

C. Sereja and Brackets

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Sereja has a bracket sequence s_1, s_2, \dots, s_n , or, in other words, a string s of length n , consisting of characters "(" and ")".

Sereja needs to answer m queries, each of them is described by two integers l_i, r_i ($1 \leq l_i \leq r_i \leq n$). The answer to the i -th query is the length of the maximum correct bracket subsequence of sequence $s_{l_i}, s_{l_i+1}, \dots, s_{r_i}$. Help Sereja answer all queries.

You can find the definitions for a subsequence and a correct bracket sequence in the notes.

Input

The first line contains a sequence of characters s_1, s_2, \dots, s_n ($1 \leq n \leq 10^6$) without any spaces. Each character is either a "(" or a ")". The second line contains integer m ($1 \leq m \leq 10^5$) — the number of queries. Each of the next m lines contains a pair of integers. The i -th line contains integers l_i, r_i ($1 \leq l_i \leq r_i \leq n$) — the description of the i -th query.

Output

Print the answer to each question on a single line. Print the answers in the order they go in the input.

Examples

input
<pre>()()()()(7 1 1 2 3 1 2 1 12 8 12 5 11 2 10</pre>
output
<pre>0 0 2 10 4 6 6</pre>

Note

A *subsequence* of length $|x|$ of string $s = s_1s_2\dots s_{|s|}$ (where $|s|$ is the length of string s) is string $x = s_{k_1}s_{k_2}\dots s_{k_{|x|}}$ ($1 \leq k_1 < k_2 < \dots < k_{|x|} \leq |s|$).

A *correct bracket sequence* is a bracket sequence that can be transformed into a correct arithmetical expression by inserting characters "1" and "+" between the characters of the string. For example, bracket sequences "()", "(())", "((()))" are correct (the resulting expressions "(1) + (1)", "((1+1)+1)", and "((()))" and "()" are not).

For the third query required sequence will be « () ».

For the fourth query required sequence will be « () (()) (()) ».