**Data Structures and Algorithms**

**LAB-09**

**Issue Date: April 28, 2025**

**Start Time: 1:45 PM**

**Completion Time: 3:20 PM**

**Total Marks: 40**

**Objective:**

In this lab, students will practice about the working of Binary Trees.

**Instructions:**

1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) Your laptops must be on airplane mode.
3) Anyone caught in an act of plagiarism would be awarded an "F" grade in this Lab.

## TASK-01: Construct Binary Tree from String [20 Marks]

You are given a string that represents a binary tree.
The string contains **integers** and **parentheses** ( ), where:

- An integer represents a node's value.
- A pair of parentheses encloses the left and right **subtrees** of the node.

You need to construct the binary tree based on this string format and implement a function that returns the root of the tree.

In the given string,

1- The first number always represents the root node of the tree.
2- If there are parentheses immediately after a number, they represent the left child of that node.
3- If there is a second set of parentheses, it represents the right child of that node.
4- Negative numbers are allowed.

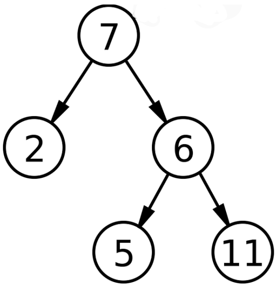
Function prototype:

**TreeNode * str2tree(string s);**

**Sample Run**

**Input**: 7(2)(6(5)(11))

**Output:**



**Explanation:**

- The first number 7 is the root of the tree.
- The first set of parentheses (2) immediately after 7 represents the left child of 7.
- The second set of parentheses (6(5)(11)) immediately after 7 represents the right child of 7.

  - Inside this, (5) represents the left child of 6.
  - Similarly, (11) represents the right child of 6.

## TASK-02: Symmetric Tree [10 Marks]

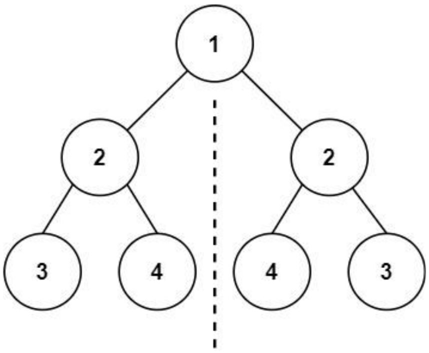
Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center).

Function prototype:

**bool isSymmetric(TreeNode*root)**

**Sample run**

**Input:** root = [1,2,2,3,4,4,3]



**Output:** true

## TASK-03: Binary Tree Tilt [10 Marks]

Given the root of a binary tree, return the sum of every tree node's tilt.

The **tilt** of a tree node is the **absolute difference** between the sum of **all left subtree** node values and all right subtree node values. If a node does not have a left child, then the sum of the left subtree node values is treated as 0. The rule is similar if the node does not have a right child.

Function prototype :

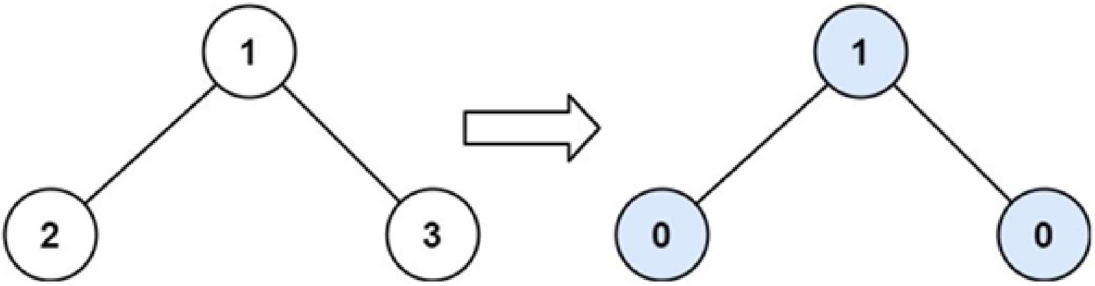
**int findTilt(TreeNode* root)**

**Sample Run:**

**Input:** root = [1,2,3]

**Output:** 1

**Explanation:**



- Tilt of node 2 : |0-0| = 0 (no children)
- Tilt of node 3 : |0-0| = 0 (no children)
- Tilt of node 1 : |2-3| = 1 (left subtree is just left child, so sum is 2; right subtree is just right child, so sum is 3)

Sum of every tilt : 0 + 0 + 1