



## Magic Parenthesis (Finland)

### TASK

In the LISP programming language *everything* is written inside balanced parentheses (LIKE THIS). This means that LISP code sometimes contains long stretches ")))...)" of closing parentheses. It is very tedious for the LISP programmer to get the number of these closing parentheses ')' to correspond exactly to the number of opening parentheses '('.

To prevent such syntax errors, some LISP dialects introduce a ***magic closing parenthesis*** ']' which substitutes *one or more closing parentheses ')' as needed to properly balance the opening parentheses '('*. But then the LISP compiler must calculate how many closing parentheses ')' each magic parenthesis ']' really substitutes. How?

Write a program which is given a string of opening, closing, and magic parentheses, and which calculates for each occurrence of the magic parenthesis the number of opening parentheses it corresponds to. In case of multiple solutions, the program should find any one of them.

### INPUT

The input file name is `LISP.IN`. The first line consists of two integers  $0 \leq N \leq 10\,000\,000$  and  $0 \leq M \leq 5\,000\,000$  separated by a space. The first number  $N$  is the length of the input string. The second number  $M$  is the number of magic closing parentheses in the string. The rest of the input file starts on the second line and is a string of length  $N$  consisting of characters '(', ')', and ']'. The character ']' occurs exactly  $M \leq N$  times in this string. This string is divided into lines of at most 72 characters each for readability.

### OUTPUT

The output file name is `LISP.OUT`. The first line consists of an integer '0' or '1'.

The number '0' means that the input cannot be balanced. (For example, the string which consists of just a single magic parenthesis "]" obviously cannot be balanced.) In this case there is no more output.

The number '1' means that the input can be balanced. In this the output continues with the following  $M$  extra lines.

The 1st of these extra lines consists of the number  $C_1 \geq 1$  of closing parentheses ')' the 1st magic parenthesis ']' in the input substitutes to. The 2nd extra line consists of the corresponding number  $C_2 \geq 1$  for the 2nd ']' in the input, and so on.



If there are many ways in which the given string can be balanced, then your program may output any one of them.

**EXAMPLE**

The input on the left describes a string with 8 characters, of which 2 are magic. The output on the right shows one way of balancing this input: the first magic parenthesis corresponds to 3 opening parentheses, and the second magic parenthesis corresponds to 1 opening parenthesis. And indeed, the magic-free string ((((( ))) ) ) is properly balanced, where the closing parentheses corresponding to the magic parentheses are underlined.

**INPUT**

8 2  
((( ( [ ] ) )

**OUTPUT**

1  
3  
1