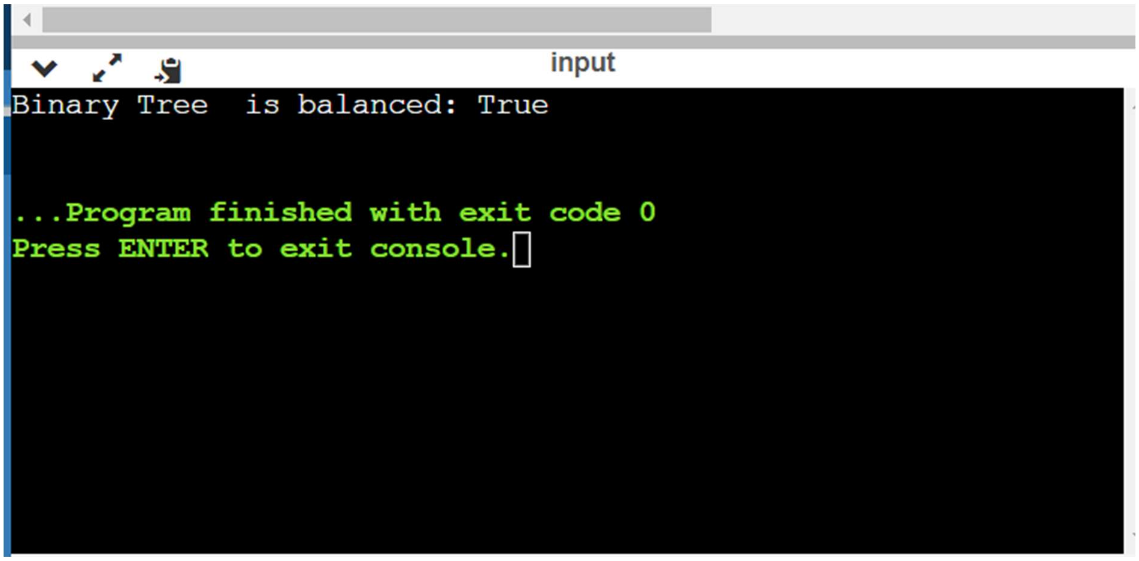
Problem 4: Check whether the given Binary Tree is a **Balanced Binary Tree** or not. A binary tree is balanced if, for all nodes in the tree, the difference between left and right subtree height is not more than 1.

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



def height(node):
    if node is None:
        return 0
    return max(height(node.left), height(node.right)) + 1



def is_balanced(root):
    if root is None:
        return True

    left_height = height(root.left)
    right_height = height(root.right)

    if (
        abs(left_height - right_height) <= 1
        and is_balanced(root.left)
        and is_balanced(root.right)
    ):
        return True

    return False
```

```
root = Node(1)

root.left = Node(2)

root.right = Node(3)

root.left.left = Node(4)

root.left.right = Node(5)

print("Binary Tree  is balanced:", is_balanced(root))
```



```
Binary Tree  is balanced: True


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