Data Structures Lab (CSC204) **Session: Monsoon 2023-24**
III BTech (CSE) **Date: 09.10.2023**

Assignment-7 (Tree-II)

---

1. A binary search tree is a binary tree which follows some order to arrange the elements. In a binary search tree, the value of left node must be smaller than the parent node, and the value of right node must be greater than the parent node. This rule is applied recursively to the left and right sub trees of the root.

   a. Write a function $\boldsymbol{create\_BST}(\,)$ to create a binary search tree by inserting the user given values. Your program should print all the *leaf nodes* from left to right after the complete build-up of the tree.

      Sample Input:
      Enter the number of nodes to be inserted: 9
      Enter the value of the nodes: $40, 55, 45, 30, 35, 65, 83, 25, 33$

      Sample Output:
      Leaf nodes from left to right is: $25, 33, 45, 83$

   b. Write a function $\boldsymbol{delete\_BST}(\,)$ which will delete a specific node from the binary search tree by maintaining its inherent properties. Your program should print all the *leaf nodes* from left to right after performing the deletion operation.

      Sample Input:
      Enter the value of the node which needs to be deleted: 33

      Sample Output:
      Leaf nodes from left to right is: $25, 35, 45, 83$

2. An AVL tree is a type of self-balancing binary search tree. In an AVL tree, the heights of the left and right sub trees of any node could differ by at most one. If, at any point, they differ by more than one, the tree is rebalanced to maintain this property. Insertions and deletions may necessitate one or more tree rotations to rebalance the tree. Write a program which can perform following operations over AVL trees.

   a. Write a function $\boldsymbol{create\_AVL}(\,)$ to create an AVL tree by inserting the user given values. Your program should print the *pre-order traversal* of the tree after the complete build-up of the tree.

      Sample Input:
      Enter the number of nodes to be inserted: 9
      Enter the value of the nodes: $40, 55, 45, 30, 35, 65, 83, 25, 33$

      Sample Output:
      Pre-order traversal of the created AVL tree is: $45, 35, 30, 25, 33, 65, 55, 83$