# K: Uniform Subtrees

Define a *Uniform* tree as one in which all of the nodes at a given level (i.e. distance from the root) have the same degree (i.e. number of children). Since all of the nodes at a given level have the same number of children, a uniform tree can be represented as simply a list of integers, indicating the number of children at each level. For example, the list [2 3 5 0] represents a tree where the root has 2 children, each child of the root has 3 children, each grandchild has 5 children, and each great-grandchild has no children, and is therefore a leaf.

For the purposes of this problem, redefine *Subtree* as a connected subgraph of a tree that includes the tree's root. This is a bit different than the typical definition of *Subtree*.

Given a description of a tree, find all of the unique *Uniform Subtrees* of that tree. For example, here is a tree and all of its unique uniform subtrees:
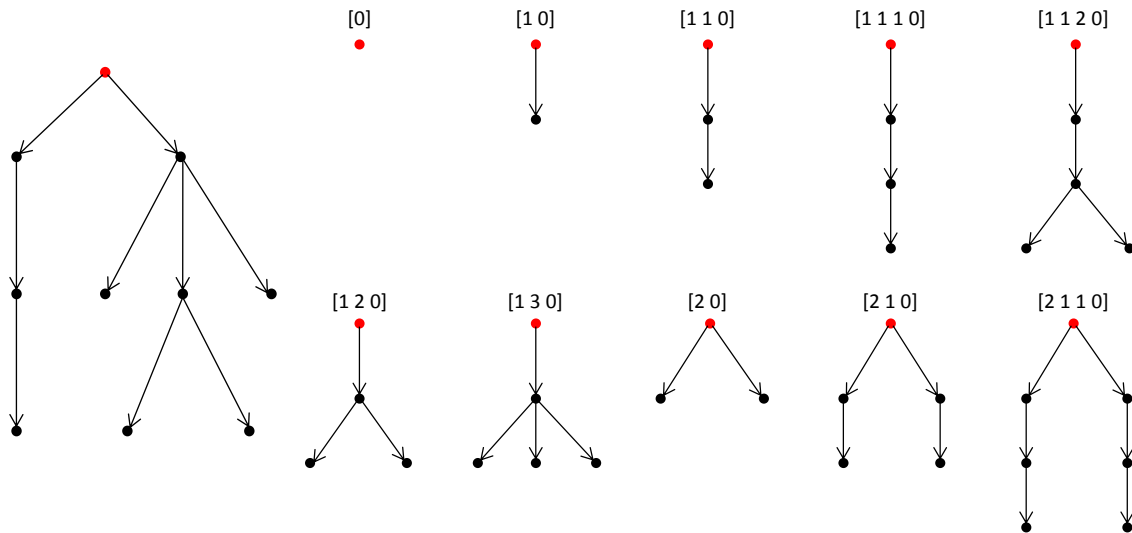
Input

There will be several test cases in the input. Each test case will will consist of a single tree, represented as a single string on one line. The string will be a sequence of matched opening and closing parentheses. Each matched pair represents a node, and the string between represents its children. There will not be more than 4,000 nodes in the tree. There will be no whitespace, or any other characters, in the string. The input will end with a line with a single 0.

Output

For each test case, output every unique uniform subtree of the given tree as a list of integers, one subtree (and thus one list) per line. Print a single space between integers, and no spaces anywhere else. Do not print any blank lines between lists, or between test cases. Print the lists for a given test case sorted by the first element, then the second, then the third, and so on.

Sample Input	Sample Output
(((( ))) (( ) (( ) ( )) ( ))) (( )) 0 (Note: the first sample input & output case corresponds to the pictured tree and subtrees.)	0 1 0 1 1 0 1 1 1 0 1 1 2 0 1 2 0 1 3 0 2 0 2 1 0 2 1 1 0 0 1 0