### First and last occurrences of x

Given a sorted array **arr** containing **n** elements with possibly duplicate elements, the task is to find indexes of first and last occurrences of an element **x** in the given array.

**Example 1:**

**Input:**

n=9, x=5

arr[] = { 1, 3, 5, 5, 5, 5, 67, 123, 125 }

**Output:** 2 5

**Explanation**: First occurrence of 5 is at index 2 and last

  occurrence of 5 is at index 5.

**Example 2:**

**Input:**

n=9, x=7

arr[] = { 1, 3, 5, 5, 5, 5, 7, 123, 125 }

**Output:** 6 6

**Your Task:**  
Since, this is a function problem. You don't need to take any input, as it is already accomplished by the driver code. You just need to complete the function **find**() that takes **array arr, integer n and integer x** as parameters and returns the required answer.  
**Note:** If the number **x** is not found in the array just return both index as -1.

**Expected Time Complexity:** O(logN)  
**Expected Auxiliary Space:** O(1).

**Constraints:**  
1 ≤ N ≤ 107

**Company Tags**

[**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon)

//{ Driver Code Starts

//Initial Template for Java

import java.io.\*;

import java.util.\*;

// } Driver Code Ends

//User function Template for Java

class CodingMaxima

{

ArrayList<Long> find(long arr[], int n, int x)

{

ArrayList<Long> list=new ArrayList<>();

int f=-1,l=-1;

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

if(arr[i]==x && f==-1){

f=i;

}

if(arr[i]==x){

l=i;

}

}

list.add((long)f);

list.add((long)l);

return list;

}

}

//{ Driver Code Starts.

// Driver class

class Array {

// Driver code

public static void main(String[] args) throws IOException {

// Taking input using buffered reader

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int testcases = Integer.parseInt(br.readLine());

// looping through all testcases

while (testcases-- > 0) {

// int n = Integer.parseInt(br.readLine());

String line = br.readLine();

String[] q = line.trim().split("\\s+");

int n =Integer.parseInt(q[0]);

int x =Integer.parseInt(q[1]);

// //int y =Integer.parseInt(q[2]);

String line1 = br.readLine();

String[] a1 = line1.trim().split("\\s+");

long arr[] = new long[n];

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

arr[i] = Long.parseLong(a1[i]);

}

CodingMaxima ob = new CodingMaxima ();

ArrayList<Long> ans=ob.find(arr,n,x);

System.out.println(ans.get(0)+" "+ans.get(1));

}

}

}