LAB-08

```
#--------------------------Task:1-7---------------------=--


class Node:
    def __init__(self,data,parent = None):
        self.data = data
        self.parent = parent
        self.left = None
        self.right = None

root = Node(10,None)
n2 = Node(5,root)
n3 = Node(7,root)
root.left = n2
root.right = n3
n4 = Node(17,n2)
n5 = Node(9,n2)
n6 = Node(25,n3)
root.left.left = n4
root.left.right = n5
root.right.right = n6

root1 = Node(10,None)
nn2 = Node(5,root)
nn3 = Node(7,root)
root1.left = nn2
root1.right = nn3
nn4 = Node(17,nn2)
nn5 = Node(9,nn2)
nn6 = Node(25,nn3)
root1.left.left = nn4
root1.left.right = nn5
root1.right.left = nn6

def depth(node):
    if node == root:
        return 0
    return 1 + depth(node.parent)

def level_Node(node):
    if node == root:
        return 1
    else:
        return 1 + depth(node)

def height_Tree(node):
    if node is None:
        return -1
    else:
        if height_Tree(node.left) > height_Tree(node.right):
            return 1 + height_Tree(node.left)
        else:
            return 1 + height_Tree(node.right)

def preOrder(root):
    print(root.data,end = ' ')
    if root.left:
        preOrder(root.left)
    if root.right:
        preOrder(root.right)

def inOrder(root):
    if root.left:
        inOrder(root.left)
    print(root.data,end = ' ')
    if root.right:
        inOrder(root.right)

def postOrder(root):
    if root.left:
        postOrder(root.left)
    if root.right:
        postOrder(root.right)
    print(root.data,end = ' ')

def is_Equal(node1,node2):
    if node1 is None and node2 is None:
        return True
    elif node1 and node2:
        if node1.data != node2.data:
            return False
        else:
            return is_Equal(node1.left, node2.left) and is_Equal(node1.right,node2.right)
    else:
        return False

def copy_Tree(root):
    temp = Node(root.data)
    if root.right:
        temp.right = copy_Tree(root.right)
    if root.left:
        temp.left = copy_Tree(root.left)
    return temp

print('Level of the Node : '+ str(level_Node(n6)))
print('Height of the Tree : '+ str(height_Tree(root)))
preOrder(root)
inOrder(root)
```

```
postOrder(root)
if is_Equal(root,root1):
    print("Trees are Equal")
else:
    print("Trees are not Equal")
copy_tree = copy_Tree(root)
preOrder(copy_tree)
inOrder(copy_tree)
postOrder(copy_tree)
```

Task-08(a)