CSC/DSCI 2720 Data Structure
Lab 10

Due: 03/24/2023 @ 11:59PM

Requirements:

(Failure to follow the requirements will result in a score of Zero.)

1. You may use whatever IDEs / editors you like, but you must submit your responses on iCollege as **.py files**.
2. Your submissions will work exactly as required.
3. Make sure you submit a file that compiles.
4. Your submission will show an output. Should you recive a Zero for no output shown do not bother to email with "but the logic is perfect".
5. Your program's output must exactly match the specs (design, style) given here for each problem to pass the test cases.
6. Design refers to how well your code is written (i.e., is it clear, efficient, and elegant), while Style refers to the readability of your code (correct indentation, good variable names). Add comments to have necessary explanations for your program.
7. Add a "heading" at the very beginning of your .java files as follow:
   Your Name
   CSc 2720 Lab #N
   Lab time: put your lab time here
   Due time: put the due date here
8. *** If you used any website/online resources as reference, please cite in your comments what website did you use for studying the lab content. Also, you are supposed to make changes to the resource you use to make it your own version. Otherwise, your submission will be considered as plagiarism. ***

PROBLEM STATEMENT :
In today's Lab we will explore ways to perform a Pre-order, In-order and Post-order Traversal on a Binary Tree.

You will solve the problem as stated below:
The root element of the binary tree is given to you. Your task is to return the pre-order/in-order/post-order traversal of the Binary Tree's nodes' values.

You will need to use the given class to represent tree node:

```
# Definition for a binary tree node.

class TreeNode:
    def __init__(self, val):
        self.val = val
        self.left = None
        self.right = None
```

Below is an illustrated sample of Binary Tree nodes for your reference:

```
root = TreeNode(4)
root.left = TreeNode(2)
root.right = TreeNode(6)
root.left.left = TreeNode(1)
root.left.right = TreeNode(3)
root.right.left = TreeNode(5)
root.right.right = TreeNode(7)
```

Your code will need to return all the nodes of the Binary Tree:

Pre-order Traversal: 4 2 1 3 6 5 7

In-order Traversal: 1 2 3 4 5 6 7

Post-order Traversal: 1 3 2 5 7 6 4

Very Very Important:

1. (1) Your code should be well commented which explains all the steps you are performing to solve the problem. A submission without code comments will immediately be deducted 15 points!
2. (2) As a comment in your code, please write your test-cases on how you would test your solution assumptions and hence your code.
   A submission without test cases (as comments) will immediately be de- ducted 15 points!
   Please Remember: Although, written as comments - You will address your test cases in the form of code and not prose :)