Problem C. Palindromic and Balanced

Input file: `standard input`
Output file: `standard output`

Yuki discovered that a palindromic bracket sequence cannot be a balanced bracket sequence, so she designed another way to define a palindromic balanced bracket sequence.

Yuki defines a **palindromic bracket sequence** according to the following rules:

- The empty string is a palindromic bracket sequence.
- `(` and `)` are palindromic bracket sequences.
- If a bracket sequence s is a palindromic bracket sequence, then `(`s`(` and `)`s`)` are palindromic bracket sequences.

Yuki defines a **balanced bracket sequence** according to the following rules:

- The empty string is a balanced bracket sequence.
- If a bracket sequence s is a balanced bracket sequence, then `(`s`)` is a balanced bracket sequence.
- If bracket sequences s and t are both balanced bracket sequences, then st (the concatenation of the two bracket sequences) is a balanced bracket sequence.

For a bracket sequence $s = s_1 \dots s_n$, Yuki defines s as a **palindromic balanced bracket sequence** if and only if:

- $s_2 \dots s_{n-1}$ is a palindromic bracket sequence.
- $s_1 \dots s_n$ is a balanced bracket sequence.

Specifically, the empty string and `()` are also palindromic balanced bracket sequences.

For example, `(())()` and `()()(()())` are palindromic balanced bracket sequences, while `((()))` and `()()(())` are not.

Now, Yuki has a bracket sequence s of length n, and she wants to find the longest subsequence* of s that is a palindromic balanced bracket sequence. However, Yuki does not know how to do this, so you need to help her find the length of the longest such subsequence.

*: A sequence a is a subsequence of sequence b if and only if a can be obtained by deleting zero or more elements from b; specifically, the empty sequence is a subsequence of any sequence.

Input

This problem contains multiple test cases.

The first line contains a positive integer t ($1 \le t \le 5000$), representing the number of test cases.

For each test case:

- The first line contains a positive integer n ($1 \le n \le 5000$).
- The second line contains a bracket sequence s of length n ($s_i \in \{(,)\}$).

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output

For each test case, output a single line containing an integer representing the length of the longest subsequence that satisfies the condition.

Example

standard input	standard output
3 5 (()((7)))((((8 ())(()()	2 0 6

Note

For the first test case:

- The longest subsequences satisfying the condition are $s_1s_3 =$ `()` and $s_2s_3 =$ `()`, so the answer is 2.

For the second test case:

- The longest subsequence satisfying the condition is the empty string, so the answer is 0.

For the third test case:

- The longest subsequence satisfying the condition is $s_1s_2s_4s_5s_6s_8 =$ `()(())`, so the answer is 6.