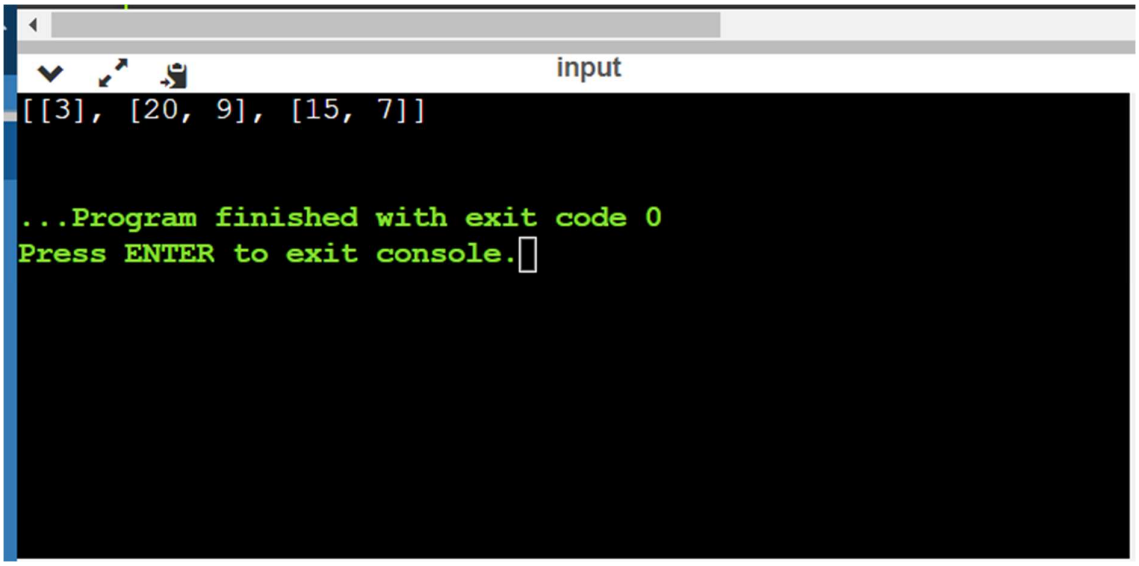
Problem Statement: Given the root of a binary tree, return the zigzag level order traversal of Binary Tree. (i.e., from left to right, then right to left for the next level and alternate between).

```
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right



def zigzagLevelOrder(root):
    if not root:
        return []

    result = []
    queue = [root]
    level = 0

    while queue:
        level_values = []
        level_size = len(queue)

        for _ in range(level_size):
```

```
            node = queue.pop(0)
            level_values.append(node.val)

            if node.left:
                queue.append(node.left)
            if node.right:
                queue.append(node.right)

        if level % 2 == 1:
            level_values.reverse()

        result.append(level_values)
        level += 1

    return result

root = TreeNode(3)
root.left = TreeNode(9)
root.right = TreeNode(20)
root.right.left = TreeNode(15)
root.right.right = TreeNode(7)
result = zigzagLevelOrder(root)
print(result)
```


input
[[3], [20, 9], [15, 7]]
...Program finished with exit code 0
Press ENTER to exit console.