**Find Smallest Missing in Sorted**

Show Topic Tags   

Given a **sorted**array Arr[] of N distinct integers where each integer is in the range from 0 to M-1 and M > N. You need to write a program to find the smallest number that is missing from the array. For example, if the array is Arr[] = {0, 1, 2, 6, 9} and n = 5, m = 10. Then the smallest missing number is 3.  
Note: Array does not contain repetitive elements.  
  
**Input**:  
First line of input contains an integer T which denotes the number of test cases. T test cases follows. First line of each test case contains two space separated integers N and M. Second line of each test case contains N space separated integers.  
**Output:**  
For each test case print the smallest missing number from the array in a new line.  
  
**Constraints:**  
1<=T<=100  
1<=N<=1000  
N  
**Example:  
Input:**  
4  
5 10  
0 1 2 6 9  
4 12  
4 5 10 11  
4 5  
0 1 2 3  
9 11  
0 1 2 3 4 5 6 7 10  
**Output:**  
3  
0  
4  
8

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

Contributor: Harsh Agarwal

New to competitive programming? Please see [How to Begin?](http://www.geeksforgeeks.org/how-to-begin-with-competitive-programming/)and [How to Pick a Category?](http://practice.geeksforgeeks.org/pickACategory.php)

<http://practice.geeksforgeeks.org/problems/find-smallest-missing-in-sorted/0>

#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

//code

int t;

scanf("%d", &t);

while(t-- > 0) {

int n,m;

scanf("%d %d", &n, &m);

int arr[n];

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

int elem;

scanf("%d",&elem);

arr[i] = elem;

}

bool entro = false;

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

if(arr[i] != i) {

printf("%d\n", i);

entro = true;

break;

}

}

if(!entro) {

printf("%d\n", arr[n-1] + 1);

}

}

return 0;

}