CSC 2720: Data Structures
Lab 10

Instructor: Shiraj Pokharel

Due : @ 11:00 PM ET , Next Day after release

Answer the below questions.
You may use whatever IDEs / editors you like, but you must submit your responses on iCollege as .java files.
Failure to comply with this simple requirement will result in a score of Zero. Please, be careful not to be assigned a Zero score this way.

Few Rules to be followed, else will receive a score of ZERO

(1) Your submissions will work exactly as required.

(2) Your files shall not be incomplete or worse corrupted such that the file does not compile at all. Make sure you submit a file that compiles.

(3) Your submission will show an output. Should you recive a Zero for no output shown do not bother to email me with ”but the logic is perfect” !

Note that your program's output must **exactly** match the specs(design , style) given here for each problem to pass the instructor's test cases .
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 (commented, correct indentation, good variable names).

PROBLEM STATEMENT :
In today's Lab we will explore ways to perform a Level Order Traversal on a **Binary Tree**.
You will implement this traversal using the ArrayDeque Class in Java.
Java Docs URL reference here:
https://docs.oracle.com/javase/8/docs/api/java/util/ArrayDeque.html

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 level order traversal of the Binary Tree's nodes' values – from left to right, level by level.

```java
/* Class to represent Tree node */
class Node {
    int data;
    Node left, right;

    public Node(int item)
    {
        data = item;
        left = null;
        right = null;
    }
}
```

Below is an illustrated sample of Binary Tree nodes for your reference, which in-fact is the same example we discussed in the lecture.

```java
tree.root = new Node(4);
tree.root.left = new Node(2);
tree.root.right = new Node(6);
tree.root.left.left = new Node(1);
tree.root.left.right = new Node(3);
tree.root.right.left = new Node(5);
tree.root.right.right = new Node(7);
```

Your code will need to return all the nodes of the **Binary Tree** in Level Order Traversal fashion - from left to right. That is: **4 2 6 1 3 5 7**
Level Order Tree Traversal with the help of the Queue data-structure should meet the following complexities as stated below.

Time Complexity: $O(n)$
Space Complexity: $O(n)$

Submissions that don't meet the mentioned Time and Space complexities will only receive 50% credit.

Very Very Important :

(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) 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 deducted 15 points ! Please Remember : Although, written as comments - You will address your test cases in the form of code and not prose :)