Problem E
"Chemical reactions"

Input file `chem.in`

Output file `chem.out`

Bill teaches chemistry in the school and has prepared a number of tests for his students. Each test has a chemical formula and a number of possible reaction outcomes that his students are to choose one correct from. However, Bill wants to make sure that he has not made any typos while entering his tests into a computer and that his students won't easily throw away wrong answers simply by counting a number of chemical elements on the left and on the right side of the equation, which should be always equal in a valid reaction.

You are to write a program that will help Bill. The program shall read the description of the test for the students that consists of the given left side of the equation and a number of possible right sides, and determines if the number of chemical elements on each right side of the equation is equal to the number of chemical elements on the given left side of the equation.

To help you, poor computer folks, that are unaware of the complex world of chemistry, Bill has formalized your task. Each side of the equation is represented by a string of characters without spaces, and consists of one or more chemical sequences separated by a '+' (plus) characters. Each sequence has an optional preceding integer multiplier that applies to the whole sequence and a number of elements. Each element is optionally followed by an integer multiplier that applies to it. An element in this equation can be either distinct chemical element or a whole sequence that is placed in round parenthesis. Every distinct chemical element is represented by either one capital letter or a capital letter that is followed by a small letter.

Even more formally, using notation that is similar to BNF, we can write:

- <formula> ::= [<number>] <sequence> { '+' [<number>] <sequence> }
- <sequence> ::= <element> [<number>] { <element> [<number>] }
- <element> ::= <chem> | '(' <sequence> ')'
- <chem> ::= <uppercase_letter> [<lowercase_letter>]
- <uppercase_letter> ::= 'A'..'Z'
- <lowercase_letter> ::= 'a'..'z'
- <number> ::= '1'..'9' { '0'..'9' }

Every distinct chemical element is said to occur in the given formula for some total number X, if X is the sum of all separate occurrences of this chemical element multiplied to all numbers that apply to it. For example, in the following chemical formula:

```
C2H5OH+3O2+3(SiO2)
```

- `C` occurs for a total of 2 times.
- `H` occurs for a total of 6 times (5 + 1).
- `O` occurs for a total of 13 times (1 + 3*2 + 3*2).
- `Si` occurs for a total of 3 times.

All multipliers in the formula are integer numbers that are at least 2 if explicitly specified and are 1 by default. Each chemical formula is at most 100 characters long, and every distinct chemical element is guaranteed to occur for a total of no more than 10000 times in each formula.

Input

The first line of the input file represents a chemical formula that is to be tested as the left side of the equation. The second line of the input file contains a single integer number N (1 = N = 10), which is the number of right sides of the equation that are to be tested. Each one of the following N lines represents one such formula.

Output

You are to write to the output file N lines - one line per each possible answer of the chemical test for Bill's students that is given in the input file. For each right-hand side formula that is encountered in the input file, write to the output file:

```
<left_formula>==<right_formula>
```

if the total number of occurrences of each distinct chemical element on the left-hand side equals to the total number of occurrences of this chemical element on the right-hand side. Otherwise write:

```
<left_formula>!=<right_formula>
```

Here `<left_formula>` must be replaced exactly (character by character) with the original left-hand side formula as it is given in the first line of the input file, and `<right_formula>` must be replaced exactly with each right-hand side formula as they are given in the input file. Do not place any spaces in the lines you write to the output file.

Sample input

```
C2H5OH+3O2+3(SiO2)
7
2CO2+3H2O+3SiO2
2C+6H+13O+3Si
99C2H5OH+3SiO2
3SiO4+C2H5OH
C2H5OH+3O2+3(SiO2)+Ge
3(Si(O)2)+2CO+3H2O+O2
2CO+3H2O+3O2+3Si
```

Output for the sample input

```
C2H5OH+3O2+3(SiO2)==2CO2+3H2O+3SiO2
C2H5OH+3O2+3(SiO2)==2C+6H+13O+3Si
C2H5OH+3O2+3(SiO2)!=99C2H5OH+3SiO2
C2H5OH+3O2+3(SiO2)==3SiO4+C2H5OH
C2H5OH+3O2+3(SiO2)!=C2H5OH+3O2+3(SiO2)+Ge
C2H5OH+3O2+3(SiO2)==3(Si(O)2)+2CO+3H2O+O2
C2H5OH+3O2+3(SiO2)!=2CO+3H2O+3O2+3Si
```

Note

The sample input and output do not contain digit '0' because it looks the same as the symbol for the chemical element oxygen. The actual tests may contain any allowed characters.