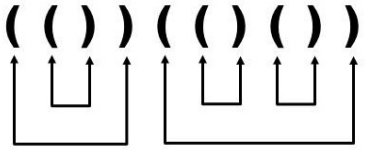
Problem B
Alternative Bracket Notation

Balanced closed bracket or parenthesis statements are ones where each opening bracket is matched with a closed bracket later in the string.



Notice how each closed parenthesis matches to the most recent unmatched open parenthesis.

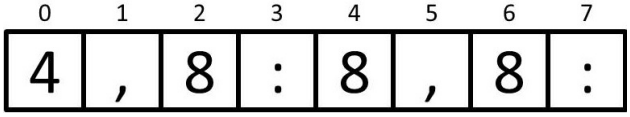
Define an alternative bracket notation as follows: each bracket pair corresponds to a header in the form of "*start*,*end*:" where *start* and *end* are indices of the new string itself! The index *start* is the index of the character immediately after the ':', and *end* is the index past the last header corresponding to the last bracket pair contained in this bracket pair. By taking a substring(*start*, *end*) of the new notation, you get an alternative bracket sequence describing all of the pairs of brackets contained by the brackets corresponding to the "*start*,*end*:"! Since an empty pair of brackets has nothing inside, in their header, *start* and *end* will be the same.

Each index takes up as many characters in the string as they do when they are base 10 numbers. (For example, the index 42 will take up 2 characters). The indices in the new string start from 0. All of the indices found in the alternative bracket notation string are absolute indices from the beginning of the new string.

Consider this parenthetical statement: `(())`

Here is it, in our new, alternate bracket notation: `4,8:8,8:`

In this example, there are two sets of matching parenthesis, the outer one and the inner one. The outer one appears before the inner one, since the start bracket appears first. So, the header for the outer brackets will appear before the header for the inner bracket. The header `4,8:` represents the outer bracket, while the header `8,8:` represents the inner bracket. The substring from the 4th character to 7th character is `8,8:`, which represents what is contained inside the outer bracket. Note that `5,11:11,11:` could also be a legitimate alternate notation, but we want the shortest one, which is why `4,8:8,8:` is the correct answer.



Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. The input will consist of a single line, containing a string s, which consists only of open and closed parentheses. The string s will be between 2 and 4,000 characters long. There will be no spaces. The string s is guaranteed to be balanced.

Output

Output the string s in our new alternative bracket notation. If there's more than one way to represent s in the new notation, choose the shortest representation, which will be unique.

Sample Input 1	Sample Output 1
`(())`	`4,8:8,8:`

Sample Input 2	Sample Output 2
`()`	`4,4:`

Sample Input 3	Sample Output 3
`((())(()))()`	`5,29:11,17:17,17:23,29:29,29:35,35:`