Exercise 7

```
class Node:
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None
class BinaryTree:
    def __init__(self):
        self.root = None
    def inorder(self, node):
        if node:
            self.inorder(node.left)
            print(node.data, end=" ")
            self.inorder(node.right)
    def preorder(self, node):
        if node:
            print(node.data, end=" ")
            self.preorder(node.left)
            self.preorder(node.right)
    def postorder(self, node):
        if node:
            self.postorder(node.left)
            self.postorder(node.right)
            print(node.data, end=" ")
bt = BinaryTree()
bt.root = Node("A")
bt.root.left = Node("B")
bt.root.right = Node("C")
bt.root.left.left = Node("D")
bt.root.right.left = Node("E")
bt.root.right.right = Node("F")
print("Inorder Traversal: ")
bt.inorder(bt.root)
print("\nPreorder Traversal: ")
bt.preorder(bt.root)
print("\nPostorder Traversal: ")
bt.postorder(bt.root)
```

Output

```
Python 3.10.9 (tags/v3.10.9
Type "help", "copyright", "crec
>>
====================
Inorder Traversal:
D B A E C F
Preorder Traversal:
A B D C E F
Postorder Traversal:
D B E F C A
>>
```