

# Occupy the Cities

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:          1 second  
Memory limit:        512 megabytes

JB is playing a game. There are  $n$  cities in the game, numbered as  $1, 2, \dots, n$ . The  $i$ -th city and the  $j$ -th city are adjacent if and only if  $i = j - 1$  or  $i = j + 1$ . Initially, some of the cities are occupied by JB.

The game runs in rounds. At the beginning of a round, each occupied city can mark at most one adjacent unoccupied city as the target of attack. At the end of the round, all the attack targets marked become occupied. The game ends when all the cities are occupied.

JB wants to occupy all the cities in minimum rounds. Can you help him?

## Input

There are multiple test cases. The first line of the test case contains a positive integer  $T$ , indicating the number of test cases. For each test case:

The first line contains an integer  $n$  ( $1 \leq n \leq 10^6$ ), indicating the number of cities.

The next line contains a string  $s$  of length  $n$ . It's guaranteed  $s$  only contains '0' and '1'. The  $i$ -th character describes the initial state of the  $i$ -th city: if  $s_i = '1'$ , the  $i$ -th city is occupied by JB initially. Otherwise, the  $i$ -th city is not occupied initially.

It's guaranteed that the sum of  $n$  over all the test cases doesn't exceed  $10^6$ . It's also guaranteed that there is at least one '1' in  $s$ .

## Output

For each test case, output one line, containing the minimum number of rounds to occupy all the cities.

## Example

standard input	standard output
5	2
3	2
010	4
4	0
0100	1
7	
0001000	
5	
11111	
6	
010101	

## Note

For the second test case, the best way is  $0100 \rightarrow 0110 \rightarrow 1111$ .