1-2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Let n be a positive integer. Luigi has a binary string* s of length n. He's initially at position 1, and can advance from position i to $i+1$ for $1 \leq i \leq n$ if $s_i = \texttt{1}$.

To help him cross, he can overwrite some (possibly none or all) of the characters in the string with `2`'s. During the crossing, if he is at position i with $s_i = \texttt{2}$, he can not only move to position $i+1$, but also has the choice to do the following operation before moving:

- For all characters in the string that aren't `2`'s, flip the `0`'s to `1`'s and vice versa.

Compute the minimum number of `2`'s he can place in the string to be able to reach position $n+1$. It can be shown that the task is always possible.

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 an integer n ($1 \leq n \leq 2 \cdot 10^5$) — the length of the string.

The second line of each test case contains a string s of length n consisting of characters `0` or `1`.

Output

For each test case, output a single integer — the minimum number of `2`'s Luigi needs to place to reach position $n+1$.

Example

standard input	standard output
5 1 0 2 01 3 010 5 00110 4 1111	1 1 2 3 0

*A binary string is a string whose characters are either `0` or `1` .