

B. Binary Typewriter

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes

You are given a binary string s of length n and a typewriter with two buttons: 0 and 1. Initially, your finger is on the button 0. You can do the following two operations:

1. Press the button your finger is currently on. This will type out the character that is on the button.
2. Move your finger to the other button. If your finger is on button 0, move it to button 1, and vice versa.

The *cost* of a binary string is defined as the minimum number of operations needed to type the entire string.

Before typing, you may reverse at most one substring* of s . More formally, you can choose two indices $1 \leq l \leq r \leq n$, and reverse the substring $s_l \dots s_r$, resulting in the new string

$s_1 s_2 \dots s_{l-1} s_r s_{r-1} \dots s_l s_{r+1} \dots s_n$.

Your task is to find the minimum possible cost among all strings obtainable by performing at most one substring reversal on s .

* A string a is a substring of a string b if a can be obtained from b by the deletion of several (possibly, zero or all) characters from the beginning and several (possibly, zero or all) characters from the end.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the length of the binary string s .

The second line of each test case contains a binary string $s_1 s_2 \dots s_n$ ($s_i = 0$ or $s_i = 1$) — the characters of the binary string s .

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the minimum cost of string s after performing at most one substring reversal.

Example

input	Copy
6	
3	
000	
3	
111	
3	
011	
3	
100	
5	
10101	
19	
1101010010011011100	
output	Copy
3	
4	
4	
4	
8	
29	

Note

Codeforces Round 1019 (Div. 2)

Finished

Practice



→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++23 14.2 (64 bit, ms) ▼

Choose file: No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
316630333	Apr/22/2025 06:51	Accepted
316569595	Apr/21/2025 18:43	Time limit exceeded on pretest 4
316568814	Apr/21/2025 18:42	Wrong answer on pretest 1
316563431	Apr/21/2025 18:31	Wrong answer on pretest 2
316557684	Apr/21/2025 18:20	Time limit exceeded on pretest 4

→ Problem tags

greedy math

No tag edit access

→ Contest materials

- Announcement (en) 

In the first test case, we can choose not to reverse any substrings. We can do operation 1 three times to type 000.

In the second test case, we can choose not to reverse any substrings. We can do operation 2 to move our finger to button 1. Then, we do operation 1 three times to type 111.

In the third test case, we can choose not to reverse any substring. We can do operation 1 to type 0. Then, we do operation 2 to move our finger to button 1. Finally, we do operation 1 two times to type 11, resulting in the final string 011 using only 4 operations.

In the fourth test case, we can reverse the substring $s_{1..3}$, resulting in the string 001. We can do operation 1 two times to type 00. Then we do operation 2 to move our finger to button 1. Finally, we do operation 1 once to type 1, resulting in the final string 001 using only 4 operations.

In the fifth test case, we can reverse the substring $s_{2..3}$, resulting in the string 11001. The cost of the string is 8 as we can do the following sequence of operations:

- Do operation 2 to move our finger to button 1.
- Do operation 1 two times to type 11.
- Do operation 2 to move our finger to button 0.
- Do operation 1 two times to type 00.
- Do operation 2 to move our finger to button 1.
- Do operation 1 once to type 1.

In the sixth test case, we can reverse the substring $s_{5..17}$, resulting in the string 110111011001001000. It can be proven that the minimum number of operations needed to type the binary string is 29.

