FUNDAMENTALS OF PROGRAMMING (LAB)

LAB MANUAL # 08

SUBMITTED BY: - ABEER ZAHRA JAFARI (476474) ME-15(C)

TASK # 01

Write a C++ program to calculate average of numbers of array.

main.cpp

Run

```
#include <iostream>
using namespace std;
int main() {
    int n,sum,avg;
     sum=0;//initializing variable
cout<<"Enter the number of elements in the array:";cin>>n;
    int arr[n]={};
for (int i = 0; i < n; ++i) {
    cout << "Enter element "<<endl;
     cin >> arr[i];} //taking input from user
for (int i = 0; i < n; ++i) {
    cout << arr[i] << " "; } //displaying the array
cout<<endl;
for (int i = 0; i < n; ++i) {
    sum=sum+arr[i];} //summing all the elements
avg= sum/n;   //finding average
cout<<" the required average is:"<<avg;
return 0;}
```

Output

Clear

```
/tmp/OoyXD7aTT2.o
Enter the number of elements you want in the array:5
Enter element
1
Enter element
2
Enter element
4
Enter element
5
Enter element
3
1 2 4 5 3
 the required average is:3
```

TASK # 02

- Implement Bubble sort on an array of 5 integers.

main.cpp

Run

```
#include <iostream>
using namespace std;
int main() {
int n, a;
    cout << "Enter the number of elements in the array: ";
    cin >> n; //no. of elements in the array
int arr[n] = {};
for (int i = 0; i < n; ++i) {//creating array from user input
    cout << "Enter element ";
    cin >> arr[i];}
    cout <<"array: ";
for (int i = 0; i < n; ++i) {
    cout << arr[i] << " ";} //displaying the original array
    cout << endl;
for (int i = 0; i < n - 1; ++i) { //arranging in
                                  ascending order
        for (int j = 0; j < n - i - 1; ++j) { //nested loop
            if (arr[j] > arr[j + 1]) {

                a = arr[j];
                 arr[j] = arr[j + 1];
                 arr[j + 1] = a;}}}
cout << "Sorted array in ascending order: ";
    for (int i = 0; i < n; ++i) {
        cout << arr[i] << " ";}
return 0;}
```

Output

Clear

```
/tmp/CCW3vY6llY.o
Enter the number of elements in the array: 5
Enter element 2
Enter element 5
Enter element 3
Enter element 1
Enter element 4
array: 2 5 3 1 4
Sorted array in ascending order: 1 2 3 4 5
```

TASK # 03

Implement Selection Sort on an array of 5 integers.

main.cpp

Run

```
#include <iostream>
using namespace std;
int main() {
int n, a;
    cout << "Enter the number of elements in the array: ";
    cin >> n; //no. of elements in the array
int arr[n] = {};
for (int i = 0; i < n; ++i) {//creating array from user input
    cout << "Enter element ";
    cin >> arr[i];}
    cout <<"array: ";
for (int i = 0; i < n; ++i) {
    cout << arr[i] << " ";} //displaying the original array
    cout << endl;
for (int i=0;i<n-1;i++){
    int x=i;
```

Output

Clear

```
/tmp/501uZ9HEjh.o
Enter the number of elements in the array: 10
Enter element 4
Enter element 7
Enter element 9
Enter element 1
Enter element 2
Enter element 8
Enter element 0
Enter element 3
Enter element 5
Enter element 6
array: 4 7 9 1 2 8 0 3 5 6
selection sorted array in ascending order: 0 1 2 3 4 5 6 7 8 9
```

TASK # 03 (CONTD)

```
17      for(int j=i+1;j<n;j++){
18          if (arr[j] < arr[x])
19              x = j;  }
20      int temp = arr[i];  // arranging in ascending
21      arr[i] = arr[x];
22      arr[x] = temp;}
23  cout << "selection sorted array in ascending order: ";
24  for (int i = 0; i < n; ++i) {
25      cout << arr[i] << " ";}
26  return 0;}
```

Output

Clear

```
/tmp/501uZ9HEjh.o
Enter the number of elements in the array: 10
Enter element 4
Enter element 7
Enter element 9
Enter element 1
Enter element 2
Enter element 8
Enter element 0
Enter element 3
Enter element 5
Enter element 6
array: 4 7 9 1 2 8 0 3 5 6
selection sorted array in ascending order: 0 1 2 3 4 5 6 7 8 9
```

Theory

- These tasks all utilize branched structures like loops to produce the required output. The loop used here is a nested for loop. A for loop is much more concise than a while loop as we can write the initial condition, the end condition and the increment in a single line of code. A nested loop is created by placing one or more loops in the scope of a main loop.
- In the first task, the required output is achieved by using three for loops. The first loop takes user input to create an array, the second loop displays the loop on the screen and the final loop sums all the elements in the array. Once the sum is found the average is found by dividing the sum by the size of the array.
- The required output of the second and third task is the same however the methodology is quite different. The second task implements a bubble sort, in which elements of an array are sorted into ascending order by comparing each element to the element adjacent to them. The final task implements selection sort which also arranges the elements in ascending order by first determining the smallest element in each iteration and then swapping it with the first element of the array.