**Element that appears once where every element occurs twice**

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

Given an array of integers, every element appears twice except for one. Find that single one in linear time complexity and without using extra memory.

**Input:**

The first line of input consists number of the test cases. The description of T test cases is as follows:

The first line of each test case contains the size of the array, and the second line has the elements of the array.

**Output:**

In each separate line print the number that appears only once in the array.

**Constraints:**

1 ≤ T ≤ 70  
1 ≤ N ≤ 100  
0 ≤ A[i] ≤ 100000

**Example:**

Input:

1  
11  
1 2 4 3 3 2 5 6 1 6 5

Output:

4

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=409>

#include <iostream>

#include <stdio.h>

#include <set>

#include <map>

using namespace std;

int main() {

    // TODO code application logic here

    int t;

    scanf("%d", &t);

    while(t-- > 0) {

        int n;

        scanf("%d", &n);

        int arr[n];

        for(int i =0; i<n; i++) {

          scanf("%d", &arr[i]);

        }

        std::map<int,int> m;

        for(int i =0; i<n; i++) {

            m[arr[i]]++;

        }

        int ans=-1;

        for(std::map<int,int>::iterator it = m.begin(); it != m.end() ; it ++) {

            if(it->second != 2) {

                ans = it->first;

                break;

            }

        }

        printf("%d**\n**", ans);

    }

}