**First element to occur k times**

Show Topic Tags   

Given an array of n integers. The task is to find the first element that occurs k number of times. If no element occurs k times the print -1. The distribution of integer elements could be in any range.

**Input:**  
The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. The first line of each test case contains an integer N denoting the size of an array and the number K. The second line of each test case contains N space separated integers denoting elements of the array A[ ].  
  
**Output:**  
For each test case in a new line print the required answer.

**Constraints:**  
1<= T <=100  
1<= N,K <=100000  
1<= A<=1000000

**Example:**

**Input :**  
1  
7 2  
1 7 4 3 4 8 7  
  
**Output :**  
7

**Explanation:**  
Both 7 and 4 occur 2 times. But 7 is the first that occurs 2 times.

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

<http://practice.geeksforgeeks.org/problems/first-element-to-occur-k-times/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication245;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import java.util.Iterator;

import java.util.LinkedHashMap;

import java.util.Map;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication245 {

/\*\*

\* @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 k = Integer.parseInt(input[1]);

String[] s\_arr = br.readLine().trim().split(" ");

int[] a = new int[n];

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

a[i] = Integer.parseInt(s\_arr[i]);

}

LinkedHashMap<Integer, Integer> hm = new LinkedHashMap<Integer, Integer>();

for(int elem : a) {

if(hm.containsKey(elem)) {

hm.put(elem, hm.get(elem)+1);

}else{

hm.put(elem, 1);

}

}

boolean hay = false;

for (Map.Entry<Integer, Integer> entry : hm.entrySet()) {

//System.out.println("Key = " + entry.getKey() + ", Value = " + entry.getValue());

if(entry.getValue() == k) {

hay = true;

System.out.println(entry.getKey());

break;

}

}

if(!hay) {

System.out.println(-1);

}

}

}

}