NAIPC North American Invitational Programming Contest 2017

April 15, 2017

Hosted by: THE UNIVERSITY OF CHICAGO

Problem A
Pieces of Parentheses

You are teaching a class in programming, and you want to cover balanced parentheses. You've got a great visual aid, a sign with a very long, balanced string of parentheses. But, alas, somehow, your visual aid has been broken into pieces, and some pieces may be missing! You've got to try to put it back together as best you can. Given the string of parentheses on each piece, what is the longest balanced string you can form by concatenating some of them in some order? Each piece may be used at most once, and the pieces cannot be reversed.

A balanced string of parentheses is defined as:

1. The empty string
2. AB where A and B are both balanced strings of parentheses
3. (A) where A is a balanced string of parentheses

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. The first line of input will contain a single integer n ($1 \leq n \leq 300$), which is the number of pieces.

Each of the next n lines will hold a single string s ($1 \leq |s| \leq 300$), which consists only of the characters '(' and ')'. This describes one of the pieces.

Output

Output a single integer, which is the length of the longest string of balanced parentheses you can form from the pieces. Note that the empty string is technically a balanced string of parentheses, so it is always possible to form a string of length at least 0 (although the empty string is not a very effective visual aid!).

Sample Input 1	Sample Output 1
3 ( )) ((( )) ( )	10

Sample Input 2	Sample Output 2
5)))))) (( )) (((	2