bracketflip

We have a (not necessarily correct) bracket sequence *S* of length *N*.

A pair (*L*, *R*) is *good* if the following conditions hold:

- $1 <= L <= R <= N$
- After flipping the parities of the brackets (open to close, close to open) in the range [*L*, *R*] of string S, S becomes a correct bracket sequence

A correct bracket sequence is defined as:

- *"" or*
- (*x*) where *x* is a correct bracket sequence, *or*
- *xy*, where *x* and *y* are correct bracket sequences

Output the number of *good* pairs.

Input

The first line of input contains a single integer *N*, the length of the bracket sequence.

The second line of input contains bracket sequence *S*.

Output

Output a single integer, the number of *good* pairs.

Scoring

For all testcases, it is guaranteed that:

- $1 <= N <= 10^6$
- *S* will only contain "(" and ")", and will be of length *N*.

Subtask	Score	*N*
1	10	$1 <= N <= 5 \cdot 10^2$
2	20	$1 <= N <= 2 \cdot 10^3$
3	70	No additional constraints
4	0	Sample Testcases

Examples

standard input	standard output
4 ((((	1
6)()())	3

Note

In the first example, the only *good* pair is (3, 4).

In the second example, there are 3 *good* pairs:

- (1, 1), S becomes (()())
- (1, 3), S becomes ()(())
- (1, 5), S becomes ()()()