Problem M. Make It Regular

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

You are given a bracket sequence consisting of n opening brackets and n closing brackets. Let S be a **non-empty** set of integers between 1 and $2n$, inclusive. You can choose two indices in S, not necessarily adjacent, and swap the brackets of the bracket sequence at those two positions.

Find the number of S that make it possible to obtain a regular bracket sequence by repeatedly applying this operation an arbitrary number of times. As this number may be very large, find it modulo the prime number 998 244 353.

Input

The first line contains one integer n ($1 \le n \le 3000$).

The second line contains a string of $2n$ brackets, either "`(`" or "`)`". The given bracket sequence contains n opening brackets and n closing brackets.

Output

Print the number of all possible S modulo 998 244 353.

Examples

standard input	*standard output*
3 `())(()`	36
6 `()))(())()((`	1536