UVa Online Judge

12942 Sub-expression Counting

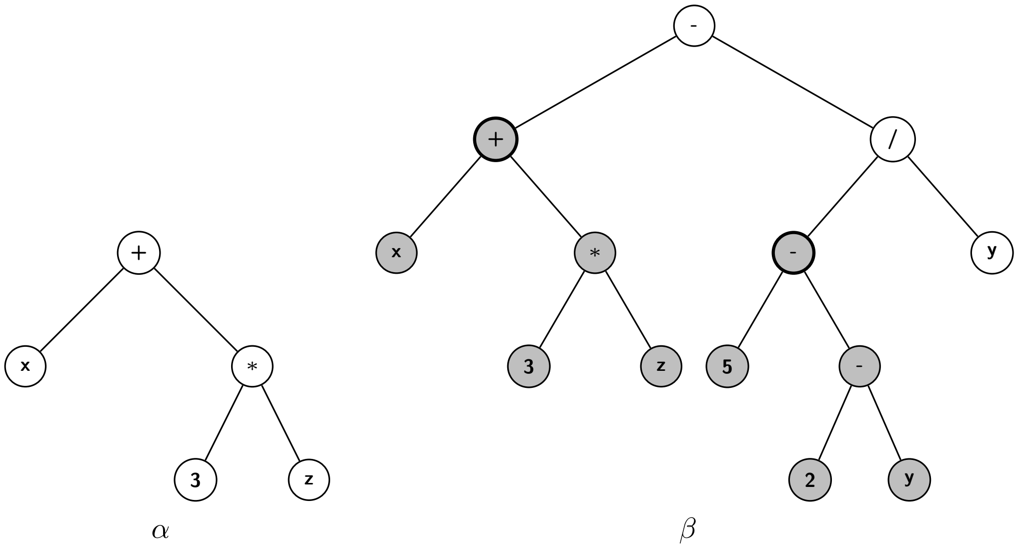
Suppose we want to search arithmetic expressions for sub-expressions of certain shape. We are considering only fully parenthesized expressions with binary operators, numerical constants, and variables, as defined in the following BNF-like notation:

$$\begin{array}{rcl} \langle expr \rangle & ::= & \langle var \rangle \mid \langle num \rangle \mid \texttt{(}\langle expr \rangle \langle binop \rangle \langle expr \rangle\texttt{)} \\ \langle var \rangle & ::= & \texttt{a} \mid \texttt{b} \mid \cdots \mid \texttt{z} \\ \langle num \rangle & ::= & \langle digit \rangle \mid \langle digit \rangle \langle num \rangle \\ \langle digit \rangle & ::= & \texttt{0} \mid \texttt{1} \mid \cdots \mid \texttt{9} \\ \langle binop \rangle & ::= & \texttt{+} \mid \texttt{-} \mid \texttt{*} \mid \texttt{/} \end{array}$$

For example, consider the arithmetic expressions α and β defined as follows:

$$\begin{array}{rcl} \alpha & : & (x + (3 * z)) \\ \beta & : & ((x + (3 * z)) - ((5 - (2 - y))/y)). \end{array}$$

The syntax tree associated to each one of these arithmetic expressions is shown below:



We want to report *all* nodes v in β such that the sub-tree rooted at v is structurally identical to α, ignoring all labels in the nodes. In this case, there are 2 such nodes because: *(i)* expression α is a sub-expression of β and *(ii)* sub-expression $(5 - (2 - y))$ of β has the same tree structure as α. The corresponding sub-trees have been shaded in the syntax tree of β depicted above.

Your task is to write an efficient computer program that, given inputs α and β, computes the number of nodes v in β such that the sub-tree rooted at v is structurally identical to α.

Input

The input consists of several test cases. Each test case consists of two lines: the first line describes the expression α and the second one the expression β. You can assume that $1 \leq |\alpha| \leq 400000$ and $1 \leq |\beta| \leq 400000$, and that these expressions do not contain any blanks.

Output

For each test case, output the number of nodes v in β such that the sub-tree rooted at v is structurally identical to α.

Sample Input

```
1978
((x+0)+z)
(x+(3*z))
((x+(3*z))-((5-(2-y))/y))
```

Sample Output

```
3
2
```