

Problem F. Frustration and Bracket Sequences

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 1024 mebibytes

Ryan is interested in strings consisting only of ‘(’ and ‘)’. Especially, he loves balanced strings. Any balanced strings can be constructed using the following rules:

- A string “()” is balanced.
- The concatenation of two balanced strings is balanced.
- If T is a balanced string, the concatenation of ‘(’, T , and ‘)’ in this order is balanced.

For example, “()()” and “((()())” are balanced strings. “()”, “()()()” and “(” are not balanced strings. We define Ryan’s frustration for a string T as the minimum number of operations required to make T into a balanced string by doing the following operations in any order and any number of times.

- Add ‘)’ to the beginning of T .
- Add ‘(’ to the end of T .
- Swap two adjacent characters of T .

Ryan has a string S of length N consisting only of ‘(’ and ‘)’. Given Q queries, process them in order. There are two kinds of queries with the following formats.

- 1 l r : For each character from the l -th to the r -th (including r -th) of S , if it is ‘(’, replace it with ‘)’. If it is ‘)’, replace it with ‘(’.
- 2 l r : Output the value of Ryan’s frustration for the substring from the l -th through r -th characters of S .

Input

The first line contains two integers N and Q ($2 \leq N \leq 150\,000$, $1 \leq Q \leq 150\,000$) separated by a space, which represent the length of the string S and the number of queries. The following line contains the string S , which consists only of ‘(’ and ‘)’, and whose length is N . Each of the next Q lines contains three integers t_i , l_i and r_i ($1 \leq t_i \leq 2$, $1 \leq l_i \leq r_i \leq N$) separated by a space, which represent the i -th query. It is guaranteed that there is at least one query with $t_i = 2$.

Output

For each query with $t_i = 2$, print the value of Ryan’s sadness, followed by a newline.

Examples

standard input	standard output
6 6 () () (2 1 6 1 2 4 2 1 4 2 2 5 1 1 5 2 1 6	2 5 0 6
7 5 ((((((2 1 7 1 1 7 2 1 7 2 3 3 2 2 6	20 26 2 20