**Missing element of AP**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[binary search](http://www.practice.geeksforgeeks.org/tag-page.php?tag=binary%20search&isCmp=0)

Find the missing element from  an ordered array **A[ ]**, consisting of **N** elements representing an Arithmetic Progression (AP) .

**Input:**  
The first line of input contains an integer **T**denoting the number of test cases. Then **T**test cases follow.   
The first line of each test case contains an integer **N**, where **N** is the size of the array**A[ ]**.  
The second line of each test case contains**N** space separated integers of an Arithmetic Progression denoting elements of the array **A[ ]**.

**Note:** The series should have a missing element in between a perfect A.P. with no missing element will not be considered.

**Output:**  
Print out the missing element. 

**Constraints:**  
1 <= **T** <= 100  
2 <= **N**<= 10  
-50 <=**A[i]** <=50

**Examples :**

**Input:**  
3  
3   
2 10 14   
4   
-28 -21 -7 0   
5   
9 12 15 21 24  
  
**Output :**  
6  
-14  
18

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=896>

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class GFG {

public static void main(String[] args) {

// TODO code application logic here

Scanner sc = new Scanner(System.in);

int t = Integer.parseInt(sc.nextLine().trim());

while(t-- > 0) {

int n = Integer.parseInt(sc.nextLine().trim());

String[] input = sc.nextLine().trim().split(" ");

int[] A = new int[n];

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

A[i] = Integer.parseInt(input[i]);

}

if(n == 2) {

System.out.println((A[0] + A[1])/2);

continue;

}

int min\_dif = Integer.MAX\_VALUE;

for (int i = 0; i + 1 < n; i++)

{

min\_dif = Math.min(min\_dif, A[i + 1] - A[i]);

}

int ans = 0;

for (int i = 0; i+1 < n; i++)

{

if (A[i] + min\_dif != A[i + 1])

{

ans = A[i] + min\_dif;

break;

}

}

System.out.println(ans);

}

}

}