

An algebraic expression can be represented by a tree. We can evaluate the expression by traversing the corresponding tree. The tree representation of an algebraic expression is not unique. Consider an example : '1+2+3\*4'. It can be represented by the following two trees :

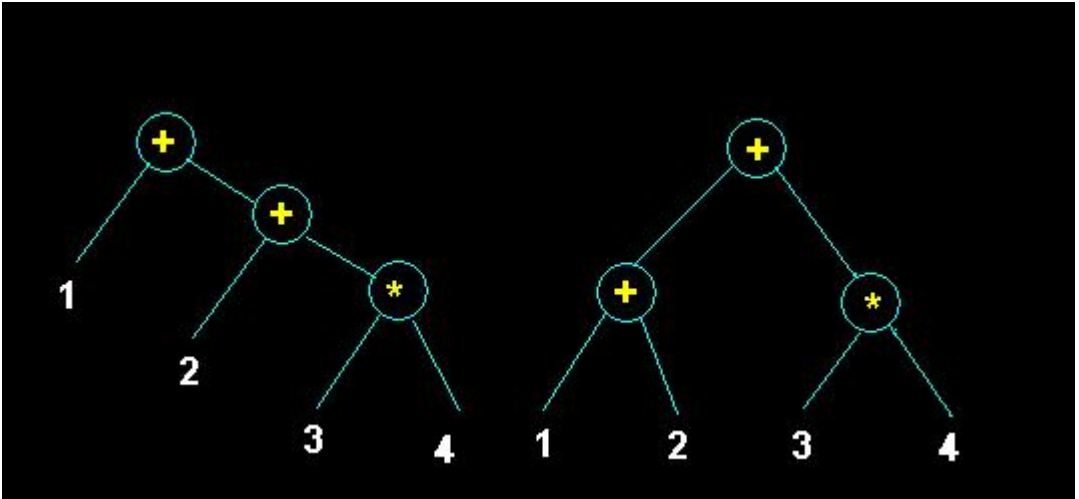


Fig: Tree representations of '1+2+3\*4'

If you closely look, you will see that the precedence of '+' and '\*' is maintained and the order of operands is not changed. In the first representation, the evaluation order will be :  $3*4 = 12$ ;  $2+12 = 14$ ;  $1+14 = 15$ . And in the 2nd representation, the expression will be evaluated like :  $1+2 = 3$ ;  $3*4 = 12$ ;  $3+12 = 15$ . Both representations produce correct result.

In this problem, we will consider simple algebraic expressions consist of digits, '+', '\*' and parentheses. The expression will follow the normal algebraic rule. You have to find out the number of tree representations which will correctly evaluate the expression.

### Input

Each line in the input file contains an algebraic expression consists of single digit numbers, parentheses, addition and multiplication operators. The expression will be syntactically correct and its length will not exceed 130. There will be atleast one operator and not more than 35 operators in the expression. Input is terminated by EOF.

### Output

For each expression you should print the number of tree representations which will correctly evaluate the expression in a line.

### Sample Input

```
1+2+3+4
(1+2)+(3+4)
1+2+3*4
1+2+(3*4)
```

### Sample Output

```
5
1
2
2
```