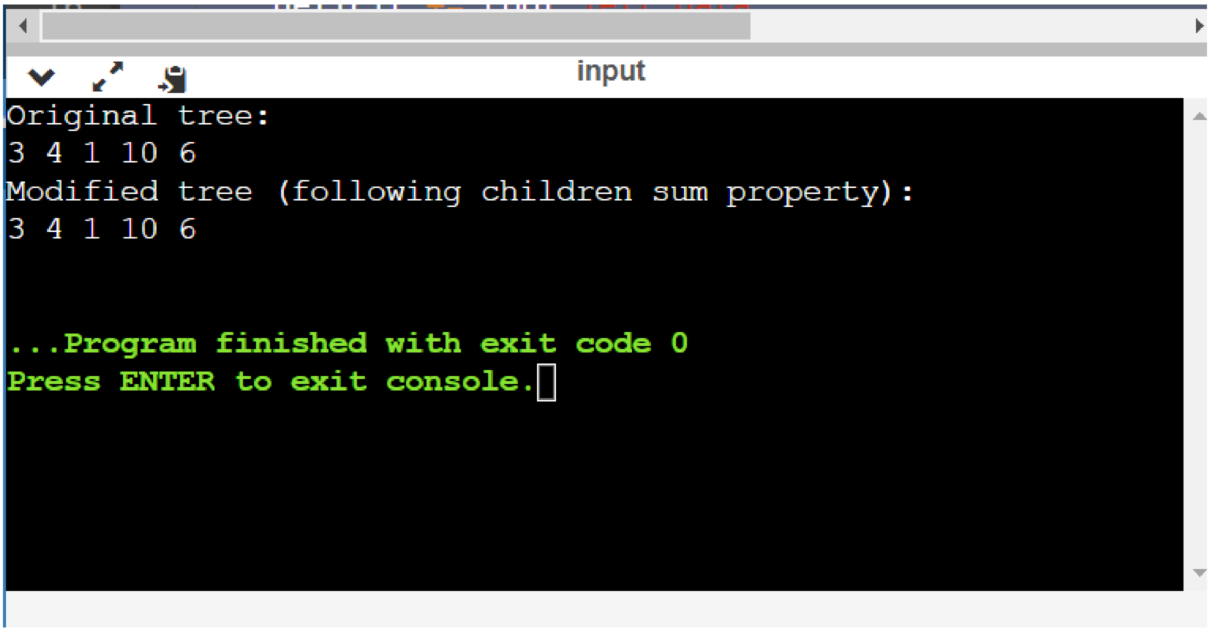
Problem 6: Write a program that converts any binary tree to one that follows the children sum property.

```
class Node:
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None


def children_sum_property(root):
    if root is None or (root.left is None and root.right is None):
        return


    children_sum_property(root.left)
    children_sum_property(root.right)


    deficit = 0
    if root.left:
        deficit += root.left.data
    if root.right:
        deficit += root.right.data - root.data


    if deficit > 0:
        if root.left:
            root.left.data += deficit
        else:
            root.left = Node(deficit)
    elif deficit < 0:
        root.data -= deficit

def inorder(root):
    if root:
        inorder(root.left)
        print(root.data, end=" ")
        inorder(root.right)

root = Node(10)
root.left = Node(4)
root.right = Node(6)
root.left.left = Node(3)
root.left.right = Node(1)

print("Original tree:")
inorder(root)
print()

children_sum_property(root)
```

```
print("Modified tree (following children sum property):")

inorder(root)

print()
```



```
Original tree:
3 4 1 10 6
Modified tree (following children sum property):
3 4 1 10 6


...Program finished with exit code 0
Press ENTER to exit console.
```