class ListNode:

    def \_\_init\_\_(self, val, left = None, right = None) -> None:

        self.val = val

        self.left = left

        self.right = right

def inverter(root):

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

        return root

    root.left, root.right = root.right, root.left

    if root.left is not None:

        root.left = inverter(root.left)

    if root.right is not None:

        root.right = inverter(root.right)

    return root

def printTree(root):

    if root is not None:

        print(root.val, end = " ")

        printTree(root.left)

        printTree(root.right)

root = ListNode(4, ListNode(2), ListNode(7))

root.left.left = ListNode(1)

root.left.right = ListNode(3)

root.right.left = ListNode(5)

root.right.right = ListNode(6)

print("Earlier")

printTree(root)

root = inverter(root)

print("\nAfter")

printTree(root)