

## Algorithm Lab Task-1

### Submission Guidelines-

- Rename the file to your ID only. If your ID is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.
- Must submit within the time in Teams.
- Must include resources for all the sections in the table and Uploaded file should be in PDF format.

#### Question- 1

Perform Insertion sort algorithms for the given array:

[6,4,2,1,9,8,3,5,12,10,11]

**Solution (Do it on paper, and the solution should be attached here):**

### Question-1

Perform insertion sort algorithms for the given array:

6, 4, 2, 1, 9, 8, 3, 5, 12, 10, 11

$i=1$   
temp=4  
Sorted  
6 4 2 1 9 8 3 5 12 10 11

$i=2$   
temp=2  
4 6 2 1 9 8 3 5 12 10 11

$i=3$   
temp=2  
4 2 6 1 9 8 3 5 12 10 11

$i=4$   
temp=1  
2 4 6 1 9 8 3 5 12 10 11

$i=4$   
temp=1  
2 4 ① 6 9 8 3 5 12 10 11

$i=4$   
temp=1  
2 ① 4 6 9 8 3 5 12 10 11

① 2 4 6 9 8 3 5 12 10 11

$i=5$   
temp=9  
1 2 4 6 9 8 3 5 12 10 11

$i=6$   
temp=9  
1 2 4 6 9 8 3 5 12 10 11

$i=7$   
temp=3  
1 2 4 6 8 9 3 5 12 10 11

1 2 4 6 8 ③ 9 5 12 10 11

$i=7$   
temp=3  
1 2 4 6 ③ 8 9 5 12 10 11

1 2 4 ③ 6 8 9 5 12 10 11

1 2 ③ 4 6 8 9 5 12 10 11

i=8

1	2	3	4	6	8	9	5	12	10	11
---	---	---	---	---	---	---	---	----	----	----

temp=5

1 2 3 4 6 8 9 5 12 10 11

i=8

1 2 3 4 6 5 8 9 12 10 11

temp=5

1 2 3 4 5 6 8 9 12 10 11

i=9

1	2	3	4	5	6	8	9	12	10	11
---	---	---	---	---	---	---	---	----	----	----

temp=12

1	2	3	4	5	6	8	9	12	10	11
---	---	---	---	---	---	---	---	----	----	----

i=10

temp=10

1	2	3	4	5	6	8	9	10	12	11
---	---	---	---	---	---	---	---	----	----	----

i=11

temp=11

1	2	3	4	5	6	8	9	10	11	12
---	---	---	---	---	---	---	---	----	----	----

Sorted

**Code-**

```
#include <iostream>
#include<conio.h>
using namespace std;

int main()
{
    int a[11]={6,4,2,1,9,8,3,5,12,10,11},i,ptr,temp;

    for(i=1;i<11;i=i+1)
    {
        temp=a[i];
        ptr=i-1;

        while(temp<a[ptr] && ptr!=-1) // you also can put ptr>=0
        {
            a[ptr+1]=a[ptr];
            ptr--; // you also can put ptr = ptr-1;
        }
        a[ptr+1]=temp;
    }
    for(i=0;i<11;i=i+1)
    {
        cout<<a[i]<<" ";
    }
    getch;
}
```

**Output Screenshot-**

The screenshot shows a C++ IDE with a file named `main.cpp`. The code implements a bubble sort algorithm. It starts with an array `a` of 12 elements: `{6, 4, 2, 1, 9, 8, 3, 5, 12, 10, 11, 1}`. A `while` loop runs until `ptr` reaches 0, and a `for` loop inside it compares adjacent elements and swaps them if they are in the wrong order. The output shows the array sorted in ascending order: `1 2 3 4 5 6 8 9 10 11 12`. The process returned 0 (0x0) and the execution time was 0.104 s.

```
1 #include <iostream>
2 #include <conio.h>
3 using namespace std;
4
5 int main()
6 {
7     int a[11]={6,4,2,1,9,8,3,5,12,10,11},i,ptr,temp;
8
9     for(i=1;i<11;i=i+1)
10     {
11         temp=a[i];
12         ptr=i-1;
13
14         while(temp<a[ptr] && ptr!=1) // you also can put <ptr>=0
15         {
16             a[ptr+1]=a[ptr];
17             ptr--; // you also can put ptr = ptr-1;
18         }
19         a[ptr+1]=temp;
20     }
21     for(i=0;i<11;i=i+1)
22     {
23         cout<<a[i]<<" ";
24     }
25     getch();
26 }
27
28
```

Output:

```
1 2 3 4 5 6 8 9 10 11 12
Process returned 0 (0x0)    execution time : 0.104 s
Press any key to continue.
```

## Question- 2

Perform Selection sort algorithms for the given array:  
[4,3,2,10,12,1,5,6]

**Solution (Do it on paper, and the solution should be attached here):**



### Question-2

#### Selection Sort

i=0 [4, 3, 2, 10, 12, 1, 5, 6]

i=1 [1, 3, 2, 10, 12, 4, 5, 6]

i=2 [1, 2, 3, 10, 12, 4, 5, 6]

i=3 [1, 2, 3, 4, 12, 10, 5, 6]

i=4 [1, 2, 3, 4, 5, 10, 12, 6]

i=5 [1, 2, 3, 4, 5, 6, 12, 10]

i=6 [1, 2, 3, 4, 5, 6, 10, 12]

$$n = 8$$

$$\therefore (n-1) = 8-1 = 7$$

**Code-**

```
#include <iostream>
#include<conio.h>
using namespace std;

int main()
{
    //input
    int n;
    cin>>n;

    int arr[n];
    for(int i=0;i<n;i++)
    {
        cin>>arr[i];
    }

    //selection operation
    for(int i=0;i<n-1;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(arr[j]<arr[i])
            {
                int temp=arr[j];
                arr[j]=arr[i];
                arr[i]=temp;
            }
        }
    }
    //output
    for(int i=0;i<n;i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    getch;
}
```

## Output Screenshot-

The screenshot displays the Code::Blocks IDE interface. The main editor window shows a C++ program named `main.cpp` implementing a selection sort algorithm. The code includes headers for `<iostream>` and `<conio.h>`, uses the `std` namespace, and defines a `main` function. It prompts the user for the number of elements `n`, reads `n` values into an array `arr`, performs a selection sort, and then prints the sorted array. The output window on the right shows the execution results: the number of elements is 8, the input array is 4 3 2 10 12 1 5 6, and the sorted array is 1 2 3 4 5 6 10 12. The process returned 0 and took 21.569 seconds to execute. The bottom status bar indicates the current line is 8, column 9, position 100.

```
1 #include <iostream>
2 #include <conio.h>
3 using namespace std;
4
5 int main()
6 {
7     //input
8     cout<<"The number of elements: ";
9     int n;
10    cin>>n;
11
12    int arr[n];
13    for(int i=0;i<n;i++)
14    {
15        cin>>arr[i];
16    }
17
18    //selection operation
19    for(int i=0;i<n-1;i++)
20    {
21        for(int j=i+1;j<n;j++)
22        {
23            if(arr[j]<arr[i])
24            {
25                int temp=arr[j];
26                arr[j]=arr[i];
27                arr[i]=temp;
28            }
29        }
30    }
31
32    //output
33    for(int i=0;i<n;i++)
34    {
35        cout<<arr[i]<<" ";
36    }
37    cout<<endl;
38    getch();
39 }
```

The number of elements: 8  
4  
3  
2  
10  
12  
1  
5  
6  
1 2 3 4 5 6 10 12

Process returned 0 (0x0) execution time : 21.569 s  
Press any key to continue.

Build messages:  
==== Build file: "no target" in "no project" (compiler: unknown) ====  
==== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ====