```
package Love_Babbar;

public class Trees {
   class Binary Tree{
         //
**********************************************************************************************************
*************** */
         //---Level order traversal
         //---Reverse Level Order traversal
         //---Height of a tree
         //---Diameter of a tree
         //---Mirror of a tree
         //Inorder Traversal of a tree both using recursion and Iteration
         //Preorder Traversal of a tree both using recursion and Iteration
         //Postorder Traversal of a tree both using recursion and Iteration
         //---Left View of a tree
         //---Right View of Tree
         //---Top View of a tree
         //---Bottom View of a tree
         //---Zig-Zag traversal of a binary tree
         //---Check if a tree is balanced or not
         //Diagonal Traversal of a Binary tree
         //---Boundary traversal of a Binary tree
         //Construct Binary Tree from String with Bracket Representation
         //Convert Binary tree into Doubled List
         //Convert Binary tree into Sum tree
         //---Construct Binary tree from Inorder and preorder traversal
         //Find minimum swaps required to convert a Binary tree into BST
         //Check if Binary tree is Sum tree or not
         //Check if all leaf nodes are at same level or not
         //Check if a Binary Tree contains duplicate subtrees of size 2 or more [ IMP ]
         //Check if 2 trees are mirror or not
         //Sum of Nodes on the Longest path from root to leaf node
         //Check if given graph is tree or not.  [ IMP ]
         //Find Largest subtree sum in a tree
         //Maximum Sum of nodes in Binary tree such that no two are adjacent
         //Print all “K” Sum paths in a Binary tree
         //---Find LCA in a Binary tree
         //Find distance between 2 nodes in a Binary tree
         //Kth Ancestor of node in a Binary tree
         //Find all Duplicate subtrees in a Binary tree [ IMP ]
         //Tree Isomorphism Problem
   }
}
```