**Find the element that appears once in sorted array**

Submissions: [16664](https://practice.geeksforgeeks.org/problem_submissions.php?pid=1832)  Accuracy:

51.1%

   Difficulty: [Basic](https://practice.geeksforgeeks.org/Basic/0/0/)   Marks: 1

Show Topic Tags   

[D-E-Shaw](https://practice.geeksforgeeks.org/company/D-E-Shaw/)[Ola Cabs](https://practice.geeksforgeeks.org/company/Ola%20Cabs/)

Given a sorted array**A**, size **N**, of integers; every element appears twice except for one. Find that element in linear time complexity and without using extra memory.

**Input:**  
The first line of input consists of **T**, the number of the test cases. T testcases follow. Each testcase contains two lines of input.  
The first line of each test case contains the size of the array, and the second line has the elements of the array.

**Output:**  
For each testcase, in a new line, print the number that appears only once in the array.

**Constraints:**  
1 ≤ T ≤ 100  
1 ≤ N ≤ 107  
0 ≤ A[i] ≤ 1017

**Example:**  
**Input:**  
1  
11  
1 1 2 2 3 3 4 50 50 65 65  
**Output:**  
4

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/find-the-element-that-appears-once-in-sorted-array/0#ExpectOP) option \*\*

[Author: kartik](https://auth.geeksforgeeks.org/user/kartik/practice/)

<https://practice.geeksforgeeks.org/problems/find-the-element-that-appears-once-in-sorted-array/0>

#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

int n;

scanf("%d", &n);

int arr[n];

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

scanf("%d", &arr[i]);

}

if(arr[0] != arr[1]) {

cout << arr[0] << endl;

continue;

}

if(arr[n-1] != arr[n-2]) {

cout << arr[n-1] << endl;

continue;

}

for(int i = 1; i<n-1; i++) {

if(arr[i] != arr[i-1] && arr[i] != arr[i+1]) {

cout << arr[i] << endl;

break;

}

}

}

return 0;

}