import java.util.\*; // Program to make binary Search using Recursion

public class BinarySearch {

static Scanner sc = new Scanner(System.in);

static int arr[],n;

public BinarySearch(int nn) {

arr = new int[nn];

}

void elements() {

int i;

System.out.println("Enter the elements: ");

for(i=0;i<arr.length;i++) {

arr[i] = sc.nextInt();

}

}

int search() {

int x = 0; // local Variable

System.out.println("Enter the number to be searched: ");

x = sc.nextInt();

return x;

}

int binarySearch(int arr[], int l,int r, int x) // Returns index of x if

{ //it is present in arr[l..r], else return -1

if ((r>=l) && (l<arr.length-1)) {

int mid = l + (r - l) / 2; // If the element is present at the middle itself

if (arr[mid] == x)

return mid;

if (arr[mid] > x)

return binarySearch(arr, l, mid - 1, x);// Else the element can only

//be present in right subarray

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

}

return -1; // We reach here when element is not present in array

}

public static void main(String args[]) {

int n,x,result,sz;

System.out.println("Enter the size of array: ");

sz = sc.nextInt();

BinarySearch ob = new BinarySearch(sz);

ob.elements(); // Method Call using object

n = arr.length;

x = ob.search();

result = ob.binarySearch(arr, 0, n - 1, x);

if (result == -1)

System.out.println("Element not found");

else

System.out.println("Element found at "+(result+1)+"th position.");

}

}