```
class TreeNode:
    def __init__(self, val):
        self.val = val
        self.left = None
        self.right = None

# Function to mirror the tree
def mirror_tree(node):
    if node is None:
        return None

    # Swap left and right recursively
    node.left, node.right = mirror_tree(node.right), mirror_tree(node.left)
    return node

# In-order traversal to print tree
def inorder(node):
    if not node:
        return
    inorder(node.left)
    print(node.val, end=' ')
    inorder(node.right)

# Create a binary tree
#        1
#      /   \
#     2     3
#    / \   /
#   4   5 6

root = TreeNode(1)
root.left = TreeNode(2)
root.right = TreeNode(3)
root.left.left = TreeNode(4)
root.left.right = TreeNode(5)
root.right.left = TreeNode(6)

print("Inorder Before Mirroring:")
inorder(root)

mirror_tree(root)

print("\nInorder After Mirroring:")
inorder(root)
```