**Day 79 coding Statement :**

You are given a binary string *S* of length *N*. You can perform the following operation on *S*:

* Pick any set of indices such that no two picked indices are adjacent.
* Flip the values at the picked indices (i.e. change 0 to 1 and 1 to 0).

For example, consider the string *S*=1101101.  
If we pick the indices {1,3,6}, then after flipping the values at picked indices, we will get 1?10?110?1→0111111.  
Note that we cannot pick the set {2,3,5} since 2 and 3 are adjacent indices.

Find the **minimum** number of operations required to convert **all** the characters of *S* to 0.

**Input Format**

* The first line contains a single integer *T* - the number of test cases. Then the test cases follow.
* The first line of each test case contains an integer *N* - the length of the binary string *S*.
* The second line of each test case contains a binary string *S* of length *N*.

**Output Format**

For each test case, output the **minimum** number of operations required to convert all the characters of *S* to 0.

**Sample Input**

3

6

101001

5

00000

3

111

**Sample Output**

1

0

2

import java.util.Scanner;

public class RatanPrajapati\_day79 {

    public static void main(String[] args) throws java.lang.Exception {

        Scanner sc = new Scanner(System.in);

        int T = sc.nextInt();

        while (T-- > 0) {

            int size = sc.nextInt();

            String st = sc.next();

            int count = 0, t = 0;

            for (int i = 0; i < size - 1; i++) {

                if (st.charAt(i) == '1') {

                    count++;

                    if (st.charAt(i) == st.charAt(i + 1)) {

                        t++;

                    }

                }

            }

            if (st.charAt(size - 1) == '1' && count == 0) {

                System.out.println('1');

                continue;

            }

            if (count == 0) {

                System.out.println(count);

            } else {

                if (t == 0) {

                    System.out.println('1');

                } else {

                    System.out.println('2');

                }

            }

        }

    }

}