**15B17CI371 – Data Structures Lab**

**ODD 2024**

**Week 5-LAB A**

**Practice Lab**

**1. Write a program using linear search to check whether the inputted element belong to the it or not.**

#include <iostream>

using namespace std;

int main()

{

int n;

cout<<"Input the number of elements: ";

cin>>n;

int \*arr=new int[n];

cout<<"Input the elements: ";

for(int i=0;i<n;i++)

cin>>arr[i];

int key;

cout<<"Input the number to be searched : ";

cin>>key;

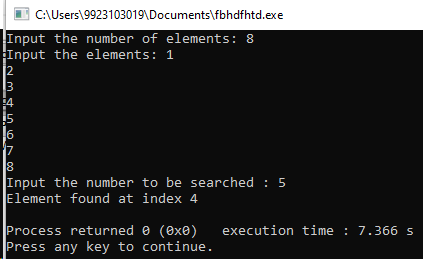
for(int i=0;i<n;i++)

if(arr[i]==key)

cout<<"Element found at index "<<i<<endl;

}

**Output :**



**2. Implement the binary search using iterative method.**

#include <iostream>

using namespace std;

int main()

{

int n;

cout<<"Input the number of elements: ";

cin>>n;

int \*arr=new int[n];

cout<<"Input the elements: ";

for(int i=0;i<n;i++)

cin>>arr[i];

int key;

cout<<"Input the number to be searched : ";

cin>>key;

int start=0,end=n-1,mid;

while(start!=end)

{

mid=start+(end-start)/2;

if(arr[mid]==key)

break;

else if(arr[mid]<key)

start=mid;

else

start=mid;

}

if(arr[mid]==key)

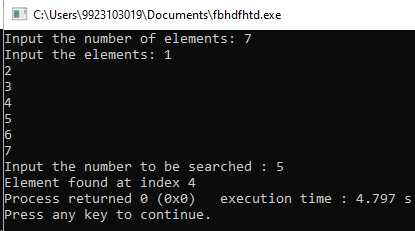
cout<<"Element found at index "<<mid;

else

cout<<"Element not found ";

}

**Output :**

****

**3. Write a function to find kth smallest /largest element in unsorted array.**

#include <iostream>

using namespace std;

int ksmall(int \*arr,int n,int k)

{

for(int i=0;i<k;i++)

for(int j=0;j<n-i-1;j++)

if(arr[j+1]<arr[j])

{

int temp=arr[j+1];

arr[j+1]=arr[j];

arr[j]=temp;

}

return arr[k-1];

}

int main()

{

int n;

cout<<"Input the number of elements: ";

cin>>n;

int \*arr=new int[n];

cout<<"Input the elements: ";

for(int i=0;i<n;i++)

cin>>arr[i];

int k;

cout<<"Input the value of k : ";

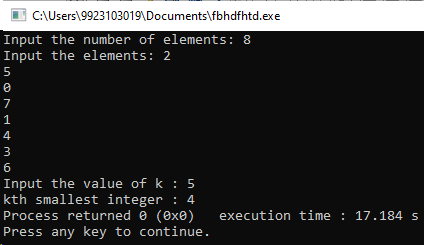
cin>>k;

int kthsmallest=ksmall(arr,n,k);

cout<<"kth smallest integer : "<<kthsmallest;

}

**Output :**

****

**4. Given a sorted array of size N and an integer K, find the position at which K is present in the array using interpolation search.**

#include <iostream>

using namespace std;

int main()

{

int n;

cout<<"Input the number of elements: ";

cin>>n;

int \*arr=new int[n];

cout<<"Input the elements: ";

for(int i=0;i<n;i++)

cin>>arr[i];

int key;

cout<<"Input the number to be searched : ";

cin>>key;

int start=0,end=n-1,pos;

while(start!=end)

{

pos=start+((key-arr[start])\*(end-start)/(arr[end]-arr[start]));

if(arr[pos]==key)

break;

else if(arr[pos]<key)

start=pos;

else

start=pos;

}

if(arr[pos]==key)

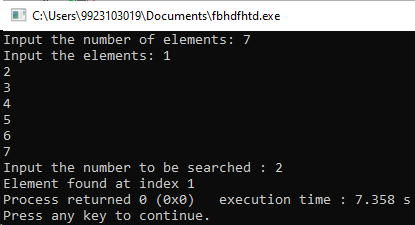
cout<<"Element found at index "<<pos;

else

cout<<"Element not found ";

}

**Output :**

****

**5. Given a sorted array of Strings and a String x, find an index of x if it is present in the array.**

#include <iostream>

using namespace std;

int main()

{

int n;

cout<<"Input the number of elements: ";

cin>>n;

string \*arr=new string[n];

cout<<"Input the elements: ";

for(int i=0;i<n;i++)

cin>>arr[i];

string key;

cout<<"Input the string to be searched : ";

cin>>key;

int start=0,end=n-1,pos;

while(start!=end)

{

pos=start+(end-start)/2;

if(arr[pos]==key)

break;

else if(arr[pos]<key)

start=pos;

else

start=pos;

}

if(arr[pos]==key)

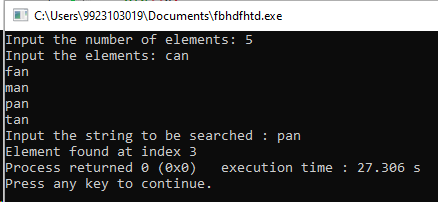
cout<<"Element found at index "<<pos;

else

cout<<"Element not found ";

}

**Output :**

****