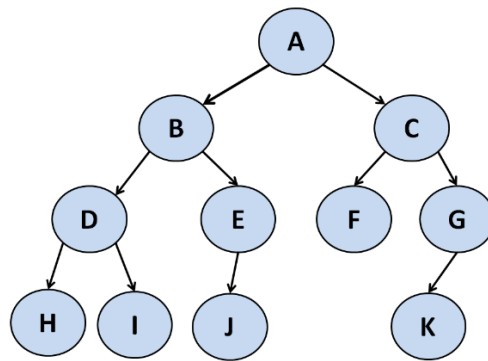


Assignment 2 of ELEC 278

Due: Nov.11, 2022, 11:59PM

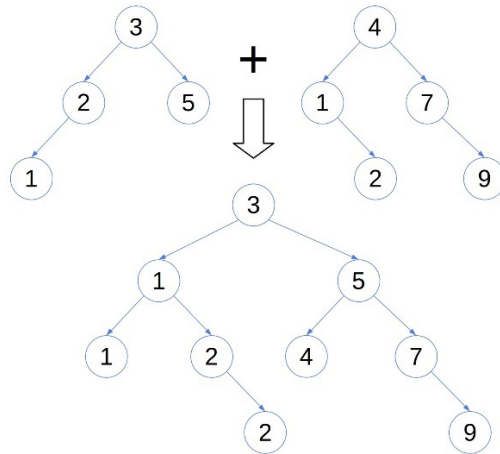
Q1: Given a binary tree, find the postorder traversal of the tree.

Tasks: 1) complete the function **postOrder()** which takes the root of the tree as parameter and returns a list containing the postorder traversal of the tree; 2) complete the function **Main()** that builds the tree below and outputs the list containing the postorder traversal of the tree. You are allowed to define other functions, such as **buildtree()**.



Q2: Given two binary search trees, return elements of both binary search trees in sorted form.

Tasks: 1) complete the function **Merge()** which takes roots of both binary search trees as its parameter and returns the binary search tree in a sorted order that merges the node values of both binary search trees; 2) complete the function **Main()** that creates two trees below and outputs an array of integers denoting the node values of the merged binary search tree in breadth-first traversal. You are allowed to define other functions, such as **buildtree()**, **breadthFirstTraversal()**.



Q4: Given a binary search tree that may be unbalanced, convert it into an AVL tree that has minimum possible height.

Tasks: 1) complete the function **isBalanced()** that takes the root node as parameter and returns true, if the tree is balanced else returns false. 2) complete the function **Rotation()** that takes the root of a binary search tree, converts it into an AVL tree that has the same node values, and returns the AVL tree. If there is more than one answer, return any of them. 2) complete the function **Main()** that creates the tree below and outputs if it is balanced or not. If not, convert it into an AVL tree and output an array of integers denoting the node values of the AVL tree in preorder traversal; otherwise, output the list containing the preorder traversal of the tree. You are allowed to define other functions, such as buildtree(), preorderTraversal().

