# F. Long number

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Consider the following grammar:

- `<expression> ::= <term> | <expression> '+' <term>`
- `<term> ::= <number> | <number> '-' <number> | <number> '(' <expression> ')'`
- `<number> ::= <pos_digit> | <number> <digit>`
- `<digit> ::= '0' | <pos_digit>`
- `<pos_digit> ::= '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'`

This grammar describes a number in decimal system using the following rules:

- `<number>` describes itself,
- `<number>-<number>` (`l-r`, $l \le r$) describes integer which is concatenation of all integers from $l$ to $r$, written without leading zeros. For example, `8-11` describes `891011`,
- `<number>(<expression>)` describes integer which is concatenation of <number> copies of integer described by `<expression>`,
- `<expression>+<term>` describes integer which is concatenation of integers described by `<expression>` and `<term>`.

For example, `2(2-4+1)+2(2(17))` describes the integer `2341234117171717`.

You are given an expression in the given grammar. Print the integer described by it modulo $10^9 + 7$.

## Input

The only line contains a non-empty string at most $10^5$ characters long which is valid according to the given grammar. In particular, it means that in terms `l-r` $l \le r$ holds.

## Output

Print single integer — the number described by the expression modulo $10^9 + 7$.

### Examples

| input |
| --- |
| 8-11 |
| **output** |
| 891011 |

| input |
| --- |
| 2(2-4+1)+2(2(17)) |
| **output** |
| 100783079 |

| input |
| --- |
| 1234-5678 |
| **output** |
| 745428774 |

| input |
| --- |
| 1+2+3+4-5+6+7-9 |
| output |
| 123456789 |