**HELP RAHUL**

Rahul had a sorted array of numbers from which he had to find a given number quickly. His friend by mistake rotated the array. Now Rahul doesn't have time to sort the elements again. Help him to quickly find the given number from the rotated array.

Hint - Think Binary Search!



**Input Format:**

The first line contains N - the size of the array. the next N lines contain the numbers of the array.The next line contains the item to be searched.

**Constraints:**

**Output Format**

Output the index of number in the array to be searched. Output -1 if the number is not found.

**Sample Input**

5

4

5

1

2

3

2

**Sample Output**

3

Program-

#include<iostream>

using namespace std;

int binarySearch(int a[], int l, int r, int x)

{

if(l>r)

return -1;

int mid=(l+r)/2;

if(a[mid]==x)

return mid;

if(a[l]<=a[mid])

{

if(x>=a[l] && x<=a[mid])

return binarySearch(a,l,mid-1,x);

else

return binarySearch(a,mid+1,r,x);

}

if(x>=a[mid] && x<=a[r])

return binarySearch(a,mid+1,r,x);

return binarySearch(a,l,mid-1,x);

}

int main()

{

int n,i,q;

cin>>n;

int \*a=new int[n];

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

{

cin>>a[i];

}

cin>>q;

cout<<binarySearch(a,0,n-1,q);

}