**8.1 Introduction Array.**

**Array:** "Array is a list of variables of similar type."

dataType arrayName[size];

Ex: {1,2,3}    {a,b,c,d}

int array[4] = {40,45,50,55};

int = 4 Bytes

Array = 4 X 4 bytes = 16 bytes

**Q. Find Maximum and Minimum Number in Array ?**

#include<iostream>

#include<limits.h> // new header file

using namespace std;

int main(){

    int n;

    cin>>n;

    int arr[n];

    for(int i=0; i<n; i++){

        cin>>arr[i];

    }

    int maxNo=INT\_MIN;  // new header file used

    int minNo=INT\_MAX;  // new header file used

    for(int i=0; i<n; i++){

        maxNo=max(maxNo,arr[i]);  //Buillting Function used 'max'

        minNo=min(minNo,arr[i]);  //Buillting Function used 'min'

    }

    cout<<"MaxNo."<<maxNo<<"\tMinNo."<<minNo<<endl;

    return 0;

}

**8.2 Searching in Array.**

**2.1 Linear Search.**

**O (n) time complexity.(Normal)**

#include<iostream>

using namespace std;

int linearSearch(int arr[], int n, int key){

    for(int i=0; i<n; i++){

        if(arr[i]==key){

            return i;

        }

    }

    return -1;  //condition not match then output -1 print.

}

int main()

{

    int n;

    cin>>n;

    int arr[n];

    for(int i=0; i<n; i++){

        cin>>arr[i];

    }

    int key;

    cin>>key;

    cout<<linearSearch(arr,n,key)<<endl;

    return 0;

}

**2. Binary Search              Date:28-02-22**

After first iteration, length of Array = n

After second iteration, length of Array = n/2

After third iteration, length of Array = (n/2)/2

After K iteration, length of Array = n/2k

    Let the length of array become 1 after K iterations

    = n/2k = 1

    = n = 2k

    = log2 (n) = log2 (2k)

    = log2 (n) = K (log2)2

    = K = (log2)2

**Time Complexity O(log2)n**

#include<iostream>

using namespace std;

int binarySearch(int arr[], int n, int key){

    int s=0;

    int e=n;

    while(s<=e){

        int mid=(s+e)/2;

        if(arr[mid]==key){

            return mid;

        }

        else if(arr[mid]>key){

            e=mid-1;

        }

        else{

            s=mid+1;

        }

    }

    return -1;

}

int main(){

    int n;

    cin>>n;

    int arr[n];

    for(int i=0; i<n; i++){

        cin>>arr[i];

    }

    int key;

    cin>>key;

    cout<<binarySearch(arr,n,key)<<endl;

    return 0;

}

**8.3 Sorting in Array.   Date:28-02-22**

**3.1 Selection Short**

"Find the minimum element in unsorted array and swap it with element at beginning."

Example:

12,45,23,51,19,8 is Given Shorting Array

   8,45,23,51,19,12

   8,12,23,51,19,45

   8,12,19,51,23,45

   8,12,19,23,51,45

   8,12,19,23,45,51

Final Shorting Array is 8,12,19,23,45,51

#include<iostream>

using namespace std;

int main()

{

    int n;

    cin>>n;

    int arr[n];

    for(int i=0; i<n; i++){

        cin>>arr[i];

    }

    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;

            }

        }

    }

    for(int i=0; i<n; i++){

        cout<<arr[i]<<" ";

    }

    cout<<endl;

    return 0;

}

**3.2. Bubble Short.**

**"Repeatedly swap two adjacent elements if they are in wrong order."**

**ith iteration = n-i**

1.12,45,23,51,19,8   3. 12,23,19,8,45,51

   12,45,23,51,19,8      12,23,19,8,45,51

   12,23,45,51,19,8      12,19,23,8,45,51

   12,23,45,51,19,8      12,19,8,23,45,51

   12,23,45,19,51,8

   12,23,45,19,8,51   4. 12,19,8,23,45,51

                         12,19,8,23,45,51

2. 12,23,45,19,8,51      12,8,19,23,45,51

   12,23,45,19,8,51

   12,23,45,19,8,51   5. 12,8,19,23,45,51

   12,23,19,45,8,51      8,12,19,23,45,51

   12,23,19,8,45,51

Final Shorting array is 8,12,19,23,45,51

#include<iostream>

using namespace std;

int main()

{

    int n;

    cin>>n;

    int arr[n];

    for(int i=0; i<n; i++){

        cin>>arr[i];

    }

    int counter=1;

    while(counter<n-1){

        for(int i=0; i<n-counter; i++){

            if(arr[i]>arr[i+1]){

                int temp=arr[i];

                arr[i]=arr[i+1];

                arr[i+1]=temp;

            }

        }

        counter++;

    }

    for(int i=0; i<n; i++){

        cout<<arr[i]<<" ";

    }

    cout<<endl;

    return 0;

}

**3.3 Insertion Shorting.**

12,45,23,51,19,8

12,45,23,51,19,8

12,23,45,51,19,8

12,23,45,51,19,8

12,19,23,45,51,8

8,12,19,23,45,51 Final Shorting Array is : 8,12,19,23,45,51

#include<iostream>

using namespace std;

int main()

{

    int n;

    cin>>n;

    int arr[n];

    for(int i=0; i<n; i++){

        cin>>arr[i];

    }

    for(int i=1; i<n; i++){

        int current = arr[i];

        int j = i-1;

        while(arr[j]>current && j>=0)

        {

            arr[j+1]=arr[j];

            j--;

        }

        arr[j+1]=current;

    }

    for(int i=0; i<n; i++){

        cout<<arr[i]<<" ";

    }

    cout<<endl;

    return 0;

}