The Value of Expression

Problem Description

Two operations are defined for 1-bit binary variables:

The Operator	Algorithm
⊕	0⊕0=0
	0⊕1=1
	1⊕0=1
	1⊕1=1
×	0×0=0
	0×1=0
	1×0=0
	1×1=1

The precedence of the operation is:

1. Evaluate what is inside the parentheses first, then evaluate what is outside parentheses.
2. "×" operation has higher precedence than "⊕" operation, that is, when evaluating the operation, evaluate × operation first, then evaluate ⊕ operation.

For example, when evaluating A⊕B×C, first evaluate B×C, then compute the result with A using the operator ⊕.

Now given an unfinished expression, such as _+(_*_), please fill in the number 0 or 1 in the horizontal line. How many ways are there to make this expression equal to 0?

[Explanation for Sample Input and Output]

The given expression includes the horizontal lines : _+(_*_)

When filling in (0, 0, 0), (0, 1, 0), (0, 0, 1), the value of the expression is 0. Therefore, there are 3 ways of filling in.

[Constraints]

For 20% of the data, $0 \le L \le 10$.

For 50% of the data, $0 \le L \le 1{,}000$.

For 70% of the data, $0 \le L \le 10{,}000$.

For 100% of the data, $0 \le L \le 100{,}000$.

For 50% of the input expression, there are no parentheses.

Input

The first line is an integer L, which indicates the number of operators and parentheses in the given expression excluding the horizontal line.

The second line is a string including L characters, which only contains 4 kinds of characters: '(', ')', '+', '*'. In which, '(' and ')' are left and right parentheses, and "+" and "*" represents the operators defined above: "⊕" and "×". This line of characters gives the operators and parentheses in a given expression in order, excluding variables.

Output

Containing a positive integer, that is, the total number of solutions. Note: This could be a very large number, please output the result of the number of schemes modulo 10007.

Sample Input

4
+(*)

Sample Output

3