**Count the Zeros**

[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)

Given an array of size **N** consisting of only 0's and 1's ,which is**sorted** in such a manner that all the 1's are placed first and then they are followed by all the 0's. You have to find  the**count** of all the 0's.

**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 all 1's follwed by all the 0's, denoting elements of the array **A[ ]**.

**Output:**  
Print out the number of 0's in the array. 

**Constraints:**  
1 <= **T** <= 100  
1 <= **N** <= 30  
0 <= **A[i]** <= 1

**Example :**

**Input:**  
3  
12  
1 1 1 1 1 1 1 1 1 0 0 0   
5  
0 0 0 0 0  
6  
1 1 1 1 1 1

**Output:**  
3  
5  
0

**Expected Complexity:**

O(logN)

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=897>

**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());

**while**(t-->0) {

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

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

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

*System.out.println(input[i]);*

*}\*/*

            ArrayList<Integer> lista = **new** ArrayList<Integer>();

**for**(**int** i =0; i<n; i++) {

                lista.add(Integer.parseInt(input[i]));

            }

**if**(lista.indexOf(0) > - 1)

                System.out.println(n - lista.indexOf(0) );

**else**

                System.out.println(0 );

        }

    }

}