**First K natural numbers**

Show Topic Tags   

Given an array of size n and a number k, we need to print first k natural numbers that are not there in given array.

**Input:**  
First line consists of T test case. First line of every test case consists of N and K. Second line consists of elements of array.

**Output:**  
Single line output, print the K missing numbers.

**Constraints:**  
1<=T<=100  
1<=N<=10^4  
-1000<=Ai<=1000

**Example:  
Input:**  
2  
3 3  
1 4 3  
3 3  
-5 -6 1  
**Output:**  
2 5 6   
2 3 4

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

Contributor: Saksham Raj Seth

<http://practice.geeksforgeeks.org/problems/first-k-natural-numbers/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication250;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.math.BigInteger;

import java.util.Arrays;

import java.util.HashMap;

import java.util.HashSet;

import java.util.LinkedHashSet;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication250 {

/\*\*

\* @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\_1 = br.readLine().trim().split(" ");

int n = Integer.parseInt(input\_1[0]);

int k = Integer.parseInt(input\_1[1]);

String[] input\_2 = br.readLine().trim().split(" ");

int[] arr = new int[n];

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

arr[i] = Integer.parseInt(input\_2[i]);

}

int[] cont = new int[10000];

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

if(arr[i] >=0) {

cont[arr[i]]++;

}

}

int p =0;

for(int i =1; p<k && i<cont.length; i++) {

if(cont[i] ==0) {

System.out.print(i+ " ");

p++;

}

}

System.out.println();

}

}

}