

# Binary Search Tree

1. Write a Python program to implement a binary search tree.

```
class Tree:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

def preOrderTraversal(Node):
    if Node:
        print(Node.value, end=" ")
        preOrderTraversal(Node.left)
        preOrderTraversal(Node.right)

def inOrderTraversal(Node):
    if Node:
        inOrderTraversal(Node.left)
        print(Node.value, end=" ")
        inOrderTraversal(Node.right)

def postOrderTraversal(Node):
    if Node:
        postOrderTraversal(Node.left)
        postOrderTraversal(Node.right)
        print(Node.value, end=" ")

def Tree1():
    rootNode = Tree(1)
    rootNode.left = Tree(2)
    rootNode.right = Tree(3)
    rootNode.left.left = Tree(4)
    rootNode.left.right = Tree(5)
    return rootNode

root = Tree1()

print("Preorder Traversal: ", end="")
preOrderTraversal(root)
print()

print("Inorder Traversal: ", end="")
inOrderTraversal(root)
print()

print("Postorder Traversal: ", end="")
postOrderTraversal(root)
print()
```

Output:

```
Preorder Traversal: 1 2 4 5 3
Inorder Traversal: 4 2 5 1 3
Postorder Traversal: 4 5 2 3 1
```