CL2001 – Data Structure

LabTask # 11

Note:

- Copied task will be awarded **zero** marks.
- Use comments wherever applicable.
- Note that these lab task marks could be graded through a viva in lab.
- Variables and functions names should be meaningful.

Problem: 1 | BST is AVL or not

Write a program in C++ to check if a given binary search tree (BST) is AVL or not. An AVL tree is a self-balancing binary search tree where the heights of the two child subtrees of any node differ by at most one.

Your program should include the following:

1. A function to construct a binary search tree from the given input.
2. A function to check if a tree is AVL or not.
3. The main function to take input, construct the BST, and call the function to check if the tree is AVL.

Input Format:

The input consists of integers representing the nodes of the binary search tree.

Output Format:

Print "AVL" if the given BST is AVL, otherwise print "Not AVL".

Problem: 2 | Insertion in AVL

Provide a C++ implementation of AVL tree must include.

- Recursive RR
- Recursive LL
- Recursive RL
- Recursive LR
- Apply on BST Insertion
- Finding Balancing Factor
- Display Nodes
- Test Your Code