CSE 220 - ASSIGNMENT 8

SADAF M. ANIS ID : 20101537 SEC : 08

In [19]:
```python
#sadaf m.anis

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

class binary_tree:
    def __init__(self,item):

        self.item = Node(item)


    #TASK 1
    def height(self,root):
        if root == None:
            return -1

        l_height = self.height (root.left)
        r_height = self.height(root.right)

        if l_height > r_height:
            return l_height + 1
        else:
            return r_height + 1



    #TASK 2

    def level(self,root):
        if root.parent == None:
            return 1

        return 1 + self.level(root.parent)

    #TASK 3

    def pre_order(self,new):

        if new == None:
            return

        else:
            print(new.value,end =" ")
            self.pre_order(new.left)
            self.pre_order(new.right)


    #TASK 4

    def in_order(self,new):
        if new == None:
            return

        else:
            self.in_order(new.left)
            print(new.value, end = " ")
            self.in_order(new.right)

    #TASK 5

    def pos_order(self,new):

        if new == None:
            return

        else:
            self.pos_order(new.left)
            self.pos_order(new.right)
            print(new.value, end = " ")



n1 = binary_tree(7)
n2 = binary_tree(10)
n3 = binary_tree(3)
n4 = binary_tree(5)
n5 = binary_tree(6)
n6 = binary_tree(30)
n7 = binary_tree(75)

root1 = n1


n1.item.left = n2.item

n2.item.parent = n1.item

n1.item.right = n3.item

n3.item.parent = n1.item

n2.item.left = n4.item

n4.item.parent = n2.item

n2.item.right = n5.item

n5.item.parent = n2.item

n3.item.right = n6.item

n6.item.parent = n3.item

n4.item.left = n7.item

n7.item.parent = n4.item



```

```python
t1 = binary_tree(7)
t2 = binary_tree(10)
t3 = binary_tree(3)
t4 = binary_tree(5)
t5 = binary_tree(6)
t6 = binary_tree(30)
t7 = binary_tree(75)

root2 = t1

t1.item.left = t2.item

t2.item.parent = t1.item

t1.item.right = t3.item

t2.item.left = t4.item

t4.item.parent = t2.item

t2.item.right = t5.item
t5.item.parent = t2.item
t3.item.right = t6.item
t6.item.parent = t3.item
t4.item.left = t7.item
t7.item.parent = t4.item


print("TASK 1: ")
print("Height :", root1.height(root1.item))

print("TASK 2: ")
print("Level:", root1.level(root1.item.left.left.left))

print("TASK 3: ")
root1.pre_order(root1.item)

print("")
print("TASK 4: ")

root1.in_order(root1.item)
print("")
print("TASK 5: ")
root1.pos_order(root1.item)

```

```
TASK 1:
Height : 3
TASK 2:
Level: 4
TASK 3:
7 10 5 75 6 3 30
TASK 4:
75 5 10 6 7 3 30
TASK 5:
75 5 6 10 30 3 7
```

In [ ]:

```

```