Finite Domain Constraints

Problem Statement

A finite domain constraint is a linear equality or inequality over integer variables. Each variable has a known and finite domain (set of possible values). A finite domain constraint is satisfiable if each variable can be assigned a value from its domain in a way that makes the equality or inequality true. For example, given the finite domain constraint:

$(X + Y) = 5$

with the domain of X as {1, 2, 3, 4} and the domain of Y also as {1, 2, 3, 4}, the constraint is satisfied by assigning 2 to X and 3 to Y. Note that the constraint can also be satisfied by assigning 3 to X and 2 to Y, 1 to X and 4 to Y, and 4 to X and 1 to Y. Hence, there are 4 distinct variable assignments that satisfy the constraint. If we change the constraint to the inequality:

$(X + Y) < 5$

with the same domains, there are now 6 distinct variable assignments that satisfy the constraint: (X = 1, Y = 1), (X = 1, Y = 2), (X = 1, Y = 3), (X = 2, Y = 1), (X = 2, Y = 2), and (X = 3, Y = 1).

Given a finite domain constraint and domains for each variable, your task is to determine how many distinct variable assignments satisfy the constraint.

Input Format

The input is made up of multiple test cases. Each test case ends with a 0 on a line by itself.

The first line in each test case is an integer n, 1 <= n <= 10. n is the number of variables in the finite domain constraint.

The next n lines are each of the form:

[Variable_Name] [Low] [High]

where:

- [Variable_Name] is the variable name
- [Low] is the lower bound of the variable's domain (inclusive)
- [High] is the upper bound of the variable's domain (inclusive)

For example:

```
X 1 4
```

means that the domain of X is {1, 2, 3, 4}. Variable names are not repeated within the same test case.

The next line of each test case is a finite domain constraint over those variable names.

Notes:

- The lower and upper bounds are integers from 1 to 10 (inclusive).

- The constraint will contain exactly one occurrence of either = or <. Note that the operator <= will never appear in a constraint.
- The constraint can additionally contain parentheses, +, -, *, variable names and integer constants.
- The constraint will be fully parenthesized so that precedence (order of operations) is not needed.
- Each distinct symbol in the constraint will be separated from other symbols by exactly one space.
- Each variable occurs exactly once in the constraint.
- The number of integer constants is less than or equal to the number of variables plus one, and each constant is between -100 and 100 (inclusive).
- When the * operator is used in a constraint, one operand will be a constant and the other will be a variable.
- You can assume that the constraint is syntactically correct.

Output Format

For each test case, your program should output, on a line by itself, the number of distinct variable assignments that satisfy the finite domain constraint with the given variable domains.

Sample Input

```
2
X 1 4
Y 1 4
( X + Y ) = 5
0
3
X 1 3
Y 2 4
Z 1 4
( ( X + Y ) - ( 2 * Z ) ) < 5
0
4
W 2 5
X 1 3
Y 2 9
Z 1 4
( ( X + ( Y - 3 ) ) + ( -2 * Z ) ) = ( ( 5 * W ) - 10 )
0
```

Sample Output

```
4
35
15
```

Explanation

The first test case in the sample input corresponds to the first example given in the problem statement.