**Missing number in array**

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

Given an array of size n-1 and given that there are numbers from 1-n with one missing, the missing number was to be found.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is N,size of array.  
The second line of each test case contains N-1 input C[i],numbers in array.  
  
**Output:**

Print the missing number in array.  
  
**Constraints:**

1 ≤ T ≤ 100  
1 ≤ N ≤ 1000  
1 ≤ C[i] ≤ 1000

**Example:**

Input  
2  
5  
1 2 3 5  
10  
1 2 3 4 5 6 7 8 10

Output  
4  
9

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=414>

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <vector>

using namespace std;

int main() {

    int T;

    scanf("%d", &T);

    //R represents rock, P represents paper, and S represents scissors.

    while(T--) {

        int N;

        scanf("%d", &N);

        int C[N-1];

        for(int i =0; i<N-1; i++) {

            scanf("%d", &C[i]);

        }

        bool marcas[N];

        memset(marcas, false, N);

        for(int i =0; i<N-1; i++) {

           marcas[C[i]] = true;

        }

        for(int i =1; i <= N; i++) {

           //printf("%d ", marcas[i]);

           if(!marcas[i]  ) {

               printf("%d", i);

               break;

            }

        }

        printf("**\n**");

    }

  system("pause");

 return 0;

}