6838 Flipping Parentheses

A string consisting only of parentheses '(' and ')' is called balanced if it is one of the following.

- A string "()" is balanced.
- Concatenation of two balanced strings are balanced.
- When a string s is balanced, so is the concatenation of three strings "(", s, and ")" in this order.

Note that the condition is stronger than merely the numbers of '(' and ')' are equal. For instance, "())(()" is *not* balanced.

Your task is to keep a string in a balanced state, under a severe condition in which a cosmic ray may flip the direction of parentheses.

You are initially given a balanced string. Each time the direction of a single parenthesis is flipped, your program is notified the position of the changed character in the string. Then, calculate and output the *leftmost* position that, if the parenthesis there is flipped, the whole string gets back to the balanced state. After the string is balanced by changing the parenthesis indicated by your program, next cosmic ray flips another parenthesis, and the steps are repeated several times.

Input

The input will contain several test cases each of them formatted as follows.

N Q
s
q_1
⋮
q_Q

The first line consists of two integers N and Q ($2 \leq N \leq 300000$, $1 \leq Q \leq 150000$). The second line is a string s of balanced parentheses with length N. Each of the following Q lines is an integer q_i ($1 \leq q_i \leq N$) that indicates that the direction of the q_i-th parenthesis is flipped.

Output

For each test case, the output must follow the description below.

For each event q_i, output the position of the leftmost parenthesis you need to flip in order to get back to the balanced state.

Note that each input flipping event qi is applied to the string after the previous flip q_{i-1} and its fix.

Note: In the first sample, the initial state is "`((()))`". The 4th parenthesis is flipped and the string becomes "`((((`". Then, to keep the balance you should flip the 2nd parenthesis and get "`()(())`". The next flip of the 3rd parenthesis is applied to the last state and yields "`())())`". To rebalance it, you have to change the 2nd parenthesis again yielding "`(()())`".

Sample Input

```
6 3
((()))
4
```

```
3
1
20 9
()(((((()))))()()()()
15
20
13
5
3
10
3
17
18
```

Sample Output

```
2
2
1
2
20
8
5
3
2
2
3
18
```