Applications of Data Structures & Algorithms

Q1

- Design a pseudocode function to find the sum of all leaf nodes in a binary tree. Assume the tree nodes contain integer values. Demonstrate your understanding of tree traversal by using a recursive approach in your solution.

Q2

- Write a pseudocode function to determine whether a given binary tree is a **binary search tree (BST)**. A BST is defined as a binary tree where the value of every node is greater than all values in its left subtree and smaller than all values in its right subtree. Use a recursive approach in your solution.

Q3

- Write a pseudocode function to calculate the height of a binary tree. The height of a tree is defined as the number of edges on the longest path from the root to a leaf node. Use a recursive approach to solve the problem.

Q4

- Write a pseudocode function to check if two binary trees are **identical**. Two trees are identical if they have the same structure and all corresponding nodes have the same values. Use a recursive approach to solve the problem.

Q5

- Write a pseudocode function to implement the **Next Greater Element** problem using a **stack**. For a given list of integers, find the next greater element for each element. The next greater element for an element *x* is the first element to its right in the list that is greater than *x*. If no such element exists, return -1 for that position.

Q6

- Write a pseudocode function to **implement a stack using two queues**. Your stack should support the operations Push(x) (to add an element to the stack) and Pop() (to remove the top element of the stack). Implement both operations using two queues.