Problem F. Fygon

Input file:	`fygon.in`
Output file:	`fygon.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Frederick is a young programmer. He participates in all programming contests he can find and always uses his favorite programming language Fygon. Unfortunately, he often receives Time Limit Exceeded outcome, even when his algorithm is asymptotically optimal. That's because the Fygon interpreter is very slow. Nevertheless, Frederick likes Fygon so much, that he uses non-asymptotical optimizations to fit the solution into time limit. To make it easier, he asks you to write a program, which will be able to estimate the exact number of operations that his Fygon program makes.

For simplicity, we will assume that Fygon has only two statements. The first statement is `lag`. It substitutes almost any other statement. The second statement is a `for` loop:

```
for <variable> in range(<limit>):
    <body>
```

This means that `<variable>` iterates over values from 0 to `<limit>`−1. In Fygon `<variable>` is a lowercase letter from `a` to `z`, and `<limit>` is either already defined `<variable>` or a positive integer constant. The `<body>` of the loop is indented by four spaces and contains at least one statement.

The program receives the input in the variable `n`. This variable has special meaning and cannot be used as a loop variable.

Your task is to find the formula that calculates the number of performed `lag` operations by the given Fygon program, depending on the value of the variable `n`.

Input

The input file contains the Fygon program. No two loops use the same variable as iterators. Each variable used inside a `range` is either `n` or declared in some outer loop.

The program has at most 20 statements and at most 6 of them are loops. All integer constants are from 1 to 9.

Output

Output the formula for the number of performed `lag` operations depending on n. The length of the formula should be at most 100 characters (excluding spaces). The formula should correspond to the following grammar:

⟨Expression⟩ ::= ⟨Product⟩ ((‘+’ | ‘-’) ⟨Product⟩) *
⟨Product⟩ ::= ⟨Value⟩ (‘*’ ⟨Value⟩) *
⟨Value⟩ ::= ‘n’ | ⟨Number⟩ | ‘-’ ⟨Value⟩ | ‘(’ ⟨Expression⟩ ‘)’
⟨Number⟩ ::= [‘0’..‘9’] + (‘/’ [‘0’..‘9’] +) ?

Example

fygon.in	fygon.out

`for i in range(n):`
`  for j in range(i):`
`  lag`
`for x in range(5):`
`  for y in range(n):`
`  for z in range(n):`
`  lag`
`  lag`
	`1/2 * n * (n-1) + 5 * (n*n + 1)`