

Input terms are defined inductively as follows:

1. X is a term;
2. if A and B are terms, then (A B) is also a term;
3. no other terms.

The terms are transformed by the rewriting rule: the pattern  $((X A) B) C$ , where A, B, and C are terms, is replaced by  $((A C)(B C))$ .

Reducing a term means applying this rule. A term is called reduced if it contains no subterm that can be further reduced. A term is considered as a subterm of itself, too.

Write a program that reads input terms and writes the corresponding reduced terms.

## Input

The input file will contain several lines with one term per line. Each line will contain no more than 80 characters. The input terms will be syntactically correct. Possible input characters are 'X', '(', and ')' (without blanks in between). The input data will be terminated by a line containing a single character '0' (zero).

## Output

For each input term, write its corresponding reduced term in a line, followed by a line containing digits in positions in which output term has parentheses, and an empty line. A term without parentheses must be followed by two empty lines. These digits mark the nesting level of parentheses in the corresponding positions. It is guaranteed that the maximum nesting level is less than 9, and that the number of reductions is finite, and that the output lines will be shorter than 80 characters. The top level is denoted by 0. Possible characters in the output terms are also 'X', '(', and ')

## Sample Input

```
((XX)X)X
((XX)(XX))X
(XX)
0
```

## Sample Output

```
((XX)(XX))
01 11 10
((XX)((XX)X))
01 112 2 10
(XX)
0 0
```