D. Least Cost Bracket Sequence

time limit per test: 1 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

This is yet another problem on regular bracket sequences.

A bracket sequence is called regular, if by inserting "+" and "1" into it we get a correct mathematical expression. For example, sequences "(())()", "()" and "(()(()))" are regular, while ")(", "(()" and "(())))(" are not. You have a pattern of a bracket sequence that consists of characters "(", ")" and "?". You have to replace each character "?" with a bracket so, that you get a regular bracket sequence.

For each character "?" the cost of its replacement with "(" and ")" is given. Among all the possible variants your should choose the cheapest.

Input

The first line contains a non-empty pattern of even length, consisting of characters "(", ")" and "?". Its length doesn't exceed $5 \cdot 10^4$. Then there follow m lines, where m is the number of characters "?" in the pattern. Each line contains two integer numbers a_i and b_i ($1 \le a_i,\ b_i \le 10^6$), where a_i is the cost of replacing the i-th character "?" with an opening bracket, and b_i — with a closing one.

Output

Print the cost of the optimal regular bracket sequence in the first line, and the required sequence in the second.

Print -1, if there is no answer. If the answer is not unique, print any of them.

Examples

input
(??) 1 2 2 8
output
4 ()()