

D. Unambiguous Arithmetic Expression

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Let's define an *unambiguous arithmetic expression* (UAE) as follows.

- All non-negative integers are UAE's. Integers may have leading zeroes (for example, 0000 and 0010 are considered valid integers).
- If X and Y are two UAE's, then " $(X) + (Y)$ ", " $(X) - (Y)$ ", " $(X) * (Y)$ ", and " $(X) / (Y)$ " (all without the double quotes) are UAE's.
- If X is an UAE, then " $-(X)$ " and " $+(X)$ " (both without the double quotes) are UAE's.

You are given a string consisting only of digits ("0" - "9") and characters "-", "+", "*", and "/". Your task is to compute the number of different possible unambiguous arithmetic expressions such that if all brackets (characters "(" and ")") of that unambiguous arithmetic expression are removed, it becomes the input string. Since the answer may be very large, print it modulo 1000003 ($10^6 + 3$).

Input

The first line is a non-empty string consisting of digits ('0'-'9') and characters '-', '+', '*', and/or '/'. Its length will not exceed 2000. The line doesn't contain any spaces.

Output

Print a single integer representing the number of different unambiguous arithmetic expressions modulo 1000003 ($10^6 + 3$) such that if all its brackets are removed, it becomes equal to the input string (character-by-character).

Examples

input
1+2*3
output
2

input
03+ - 30+40
output
3

input
5 // 4
output
0

input
5/0
output
1

input
1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1

output
100728

Note

For the first example, the two possible unambiguous arithmetic expressions are:

$((1) + (2)) * (3)$

$(1) + ((2) * (3))$

For the second example, the three possible unambiguous arithmetic expressions are:

$(03) + ((- (30)) + (40))$

$(03) + (- ((30) + (40)))$

$((03) + (- (30))) + (40)$