**K-th distinct element**

Show Topic Tags   

Given an array of N integers, write a program to print K-th distinct element in the array. The given array may contain duplicates and the output should print K-th element among all unique elements. If K is more than number of distinct elements, print -1.

**Input:**  
First line of input contains a single integer T which denotes the number of test cases. First line of each test case contains two space separated integers N and K. Second line of each test case contains N space separated integers.  
**Output:**  
For each test case, print the K-th distinct element as described above.  
  
**Constraints:**  
1<=T<=100  
1<=N<=100000  
1<=K<=100  
  
**Example:**  
**Input:**  
3  
6 2  
1 2 1 3 4 2  
6 3  
1 2 50 10 20 2  
4 2  
2 2 2 2  
**Output:**  
4  
10  
-1

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

Contributor: Harsh Agarwal

<http://practice.geeksforgeeks.org/problems/k-th-distinct-element/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication250;

import java.io.\*;

import java.math.\*;

import java.util.\*;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication250 {

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

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

int k = Integer.parseInt(nk[1]);

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

//int[] arr = new int[n];

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

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

//arr[i] = Integer.parseInt(input[i]);

int key = Integer.parseInt(input[i]);

if(hm.containsKey(key)) {

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

}else{

hm.put(key, 1);

}

}

ArrayList<Integer> unicos = new ArrayList();

for(int key : hm.keySet()) {

if(hm.get(key) == 1) {

unicos.add(key);

}

}

if(k -1 >= unicos.size()) {

System.out.println(-1);

}else{

System.out.println(unicos.get(k-1));

}

}

}

}