amppz

Evaluation of an Expression EWA

Available memory: 128 MB.

Consider an expression E, containing integer constants from 0 to 9, variables from a to z, and the following operations: addition, multiplication and exponentiation with a constant exponent. Quite surprisingly, each of the variables $a, b, \ldots, z$ appears in the expression E at most once. For a given prime number p, we would like to know how many roots modulo p the polynomial represented by this expression has. In other words, we want to count the number of ways in which integers from 0 to $p-1$ can be assigned to the variables in E, so that the value of E is divisible by p. Since the number of such roots can turn out large, it suffices to output it modulo 30 011.

For example, the polynomial represented by the following expression:

$$E = ((a+y)\cdot(z+8))^2$$

has 15 roots modulo $p = 3$, among which the roots:

$$(a=0,\ y=0,\ z=0), \qquad (a=1,\ y=2,\ z=0), \qquad (a=2,\ y=0,\ z=1)$$

can be found.

More formally, an *expression* is defined as follows:

- Each integer constant `0`, `1`, ..., `9` is an expression.
- Each variable `a`, `b`, ..., `z` is an expression.
- If `A` and `B` are any expressions, then each of `(A+B)` and `(A*B)` is also an expression: the first is the sum of expressions `A` and `B`, and the second is their product.
- If `A` is any expression, and `B` is an integer constant from 2, 3, ..., 9, then `(A^B)` is also an expression: the expression `A` raised to the power of `B`.

Input

The first line of input contains one prime number p ($2 \leqslant p < 15\,000$). The second line contains an expression E as specified above, described by a sequence of at most 300 characters `0`, `1`, ..., `9`, `a`, `b`, ..., `z`, `+`, `*`, `^`, `(`, `)`, without any white space.

Output

Let k denote the number of roots modulo p of the polynomial E. Your program should output one non-negative integer, k mod 30 011.

Example

For the input data:

```
3
(((a+y)*(z+8))^2)
```

the correct result is:

```
15
```