

**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):
```

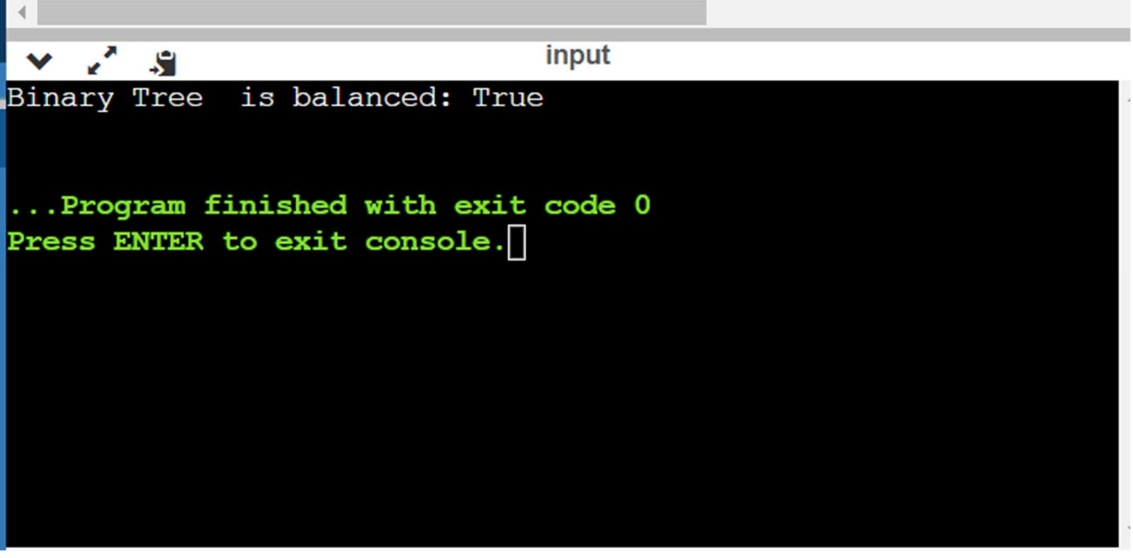
```
    if root is None:  
        return True
```

```
    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))
```

A screenshot of a terminal window with a title bar that includes a back arrow, a maximize icon, and a close icon, followed by the text "input". The terminal has a black background with white text. The first line of output is "Binary Tree is balanced: True". The second line is "...Program finished with exit code 0". The third line is "Press ENTER to exit console." followed by a white cursor box.

```
input
Binary Tree is balanced: True

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