

Shortest Regular Bracket

[English](#)

[Vietnamese](#)

Lets observe sequences made only of round and square brackets, i.e. characters '() []'.

A sequence of brackets is regular if it satisfies this inductive definition:

1. '()' and '[]' are regular sequences.
2. If A is regular, then (A) and [A] are regular sequences.
3. If A and B are regular, then AB is regular sequence.

For example '() () []', '([]) [()]' and '[(())] []' are regular, while '(', ']', '[(]' and '([)]' are not regular. The sequence of brackets is given.

In every step, one bracket is inserted at the beginning or at the end of the sequence (round or square, left or right).

Write a program that will, after each step, determine the length of the shortest regular subsequence of consecutive characters that contains the bracket added in that step.

Input

First line contains initial sequence of brackets, whose length is at most 100,000 characters.

Next line contains integer N, $1 \leq N \leq 100,000$, a number of steps.

In each of next N lines there are integer A and character C, separated by a single space. If A is zero (0), than character C is inserted at the beginning, and if A is one (1) then C is inserted at the end.

Output

In each of N lines, you should write the length of subsequence after that step. If there is no such subsequence, write number 0.

Sample

Input:

```
[])  
3  
0 )  
0 (  
0 (  

```

Output:

```
0  
2  
6  

```

Input:

(
3
1)
0)
0 (

Output:

0
0
2