Practical No. 03

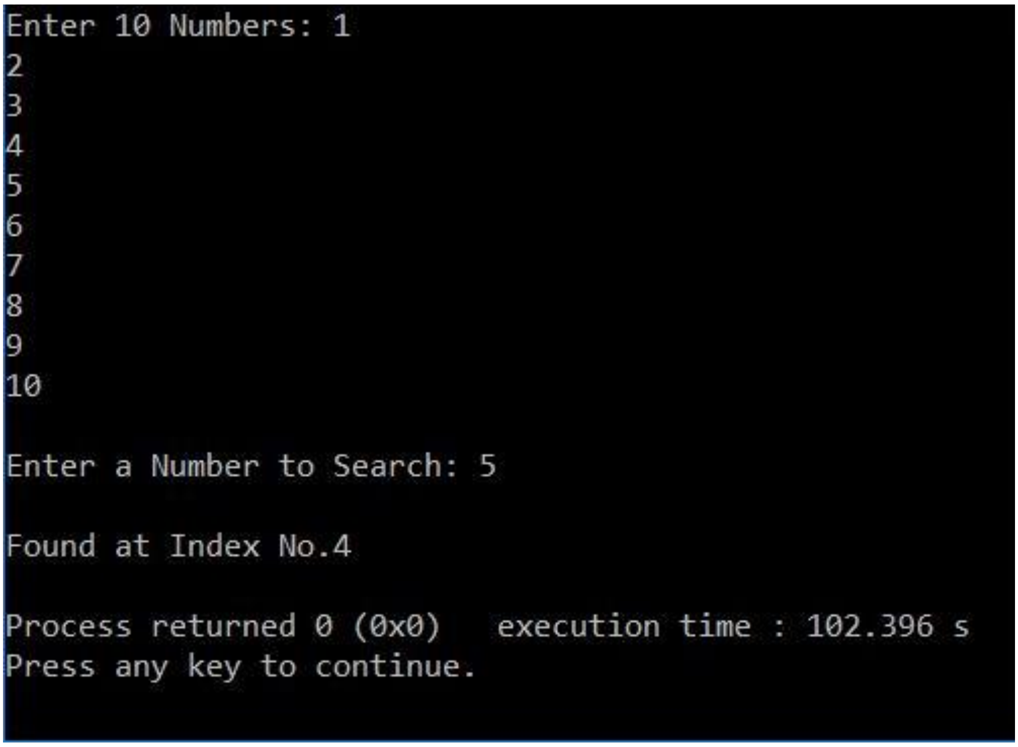
Name: Vijiyant Tanaji Shejwalkar

Reg No: 2020BIT057

Searching Algorithm

1. Linear Search in C++

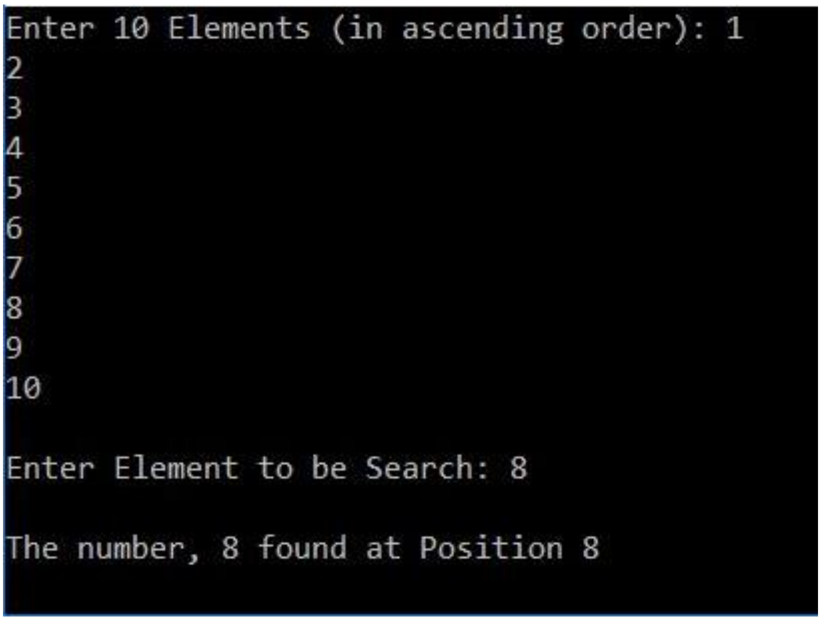
```
using namespace std;
int main()
{
    int arr[10], i, num, index;
    cout<<"Enter 10 Numbers: ";
    for(i=0; i<10; i++)
        cin>>arr[i];
    cout<<"\nEnter a Number to Search: ";
    cin>>num;
    for(i=0; i<10; i++)
    {
        if(arr[i]==num)
        {
            index = i;
            break;
        }
    }
    cout<<"\nFound at Index No."<<index;
    cout<<endl;
    return 0;
}
```



```
Enter 10 Numbers: 1
2
3
4
5
6
7
8
9
10

Enter a Number to Search: 5

Found at Index No.4

Process returned 0 (0x0)   execution time : 102.396 s
Press any key to continue.
```

2. Binary Search in C++

```
#include<iostream>
using namespace std;
int main()
{
   int i, arr[10], num, first, last, middle;
   cout<<"Enter 10 Elements (in ascending order): ";
   for(i=0; i<10; i++)
      cin>>arr[i];
   cout<<"\nEnter Element to be Search: ";
   cin>>num;
   first = 0;
   last = 9;
   middle = (first+last)/2;
   while(first <= last)
   {
      if(arr[middle]<num)
         first = middle+1;
      else if(arr[middle]==num)
      {
         cout<<"\nThe number, "<<num<<" found at Position "<<middle+1;
         break;
      }
      else
         last = middle-1;
      middle = (first+last)/2;
   }
   if(first>last)
      cout<<"\nThe number, "<<num<<" is not found in given Array";
   cout<<endl;
   return 0;
}
```



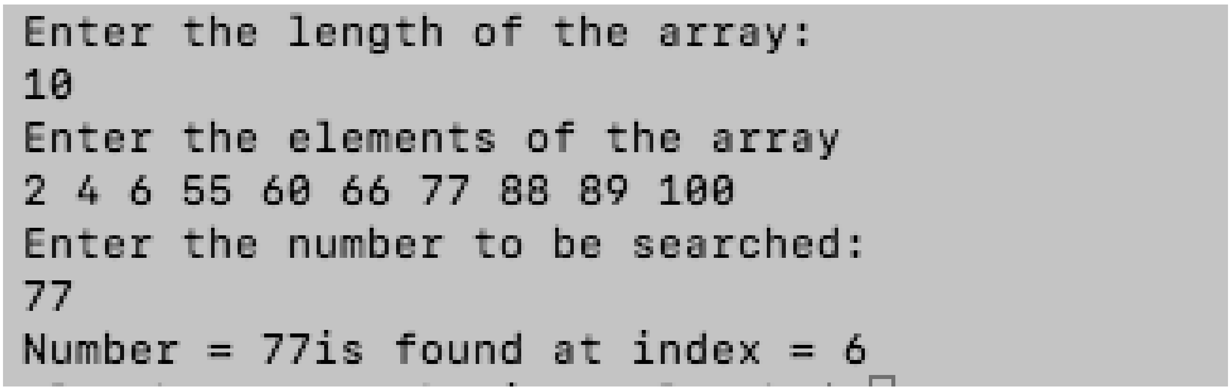
```
Enter 10 Elements (in ascending order): 1
2
3
4
5
6
7
8
9
10

Enter Element to be Search: 8

The number, 8 found at Position 8
```

3. Jump search using c++

```
#include <iostream>

#include <vector>

#include <cmath>


using namespace std;


int jumpSearchProg(vector <int> arr, int noToSearch, int ArrayLim)

{

        int previous = 0;

        int step = sqrt(ArrayLim);

        //Step to skip elements for jumping


        while (arr[min(step,ArrayLim)-1] < noToSearch)

        {

                previous = step;

                step += sqrt(ArrayLim);

                if(previous >= ArrayLim) return -1;

        }


        /*Applying linear Search and starting from the previous elements*/

        while (arr[previous] < noToSearch)

        {

                previous++;

        /*If element has not found yet then it means element is not present in the array*/

                if (previous == min(step, ArrayLim)) return -1;
```

```
    }
    // if we found the element then

    if (arr[previous] == noToSearch)
        return previous;

    return -1;
}

//Start of main
int main()
{
    int n,NoToSr;
    cout<<"Enter the length of the array:"<<endl;
    cin>>n;
    vector<int> arr(n);

    cout<<"Enter the elements of the array"<<endl;
    for(int i=0;i<n;i++)
    {
        cin>>arr[i];
    }

    cout<<"Enter the number to be searched:"<<endl;
    cin>>NoToSr;

    //function calling
    int result = jumpSearchProg(arr, NoToSr, n);

    //displayin foud number
    cout<<"Number = "<<NoToSr<<"is found at index = "<<result<<endl;
```

```
        return 0;

}
```



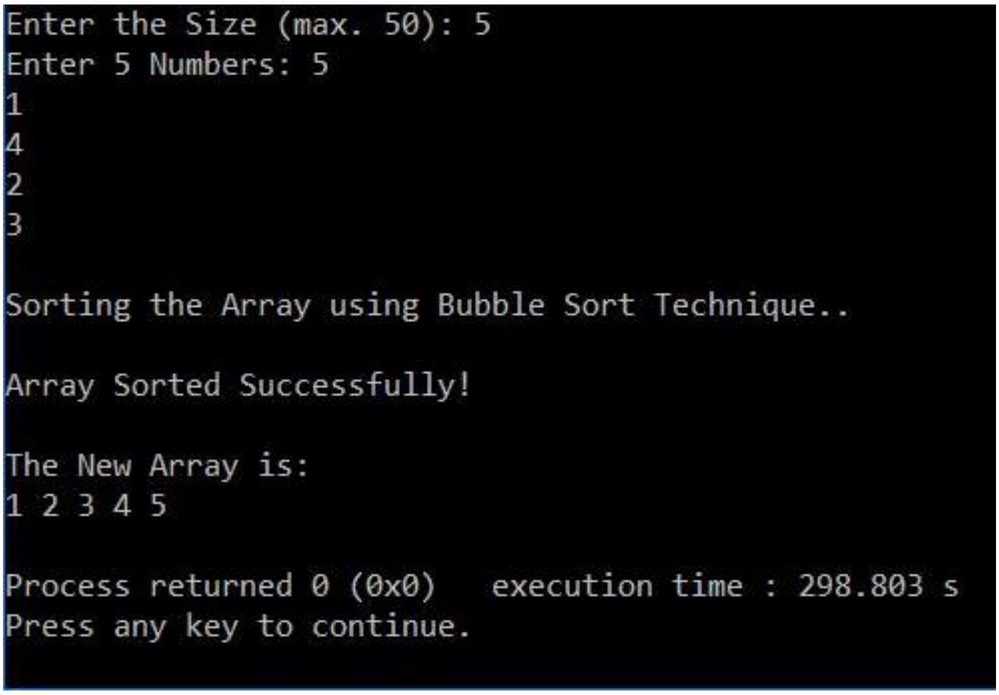
```
Enter the length of the array:
10
Enter the elements of the array
2 4 6 55 60 66 77 88 89 100
Enter the number to be searched:
77
Number = 77is found at index = 6
```

Sorting Algorithms

1. Bubble Sort in C++

```
#include<iostream>
using namespace std;
int main()
{
    int n, i, arr[50], j, temp;
    cout<<"Enter the Size (max. 50): ";
    cin>>n;
    cout<<"Enter "<<n<<" Numbers: ";
    for(i=0; i<n; i++)
        cin>>arr[i];
    cout<<"\nSorting the Array using Bubble Sort Technique..\n";
    for(i=0; i<(n-1); i++)
    {
        for(j=0; j<(n-i-1); j++)
        {
            if(arr[j]>arr[j+1])
            {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
```

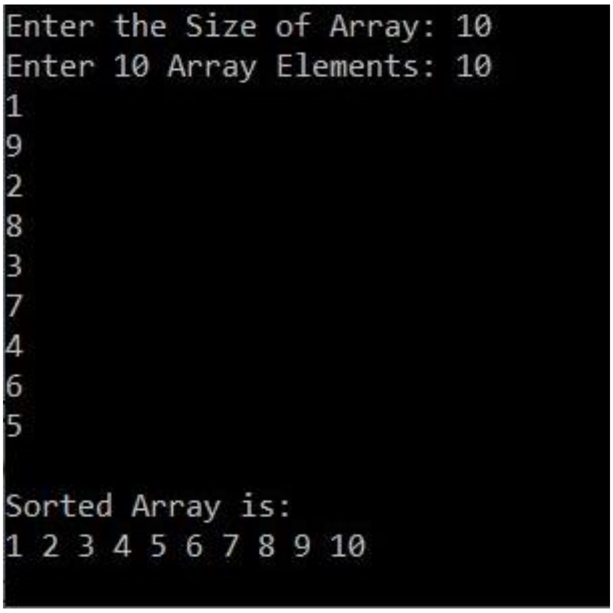
```
   }
    cout<<"\nArray Sorted Successfully!\n";
   cout<<"\nThe New Array is: \n";
   for(i=0; i<n; i++)
       cout<<arr[i]<<" ";
   cout<<endl;
   return 0;
}
```



```
Enter the Size (max. 50): 5
Enter 5 Numbers: 5
1
4
2
3

Sorting the Array using Bubble Sort Technique..

Array Sorted Successfully!

The New Array is:
1 2 3 4 5

Process returned 0 (0x0)   execution time : 298.803 s
Press any key to continue.
```

2. Selection Sort in C++

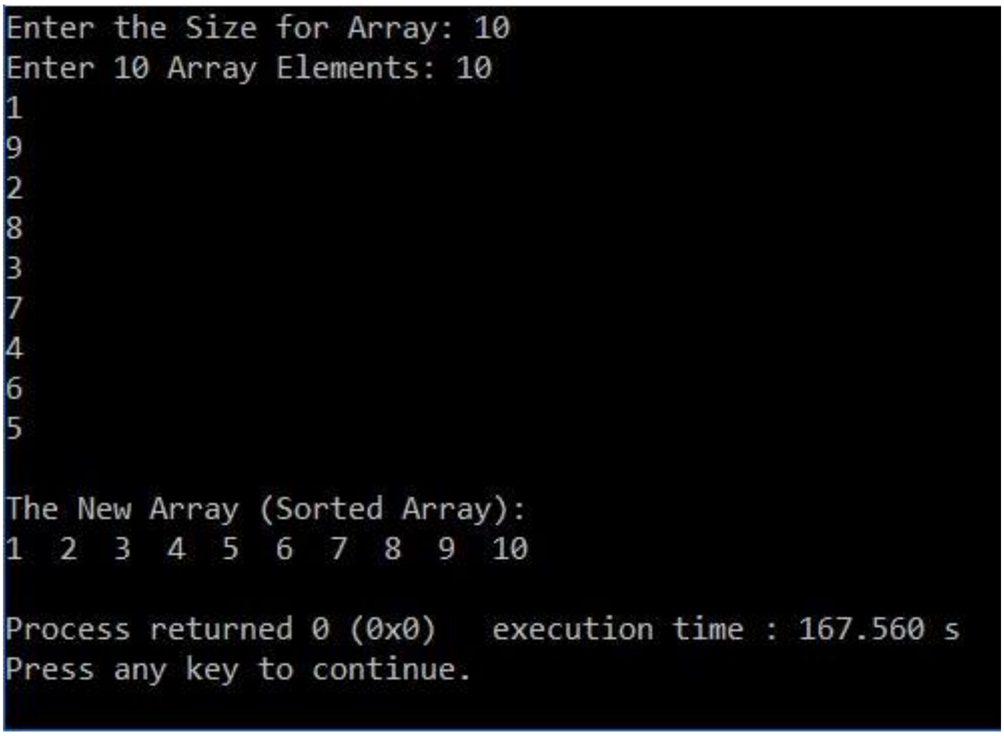
```
#include<iostream>
using namespace std;
int main()
{
    int tot, arr[50], i, j, temp, small, chk, index;
    cout<<"Enter the Size of Array: ";
    cin>>tot;
    cout<<"Enter "<<tot<<" Array Elements: ";
    for(i=0; i<tot; i++)
        cin>>arr[i];
    for(i=0; i<(tot-1); i++)
    {
        chk=0;
        small = arr[i];
        for(j=(i+1); j<tot; j++)
        {
            if(small>arr[j])
            {
                small = arr[j];
                chk++;
                index = j;
            }
        }
        if(chk!=0)
        {
            temp = arr[i];
            arr[i] = small;
            arr[index] = temp;
        }
    }
    cout<<"\nSorted Array is:\n";
    for(i=0; i<tot; i++)
        cout<<arr[i]<<" ";
    cout<<endl;
    return 0;
}
```



```
Enter the Size of Array: 10
Enter 10 Array Elements: 10
1
9
2
8
3
7
4
6
5

Sorted Array is:
1 2 3 4 5 6 7 8 9 10
```

3. Insertion Sort in C++

```
#include<iostream>
using namespace std;
int main()
{
   int arr[50], tot, i, j, k, elem, index;
   cout<<"Enter the Size for Array: ";
   cin>>tot;
   cout<<"Enter "<<tot<<" Array Elements: ";
   for(i=0; i<tot; i++)
      cin>>arr[i];
   for(i=1; i<tot; i++)
   {
      elem = arr[i];
      if(elem<arr[i-1])
      {
         for(j=0; j<=i; j++)
         {
            if(elem<arr[j])
            {
               index = j;
               for(k=i; k>j; k--)
                  arr[k] = arr[k-1];
               break;
            }
      }
    }
    else
      continue;
    arr[index] = elem;
  }
  cout<<"\nThe New Array (Sorted Array):\n";
  for(i=0; i<tot; i++)
    cout<<arr[i]<<"  ";
  cout<<endl;
  return 0;
}
```



```
Enter the Size for Array: 10
Enter 10 Array Elements: 10
1
9
2
8
3
7
4
6
5

The New Array (Sorted Array):
1  2  3  4  5  6  7  8  9  10

Process returned 0 (0x0)   execution time : 167.560 s
Press any key to continue.
```