

Problem 2: Construct A Binary Tree from Inorder and Preorder Traversal.

```
class TreeNode:
```

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

```
        self.val = val
```

```
        self.left = left
```

```
        self.right = right
```

```
def buildTree(inorder, preorder):
```

```
    if not inorder or not preorder:
```

```
        return None
```

```
    root_val = preorder[0]
```

```
    root = TreeNode(root_val)
```

```
    root_index = inorder.index(root_val)
```

```
    root.left = buildTree(inorder[:root_index], preorder[1:root_index + 1])
```

```
    root.right = buildTree(inorder[root_index + 1:], preorder[root_index + 1:])
```

```
    return root
```

```
def inorderTraversal(root):
```

```
    if root:
```

```
        inorderTraversal(root.left)
```

```
        print(root.val, end=" ")
```

```
        inorderTraversal(root.right)
```

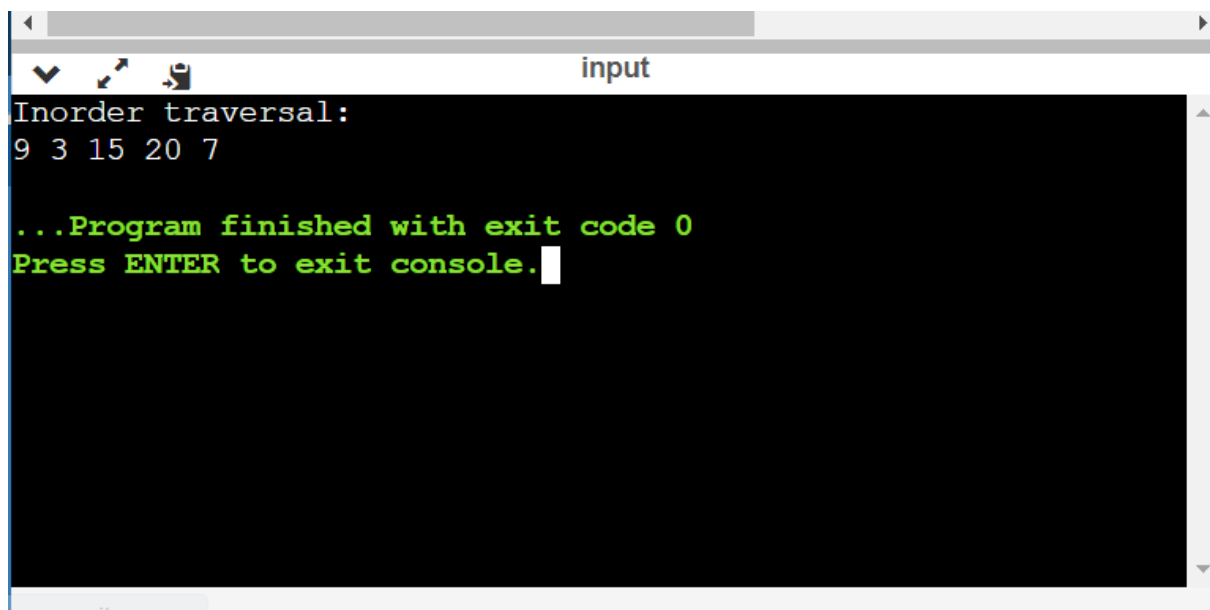
```
inorder = [9, 3, 15, 20, 7]
```

```
preorder = [3, 9, 20, 15, 7]
```

```
root = buildTree(inorder, preorder)
```

```
print("Inorder traversal:")
```

inorderTraversal(root)



```
Inorder traversal:
9 3 15 20 7

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