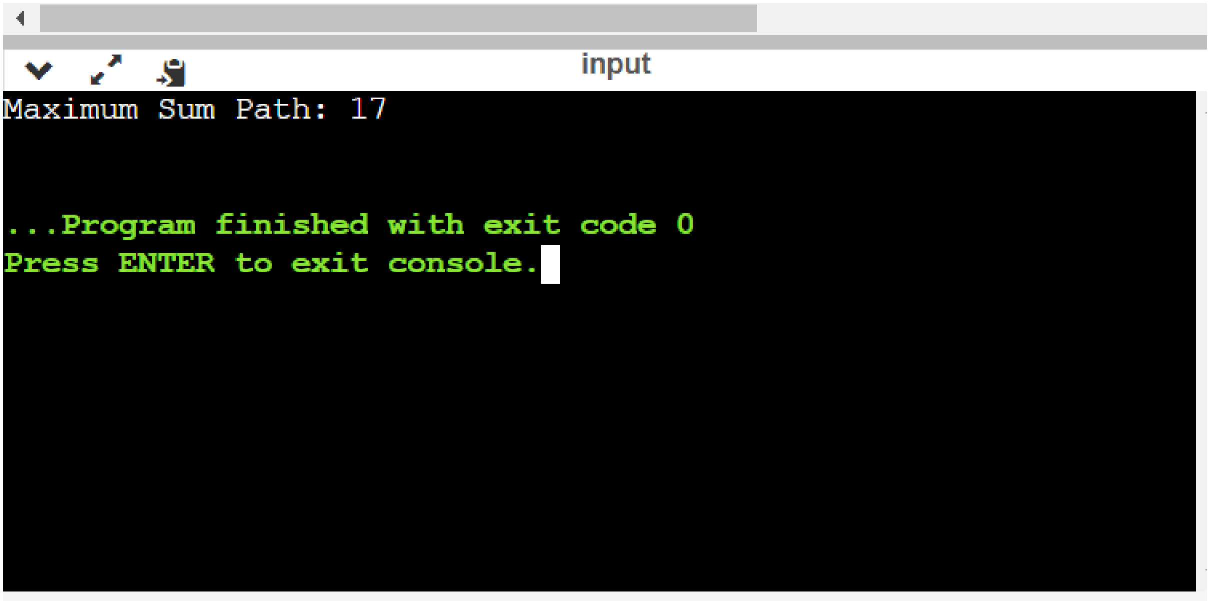
Problem 1: Write a program to find the maximum sum path in a binary tree. A path in a binary tree is a sequence of nodes where every adjacent pair of nodes are connected by an edge. A node can only appear in the sequence at most once. A path need not pass from the root. We need to find the path with the maximum sum in the binary tree.

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


def find_max_sum_path(root):
    if root is None:
        return 0

    max_sum = float('-inf')

    def dfs(node):
        nonlocal max_sum
        if node is None:
            return 0

        left_sum = max(dfs(node.left), 0)
        right_sum = max(dfs(node.right), 0)

        current_sum = node.value + left_sum + right_sum
        max_sum = max(max_sum, current_sum)

        return node.value + max(left_sum, right_sum)
    dfs(root)

    return max_sum
root1 = Node(1)
root1.left = Node(2)
root1.right = Node(3)
root1.left.left = Node(4)
root1.left.right = Node(5)
root1.right.right = Node(6)
print("Maximum Sum Path:", find_max_sum_path(root1))
```



```
Maximum Sum Path: 17


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