

Brackets

Let's define a *correct string of brackets* as follows:

- () and [] are correct strings of brackets;
- if A is a correct string of brackets, then (A) and [A] are also correct strings of brackets;
- if A and B are both correct strings of brackets, then the concatenation AB is also a correct string of brackets;

In a correct string of brackets which contains at least one pair of square brackets: [and corresponding], each square bracket (both opening and closing) was replaced by the **opening** round bracket, therefore obtaining a *broken string of brackets*.

For example, ((and ((((((both are broken strings of brackets. First string is obtained from the correct strings of brackets []. Second string may be obtained only from the following four correct strings of brackets: []((((), ([]((((), (([](((() or ((([]((().

Your task is for a given broken string of brackets calculate the number of possible correct strings of brackets from which the given broken string may have been obtained.

Input data

The first line of text file **brackets.in** contains a single even integer N ($2 \leq N \leq 30000$) - the length of the given broken string of brackets. The second line contains N characters '(' and '[' - the given broken string of brackets.

Output data

The single line of the text file **brackets.out** should contain one integer - the required number of correct strings of brackets. Because the number of correct strings of brackets can be large, you should output the answer **modulo 100000009**.

Examples

Input data (file brackets.in)	Output data (file brackets.out)	Corresponding correct strings of brackets
4 ((()	2	[] (), ([])

Input data (file brackets.in)	Output data (file brackets.out)	Corresponding correct strings of brackets
8 ((((((((	14	[] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] [], [] [] []

Grading

Test cases where $N \leq 50$ are worth 20 points.

Test cases where $N \leq 1000$ are worth 45 points.