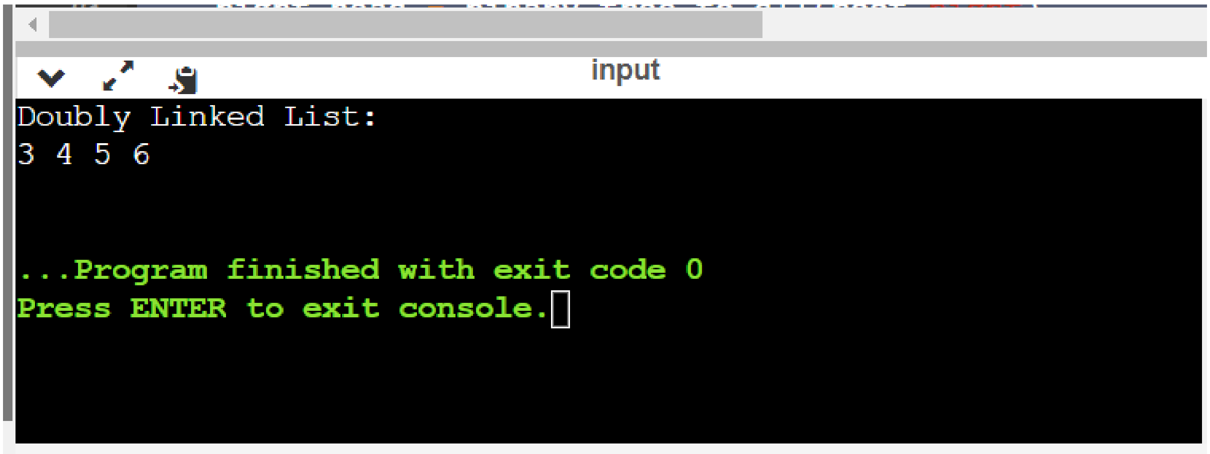
Problem 1: Binary Tree to Double Linked List

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

def binary_tree_to_dll(root):
    if root is None:
        return None

    left_head = binary_tree_to_dll(root.left)

    while left_head is not None and left_head.right is not None:
        left_head = left_head.right

    if left_head is not None:
        left_head.right = root
        root.left = left_head

    right_head = binary_tree_to_dll(root.right)

    if right_head is not None:
        right_head.left = root
        root.right = right_head

    return root if root.left is None else root.left

def print_dll(head):
    while head is not None:
        print(head.data, end=" ")
        head = head.right
    print()

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

dll_head = binary_tree_to_dll(root)

print("Doubly Linked List:")
print_dll(dll_head)
```



```
Doubly Linked List:
3 4 5 6


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