

**SSN COLLEGE OF ENGINEERING`
DEPARTMENT OF CSE**

DATA STRUCTURES LABORATORY (CS8381)

Ex. No. 8 Implementation of Binary Search Tree and AVL Tree

i) Create a **binary search tree** (BST) to hold integer data and perform the following operations on it:

- (a) Insertion of an integer into BST (no duplicates)
- (b) Deletion of a given integer
- (c) Find a given integer in BST
- (d) Find minimum element in BST
- (d) Display the integers in ascending order

ii) Create a balanced binary search tree i.e **AVL tree** for the above mentioned problem and perform the above operations except deletion. The program should work for all cases of insertion (i.e. the program should perform left rotation, right rotation, left-right rotation, right-left rotation depending upon the type of insertion).

Both BST and AVL tree should be compared for the following cases:

- 1) Insert 14, 25, 12, 20, 37, 9, 13
- 2) Insert 25, 26, 27, 28, 29, 30, 31

Also calculate the number of times, the find function is being called recursively to find number 31 in BST and AVL tree for the second case, using a count variable.