Amazing Tree

Tree is an amazing data structure in computer science and many data can be represented as tree. For example, a black-white image can be represented as a quad tree and the arithmetic expression can be represented as a binary tree, a file system is also a tree, etc. Today, we introduce a method that can represent a general tree by using a string.

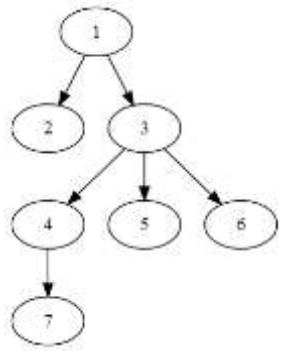

Figure 1:tree

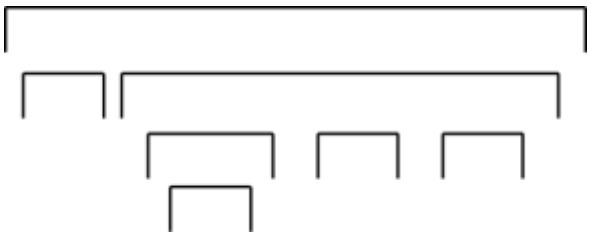
Figure 2: file system

((()(((()))()(()))

Figure 3: string

The tree in figure 1 can be represented as a file system in figure 2, and if we replace each file in figure 2 as a bracket pair () and maintain the relationship of the file folders, we would get the corresponding string in figure 3.

In summary, a tree can be represented as a string of bracket pairs and also a string of bracket pairs represents a tree.

Now, here comes the task.

You are given a tree, for each node i in the tree, the weight of node i is defined by the number of nodes in the sub-tree rooted at node i. The task is to calculate the XOR value of the weight of all nodes in the tree.

Input

The input contains multiple test cases. Each test case contains a string S in one line which only consists of brackets ( ). It is guaranteed that the string S forms a tree and the length of S is in the range from 2 to 1000.

Output

For each test case, print the result in a separate line.

Sample Input	Sample Output
(()((()()()))) (()(((())))	0 4

Hints: Recursion

Explanation:

In the first test case, it represents the tree in figure 1 and the weight of the nodes is as follows

Node	1	2	3	4	5	6	7
Weight	7	1	5	2	1	1	1

(Node 3 has weight 5 because there are 5 nodes in the subtree rooted at node 3)

Hence, the XOR value of the weight of all nodes is

$$7 \oplus 1 \oplus 5 \oplus 2 \oplus 1 \oplus 1 \oplus 1 = 0$$

C++ XOR operator: c = a ^ b;