**Longest Equal Prefix**

Show Topic Tags   

Given two positive integers x and y, and an array arr[] of positive integers. We need to find the longest prefix index which contains an equal number of x and y. Print the maximum index of largest prefix if exist otherwise print -1.

**Input:**  
The first line of input contains an integer T denoting the number of test cases. Each test case contains three integers n, x, and y where n denotes the number of elements in the array a[]. Next line contains space separated n elements in the array a[].  
  
**Output:**  
Print an integer which denotes the required prefix (0 based indexing).  
  
**Constraints:**  
1<=T<=50  
1<=n<=1000  
1<=a[i]<=100  
1<=x,y<=100  
  
**Example:  
Input:**  
2  
11 7 42  
7 42 5 6 42 8 7 5 3 6 7  
5 2 3  
2 2 3 3 1  
  
**Output:**  
9  
4

\*\*For More Examples Use Expected Output\*\*

Contributor: Shashwat Jain

<http://practice.geeksforgeeks.org/problems/longest-equal-prefix/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication252;

import java.util.\*;

import java.lang.\*;

import java.io.\*;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication252 {

/\*\*

\* @param args the command line arguments

\*/

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

// TODO code application logic here

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

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

while(t-- > 0) {

String[] input = br.readLine().trim().split(" ");

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

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

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

String[] sarr = br.readLine().trim().split(" ");

int[] arr = new int[n];

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

arr[i] = Integer.parseInt(sarr[i]);

}

int ans = -1;

int cont\_x = 0, cont\_y = 0;

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

{

if (arr[i] == x)

{

cont\_x++;

}

if (arr[i] == y)

{

cont\_y++;

}

if (cont\_x == cont\_y && cont\_x != 0 && cont\_y != 0)

{

ans = i;

}

}

System.out.println(ans);

}

}

}