Tweedledum's Nested Brackets

Time limit: *1000 ms*
Memory limit: *256 MB*

The nested level of a bracket sequence S is defined as the number of unmatched left brackets that may exist in a prefix of S. For example, the nested levels of `((()))`, `(())()`, and `(()(()))()` are 3, 2, and 3 respectively.

Tweedledum has a string X consisting of a valid sequence of round brackets. In how many ways can you create a maximally nested bracket sequence by removing a continuous substring Y from X and re-inserting it into X at any location? The substring Y has to be a valid bracket sequence.

A valid bracket sequence is defined as follows:

- `()` is a valid bracket sequence;
- If S is a valid bracket sequence, then `(` S `)` is also a valid bracket sequence;
- If Q and R are two valid bracket sequences, then the concatenation of them, $S = QR$, is also a valid bracket sequence.

Two ways are considered different if you either choose a different substring Y, or if you re-insert Y at a different position. Two equal substrings occurring at two different positions in X are considered two different choices of Y. Re-inserting Y at its original position counts if it is a maximally nested bracket sequence.

Standard input

The input has a single line consisting of the bracket sequence X.

Standard output

Output the number of ways in which you can create a maximally nested bracket sequence, by choosing and re-inserting a substring Y from X.

Constraints and notes

- $2 \leq |X| \leq 10^6$

Input	Output	Explanation
`(())()`	2	`(())` on the left can be moved into `()` on the right, or vice versa.
`((()))`	3	`()`, `(())`, or `((()))` can all be re-inserted at their original positions.
`()()()`	8	There are 6 ways to choose one pair of brackets and insert it into either of the other two pairs and there are 2 ways to choose two consecutive pairs of brackets and insert them into the remaining pair of brackets.