**FIND UPPER AND LOWER BOUND**

Find position of the last and first occurrence of a given number in a sorted array. If number does not exist then print lower and upper bound as -1.

**Input Format:**

First line contains an integer n denoting the size of the array.  
Second line contains n space separated integers denoting array elements.  
Third line contains single integer Q denoting the no of queries.  
Q lines follow , each containing a single integer x that is to be searched in the array.

**Constraints:**

1 <= N <= 10^5  
1 <= Q <= 100

**Output Format**

For each query , print the lowerbound and the upperbound for the number x in a new line each.

**Sample Input**

5

1 2 3 3 4

3

2

3

10

**Sample Output**

1 1

2 3

-1 -1

**PROGRAM-**

#include <iostream>

using namespace std;

int firstocc(int a[],int l,int r,int key)

{

int mid,ans=-1;

while(l<=r)

{

mid=(l+r)/2;

if(a[mid]==key)

{

ans=mid;

r=mid-1;

}

else if(a[mid]>key)

r=mid-1;

else

l=mid+1;

}

return ans;

}

int lastocc(int a[],int l,int r,int key)

{

int mid,ans=-1;

while(l<=r)

{

mid=(l+r)/2;

if(a[mid]==key)

{

ans=mid;

l=mid+1;

}

else if(a[mid]>key)

r=mid-1;

else

l=mid+1;

}

return ans;

}

int main()

{

int n,q,i,key;

cin>>n;

int \*a=new int[n];

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

cin>>a[i];

cin>>q;

for(i=0;i<q;i++)

{

cin>>key;

int first=firstocc(a,0,n-1,key);

int last=lastocc(a,0,n-1,key);

cout<<first<<" "<<last<<endl;

}

}