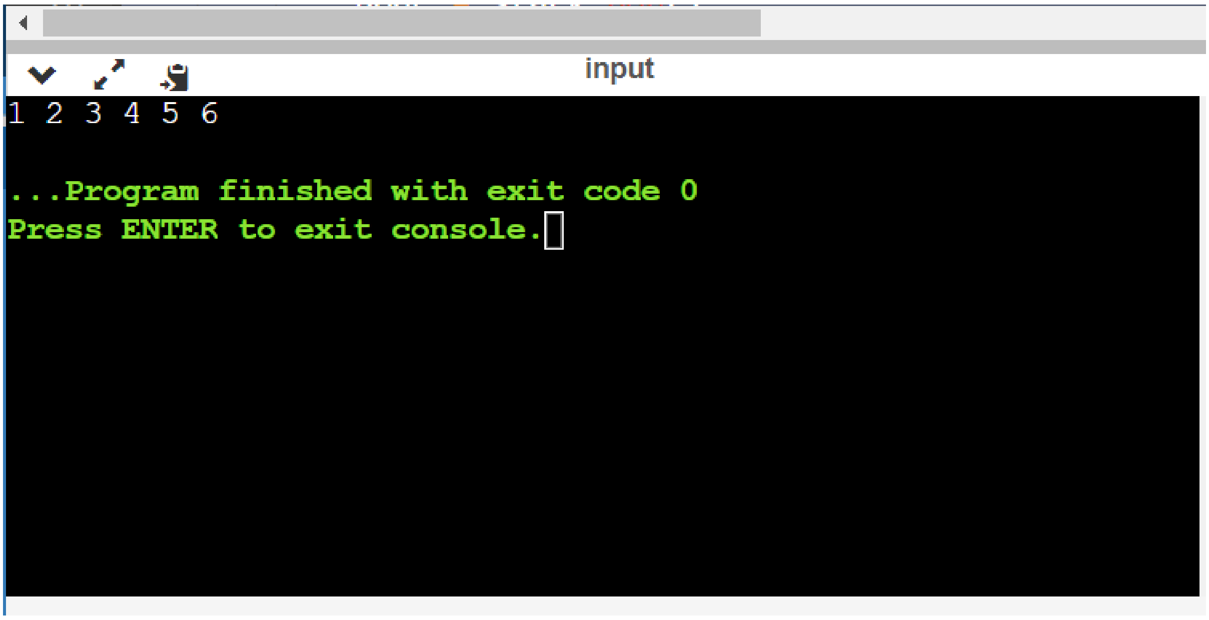
Problem 5: Flatten Binary Tree To Linked List. Write a program that flattens a given binary tree to a linked list.

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


class Solution:
    def flatten(self, root):
        if not root:
            return None


        stack = []
        stack.append(root)


        while stack:
            node = stack.pop()
            if node.right:
                stack.append(node.right)
            if node.left:
                stack.append(node.left)
            if stack:
                node.right = stack[-1]
            node.left = None


        return root


root = TreeNode(1)
root.left = TreeNode(2)
root.right = TreeNode(5)
root.left.left = TreeNode(3)

root.left.right = TreeNode(4)

root.right.right = TreeNode(6)

solution = Solution()

flattened = solution.flatten(root)

current = flattened

while current:

    print(current.val, end=" ")

    current = current.right
```



```
1 2 3 4 5 6

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