CSC 120 **ICA-24**

Work with your neighbor. (This will be graded for participation only.)

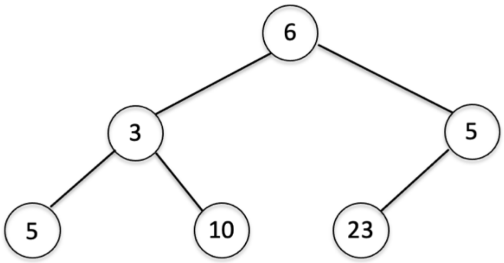
1. There are three common ways to traverse a tree. For a binary tree, list the order of visiting the nodes and the children for each of these traversals.

   pre-order: ______________________________

   in-order: ______________________________

   post-order: ______________________________

2. Give the above traversals for the following tree:



   pre-order: ______________________________

   in-order: ______________________________

   post-order: ______________________________

For the remaining problems assume that a `BinaryTree` class has been defined with attributes `_value`, `_left`, and `_right`. The following getters have also been defined in the class:

```
def value(self):
    return self._value

def left(self):
    return self._left

def right(self):
    return self._right
```

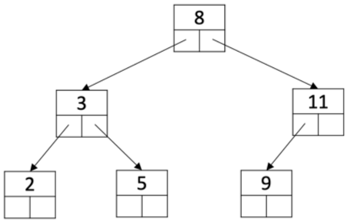
Also, for reference, here is the code for an inorder traversal of a binary tree:

```
def inorder(tree):
    if tree == None:
        return
    else:
        inorder(tree.left())
        print(tree.value())
        inorder(tree.right())
```

3. Write a function `postorder(tree)` that prints the nodes of a tree in postorder.

4. Write a function `sum_leaves(tree)` that returns the *sum* of the values of the leaf nodes in the binary tree `tree`.

5. Write a function `inorder_str(tree)` that produces a *string* of the inorder traversal of the binary tree argument `tree`. For the tree below,



The string returned would be

`"2,3,5,8,9,11"`