String of CCPC

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

BaoBao has just found a string s of length n consisting of 'C' and 'P' in his pocket. As a big fan of the China Collegiate Programming Contest, BaoBao thinks a substring $s_i s_{i+1} s_{i+2} s_{i+3}$ of s is "good", if and only if $s_i = s_{i+1} = s_{i+3} =$ 'C', and $s_{i+2} =$ 'P', where s_i denotes the i-th character in string s. The value of s is the number of different "good" substrings in s. Two "good" substrings $s_i s_{i+1} s_{i+2} s_{i+3}$ and $s_j s_{j+1} s_{j+2} s_{j+3}$ are different, if and only if $i \neq j$.

To make this string more valuable, BaoBao decides to buy some characters from a character store. Each time he can buy one 'C' or one 'P' from the store, and insert the character into any position in s. But everything comes with a cost. If it's the i-th time for BaoBao to buy a character, he will have to spend $(i-1)$ units of value.

The final value BaoBao obtains is the final value of s minus the total cost of all the characters bought from the store. Please help BaoBao maximize the final value.

Input

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

The first line contains an integer n ($1 \leq n \leq 2 \times 10^5$), indicating the length of string s.

The second line contains the string s ($|s| = n$) consisting of 'C' and 'P'.

It's guaranteed that the sum of n over all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer, indicating the maximum final value BaoBao can obtain.

Example

standard input	standard output
3 3 CCC 5 CCCCP 4 CPCP	1 1 1

Note

For the first sample test case, BaoBao can buy one 'P' (cost 0 value) and change s to "CCPC". So the final value is $1 - 0 = 1$.

For the second sample test case, BaoBao can buy one 'C' and one 'P' (cost $0 + 1 = 1$ value) and change s to "CCPCCPC". So the final value is $2 - 1 = 1$.

For the third sample test case, BaoBao can buy one 'C' (cost 0 value) and change s to "CCPCP". So the final value is $1 - 0 = 1$.

It's easy to prove that no strategies of buying and inserting characters can achieve a better result for the sample test cases.