display(arr,size);**

**}**

**}**

**Output:**

****

**Task # 02:**

Write a code to sort the following arrays using bubble sort method. [10, 34, 2, 56, 7, 67, 88, 42].

**Code:**

**#include<iostream>**

**using namespace std;**

**void display(int arr[],int size);**

**void BubbleSort(int arr[],int size);**

**int main()**

**{**

**int size=8;**

**int arr[size] = {10,34,2,56,7,67,88,42};**

**cout<<"Input array is: ";**

**display(arr,size);**

**BubbleSort(arr,size);**

**return 0;**

**}**

**void display(int arr[],int size)**

**{**

**cout<<"\n[";**

**for(int i=0; i<size; i++)**

**{**

**cout<<arr[i]<<" ";**

**}**

**cout<<"]";**

**}**

**void BubbleSort(int arr[],int size)**

**{**

**int i,j;**

**for(i=0; i<size; i++)**

**{**

**for(j=0; j<size-1-i; j++)**

**{**

**if(arr[j]>arr[j+1])**

**{**

**cout<<"\nSwapping the number "<<arr[j]<<" and "<<arr[j+1]<<" :";**

**int temp=arr[j];**

**arr[j]=arr[j+1];**

**arr[j+1]=temp;**

**display(arr,size);**

**}**

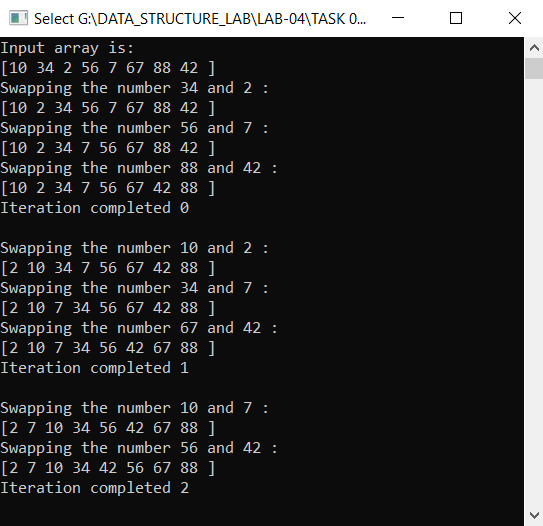
**}**

**cout<<"\nIteration completed "<<i<<endl;**

**}**

**}**

**Output:**

****

**Results & Observations:**

In this Lab I've learned about the concept of Sorting Algorithms like Insertion Sort & Bubble Sort & also understand that how they work. In the first task, I've used Selection Sort method in which first I’ve selected first index as a minimum number and then I find the minimum element other than that index after that I swapped the index number to the minimum number. In the second task, I've used Bubble Sort method in which I’ve compared two adjacent elements in the sequence wise.