#include<iostream>

SourceCode

using namespace std;

void insertionSort(int \*a,int n){

    int key=0;

    int i,j;

    for(int i=1;i<n;i++){

        key=a[i];

        j=i-1;

        while(j>=0 and a[j]>key){

            a[j+1]=a[j];

            j=j-1;

        }

        a[j+1]=key;

    }

}

int main(){

    int n;

    int \*a;

    cout<<"Enter the Size: ";

    cin>>n;

    a=**new** int[n];

    cout<<"Enter the Elements: "<<endl;

    for(int i=0;i<n;i++){

        cin>>a[i];

    }

    insertionSort(a,n);

    for(int i=0;i<n;i++){

        cout<<a[i]<<" ";

}

}

OUTPUT

//Time Compexity : O(nlogn)

SourceCode

#include<iostream>

using namespace std;

void Merge(int arr[],int l,int mid,int r){

    int n1=mid-l+1;

    int n2=r-mid;

    int a[n1],b[n2];

    for(int i=0;i<n1;i++){

        a[i]=arr[l+i];

    }

    for(int i=0;i<n2;i++){

        b[i]=arr[mid+1+i];

    }

    int i=0,j=0,k=l;

    while(i<n1 and j<n2){

        if(a[i]<b[j]){

            arr[k]=a[i];

            i++;

        }

        else{

            arr[k]=b[j];

            j++;

        }

        k++;

    }

    while(i<n1){

        arr[k]=a[i];

        i++;

        k++;

    }

    while(j<n2){

        arr[k]=b[j];

        j++;

        k++;

    }

}

void MergeSort(int arr[],int l,int r){

    if(l<r){

        int mid=(l+r)/2;

        MergeSort(arr,l,mid);

        MergeSort(arr,mid+1,r);

        Merge(arr,l,mid,r);

    }

}

int main(){

    // int arr[]={5,4,3,2,1};

    int n;

    int \*a;

    cout<<"Enter the Size: ";

    cin>>n;

    a=**new** int[n];

    cout<<"Enter the Elements: "<<endl;

    for(int i=0;i<n;i++){

        cin>>a[i];

    }

    MergeSort(a,0,n-1);

    for(int i=0;i<5;i++){

        cout<<a[i]<<" ";

    }

    cout<<endl;

return 0;

}

OUTPUT