Perfect Match

Time Limit: 4 seconds

Given a sequence **S** of length ***n***, consists of only **'('** and **')'**. Your task is to find number of different sub-sequences of ***S*** that are regular bracket sequences. For example, the sequence "((())())(" has 8 such subsequences: "((())())", "(())()", "((()))", "(()())", "(())", "()()", "()", and "".

Input

Input starts with an integer **T (< 101)**, denoting the number of test cases. Each test case contains an integer **n (0 < n < 1001)** and followed by a sequence **S** of **'('** and **')'**. **S** doesn't contain any white space.

Output

For each case, print the case number and the answer modulo **1000000007**.

Sample Input	Sample Output
1 9 ((())())(	Case 1: 8