**Finding Number**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[Amazon](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Amazon&isCmp=1)

An array is bitonic if it is comprises of an increasing sequence of integers followed immediately by a decreasing sequence of integers.  
Given such a array, you need to find a element X in it and print its index.  
In case, X does not exist in the array print "OOPS! NOT FOUND" without quotes.  
Note: It is guaranteed that array consist of distinct elements. And array indexing is from 0.

Input:

First line will consist  a number T, the number of test cases.  
Each test case will then consist of two numbers N and X. N being the array size and X being the element to be searched for.  
Next line will consist of N space separated integers.

Output:

Print the required answer on separate lines.

Constraints:  
1<=T<=10  
1<=N<=200  
-100<=A[i]<=100

Example:

INPUT

1  
5 2  
1 2 7 3 4

OUTPUT

1

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=365>

#include <iostream>

#include <stdio.h>

#include <set>

#include <map>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

    // TODO code application logic here

    int t;

    scanf("%d", &t);

    while(t--) {

       int n,x;

       scanf("%d %d", &n, &x);

       std::vector<int> v;

       for(int i =0; i<n; i++) {

          int elem;

          scanf("%d", &elem);

          v.push\_back(elem);

        }

       std::vector<int>::iterator i = std::find(v.begin(), v.end(), x) ;

        int pos;

        if(i != v.end()) {

            pos = distance(v.begin(), i);

           //cout << pos << endl;

        }

        if(i == v.end()) {

            printf("OOPS! NOT FOUND**\n**");

        }else {

            printf("%d**\n**", pos);

        }

    }

}