Correct Sequence

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	524 megabytes

A bracket sequence is considered correct if you can get a correct mathematical formula by inserting '+' and '1' in it. Some examples of correct bracket sequences are: ()()(), ((())). While ()))() is not a correct bracket sequence.

In a reorder operation you take a substring of the bracket sequence and reordering the symbols in it in any arbitrary way. If the substring has a length 'l', the cost of this operation is also 'l'. Reordering in no way changes the number of opening and closing brackets in the substring. So if you reorder the string '))((' to get ')()(' or '(())' the cost of this operation will be 4.

Given an input bracket sequence, you need to find out the minimum cost of reordering the string into a correct bracket sequence. If it is not possible to get a correct sequence the output should be -1.

Input

The first line contains an integer *'i'*, the number of brackets in the sequence.
$(1 \leq i \leq 10^6)$.
The next line contains the bracket sequence of length i.

Output

Output the lowest possible cost of reordering the whole string to get a correct bracket sequence. If such a sequence is not possible then the output should be -1.

Example

standard input	standard output

8))((())(	6

Explanation

In the example we can firstly reorder the segment from first to the fourth character, replacing it with "()()", the whole sequence will be "()()())(". And then reorder the segment from the seventh to eighth character, replacing it with "()". In the end the sequence will be "()()()()", while the total time spent is 4+2=6.