



Security Gate

Do you know Just Odd Inventions Co., Ltd.? The business of this company is doing “just odd inventions.” Here we just call it JOI Company.

There is a security gate installed in the doorway of JOI Company, in order to prevent confidential information leaks. One must pass through the gate when entering or exiting the company. It is impossible for two or more people to pass through the gate at the same time.

Whenever a person pass through this gate, it records the information indicating that a person enters or exits the company. Now, IOI-kun, an employee of JOI Company, has a record of the gate for some day. The record is represented by string S : if the i -th character of S is ‘(’, it means that the i -th person who passed through the gate entered the company, and if the i -th character of S is ‘)’, it means that the i -th person who passed through the gate exited the company. IOI-kun knows that there were no people inside JOI Company at the beginning or at the end of this day. Note that there exist strings consisting only of ‘(’ and ‘)’ which cannot represent a record: for example, a record cannot be $()()$ or $(()$ because it would follow that the number of people inside JOI Company had become negative or there had been a person inside JOI Company at the end of the day, respectively.

The next moment IOI-kun checked the record, the string S was modified by a computer virus spread across JOI Company! After some investigation, he supposed that the modification took the following process:

- First, a contiguous segment in S changed as follows: for each character in the segment, if it was ‘(’ then it became ‘)’, and if it was ‘)’ then it became ‘(’. We denote the string after this change by S' . The length of the changed segment might be 0, that is, $S = S'$.
- Then, 0 or more characters in S' became ‘x’. We denote the string after this change by S'' .

IOI-kun does not remember S , so he tries to recover S from S'' . For that, he wants first to count the number of strings which can be S' (not S , be careful).

Task

Given string S'' , write a program which calculates the number of strings which can be S' , modulo 1 000 000 007.

Input

Read the following data from the standard input.

- The first line of the input contains an integer N . This means that length of S'' , the string after the modification, is N .
- The second line of the input contains a string S'' , each character of which is ‘(’, ‘)’ or ‘x’. This means that the string after the modification is S'' .



Output

Write one line to the standard output. The output should contain the number of strings which can be S' , modulo 1 000 000 007. If there is no such string, output 0.

Constraints

All input data satisfy the following conditions.

- $1 \leq N \leq 300$.

Subtasks

There are 5 subtasks. The score and additional constraints of each subtask are as follows:

Subtask 1 [4 points]

- $N \leq 100$.
- The number of 'x's in S'' is at most 4.

Subtask 2 [8 points]

- $N \leq 100$.
- The number of 'x's in S'' is at most 12.

Subtask 3 [18 points]

- $N \leq 100$.
- The number of 'x's in S'' is at most 20.

Subtask 4 [43 points]

- $N \leq 100$.

Subtask 5 [27 points]

There are no additional constraints.



Sample Input and Output

| Sample Input 1 | Sample Output 1 |
|----------------|-----------------|
| 4 x))x | 3 |

In Sample Input 1, $S' =)))$ is not possible because there is no string which can be S .
The following three strings can be S' :

- $S' = ())$. For example, consider $S = ()$.
- $S' = ())$. For example, consider $S = ()$.
- $S' =)))$. For example, consider $S = (())$.

Since only these strings can be S' , output 3.

| Sample Input 2 | Sample Output 2 |
|-----------------|-----------------|
| 10 xx(xx)x(x | 45 |

| Sample Input 3 | Sample Output 3 |
|----------------|-----------------|
| 5 x))x(| 0 |

| Sample Input 4 | Sample Output 4 |
|------------------|-----------------|
| 10 xxxxxxxxxx | 684 |