Homework(16/06/2025)

Q1-- write a java program to define TreeNode, construct a tree , perform level order traversal on it and result will be like

1

2 3

4 5

Java program-

```
import java.util.LinkedList;
import java.util.Queue;
class TreeNode {
    int data;
    TreeNode left, right;

    TreeNode(int value) {
        data = value;
        left = right = null;
    }
}

public class BinaryTreeLevelOrderLineByLine {
    public static void levelOrderLineByLine(TreeNode root) {
        if (root == null) return;

        Queue<TreeNode> queue = new LinkedList<>();
        queue.add(root);
```

```
    System.out.println("Level Order Traversal (Line by Line):");
    while (!queue.isEmpty()) {
        int levelSize = queue.size();

        for (int i = 0; i < levelSize; i++) {
            TreeNode current = queue.poll();
            System.out.print(current.data + " ");

            if (current.left != null)
                queue.add(current.left);
            if (current.right != null)
                queue.add(current.right);
        }
        System.out.println();
    }
}

public static void main(String[] args) {

    TreeNode root = new TreeNode(1);
    root.left = new TreeNode(2);
    root.right = new TreeNode(3);
    root.left.left = new TreeNode(4);
    root.left.right = new TreeNode(5);
```

```
        levelOrderLineByLine(root);
    }
}
```

Output—

```
1
2 3
4 5
```

Q2—Symmetric tree

Java program

```
class Solution {
    public boolean isSymmetric(TreeNode root) {
        if (root == null) {
            return true;
        }
        return isMirror(root.left, root.right);
    }

    private boolean isMirror(TreeNode node1, TreeNode node2) {
        if (node1 == null && node2 == null) {
            return true;
        }
        if (node1 == null || node2 == null) {
            return false;
```

```
        }
        return node1.val == node2.val && isMirror(node1.left, node2.right) &&
isMirror(node1.right, node2.left);
    }
}
```

Q3—Mirror tree

Java program

```
class Solution {

    void mirror(Node node) {
         if(node==null){
            return;
        }

        if(node.left==null && node.right!=null){
            Node temp1 = node.left;
            node.left = node.right;
            node.right = temp1;

        }

        else if(node.left!=null && node.right==null){
            Node temp2 = node.right;
            node.right = node.left;
            node.left = temp2;
```

```
        }
        else{
            Node temp = node.left;
            node.left = node.right;
            node.right = temp;
        }

        mirror(node.left);
        mirror(node.right);
    }
}
```