**LAB # 11**

**Implementation of Binary Search Tree**

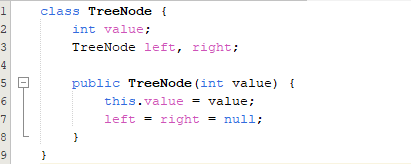
**OBJECTIVE:** Algorithm to implement binarysearch tree.

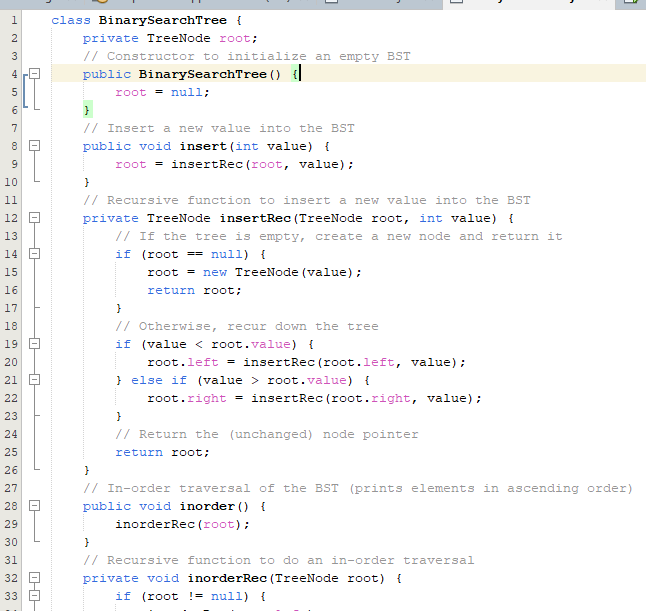
**Lab Task**

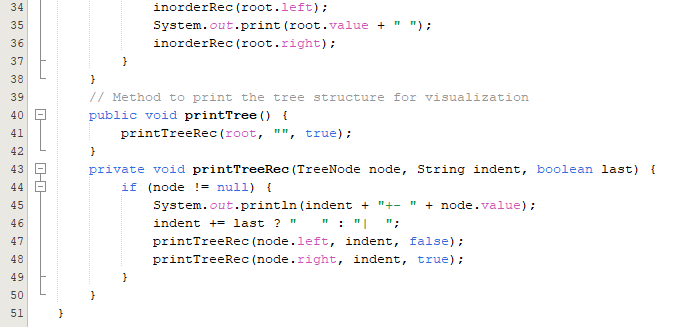
1. Write a program to create binary search tree of the following list of elements;

**- 45, 15, 79, 90, 10, 55, 12, 20, 50**

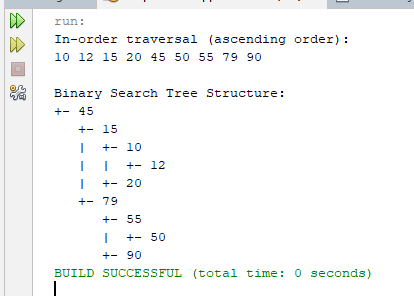
**CODE**

****

****

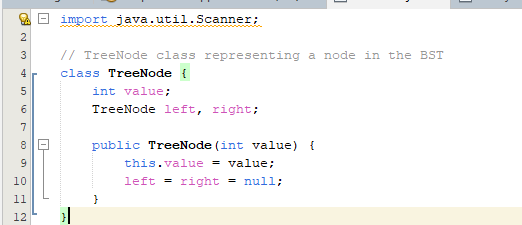
****

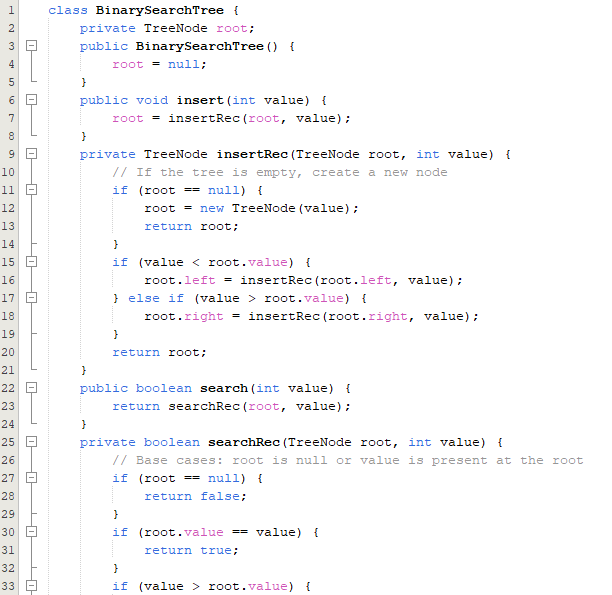
**OUTPUT**

****

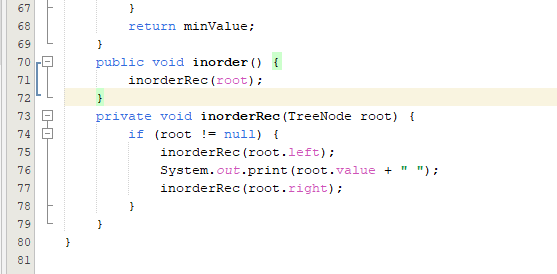
1. Write a program that takes an item input to search if there is an item present, deletes it otherwise insert it in a tree.

CODE

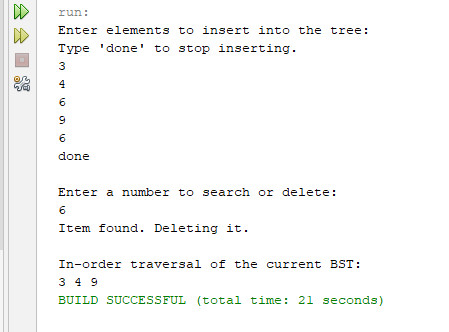






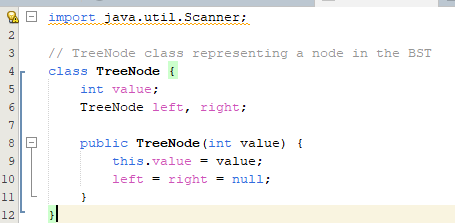
****

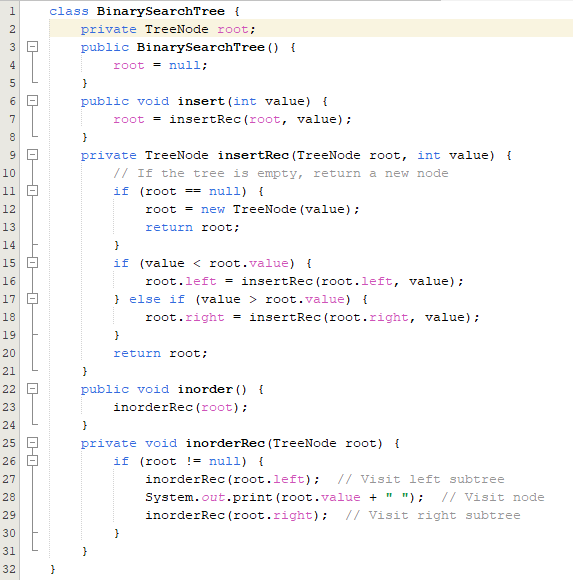
**OUTPUT**

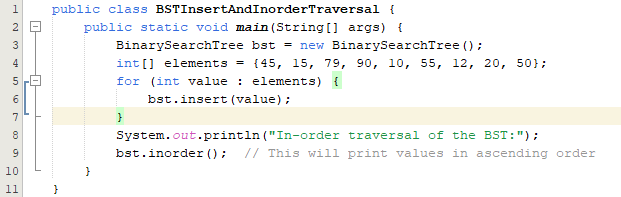
****

1. Insert multiple values into the tree, and perform an in-order traversal to verify the values are inserted correctly.

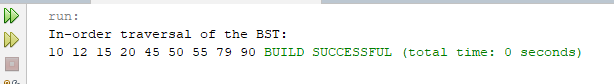
CODE





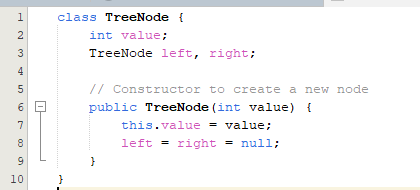


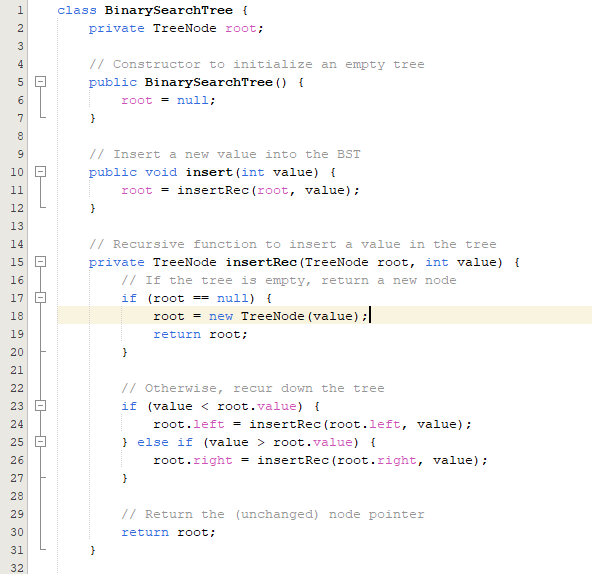
OUTPUT

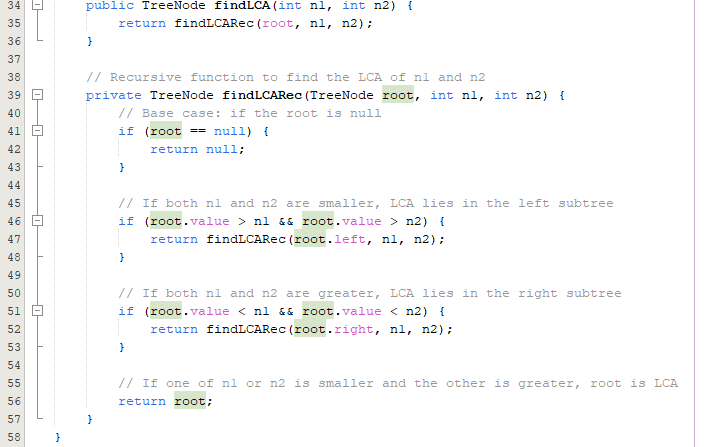


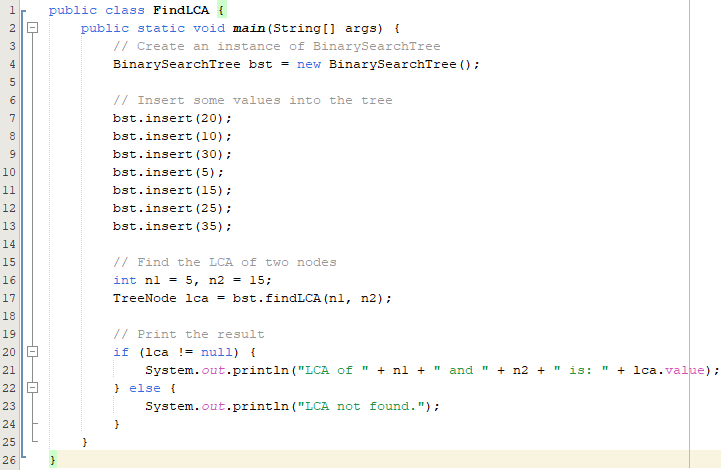
1. Find the lowest common ancestor (LCA) of two nodes in a Binary Search Tree.

CODE

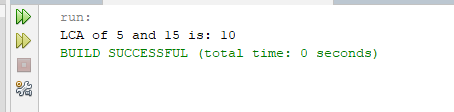






****

**OUTPUT**

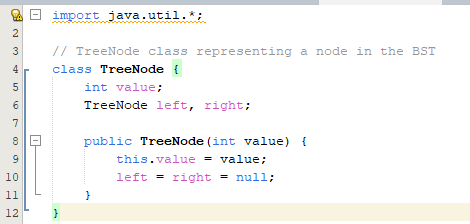
****

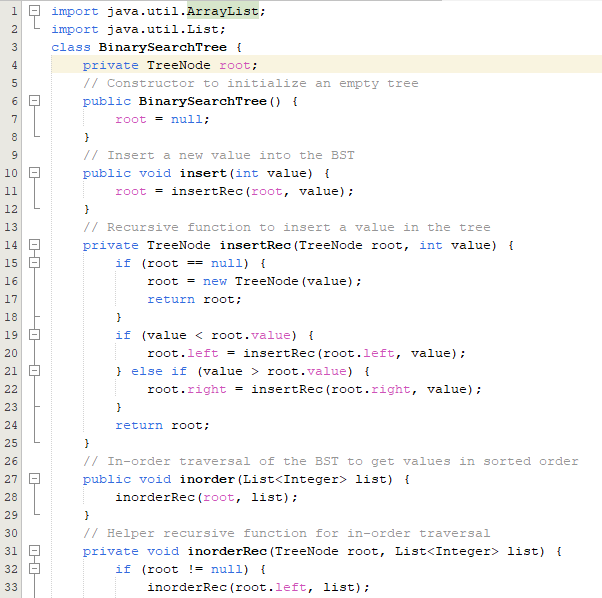
**Home Task**

1. Write a function called MergeBST that give a combine Binary search tree of two

different binary search trees according to rules of binary tree then print all nodes.

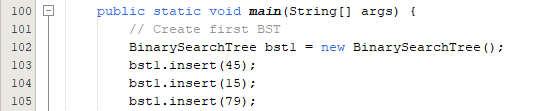
CODE

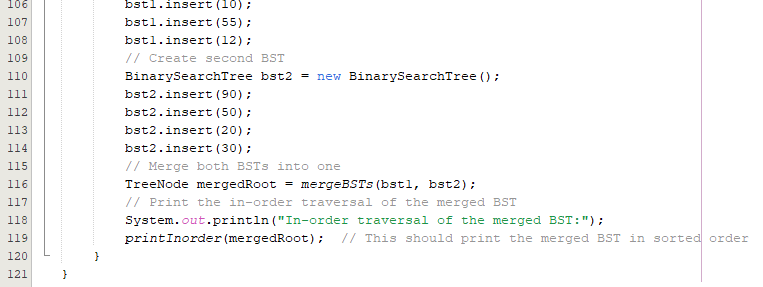




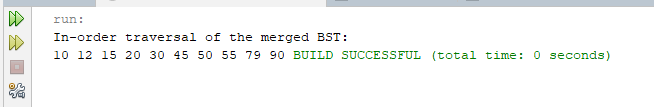






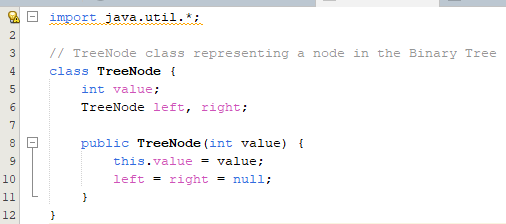


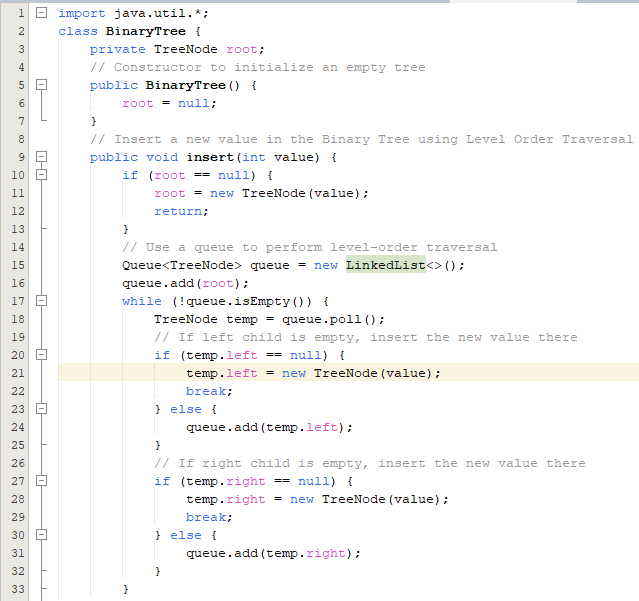
OUTPUT

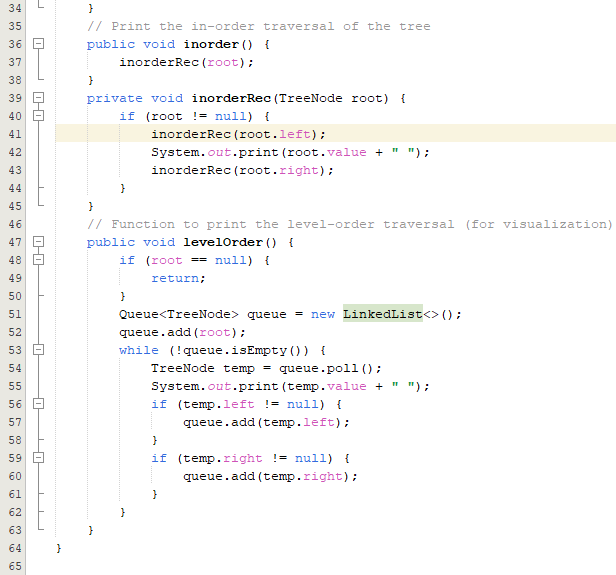


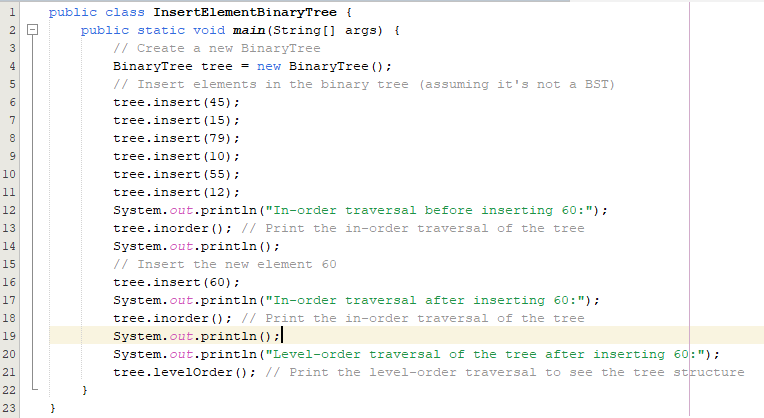
1. Write a program to insert new element ‘60’ in the Binary tree created in above ques.2.

CODE

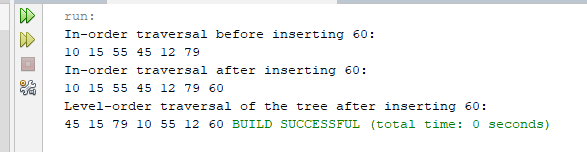








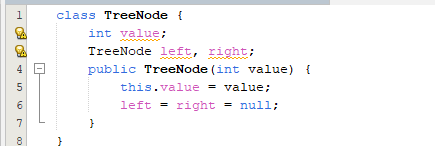
OUTPUT

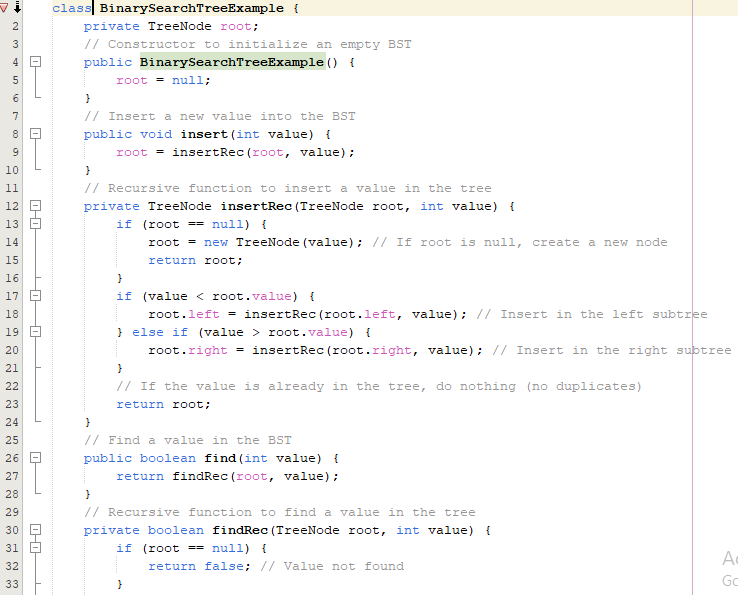


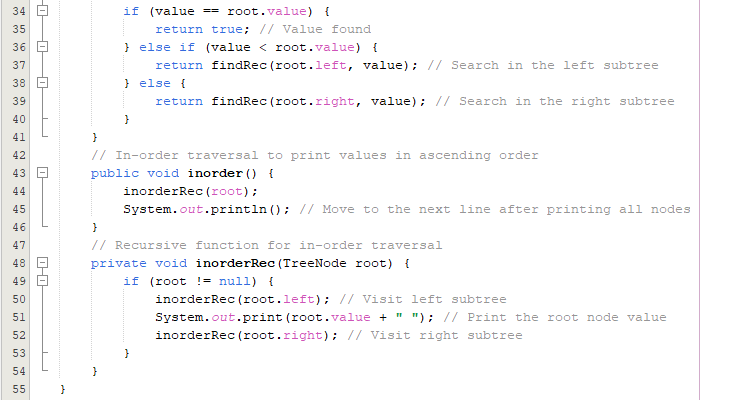
1. You are given a sequence of operations to be performed on a **Binary Search Tree (BST)**. The operations are:
2. **INSERT X**: Insert a new node with the value X into the BST. If a node with the value X already exists, do not insert it again.
3. **FIND X**: Check if the node with value X exists in the BST.
4. **INORDER**: Print all values in the BST in **ascending order** (inorder traversal).

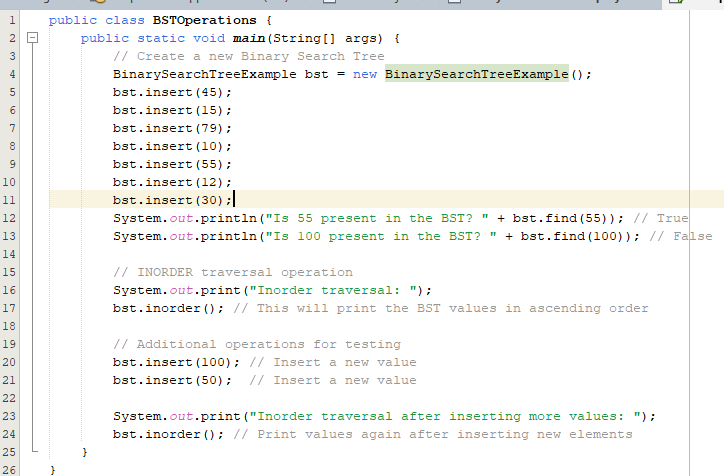
Your task is to implement a program that will execute the given sequence of operations and output the results accordingly.

CODE

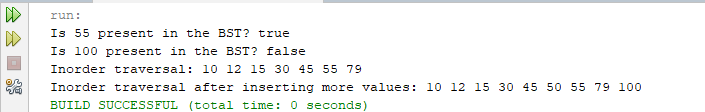






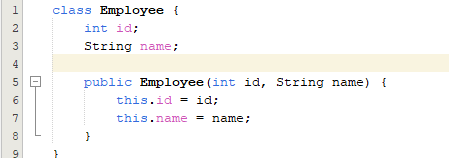


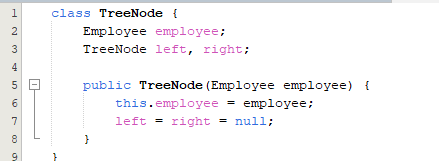
OUTPUT

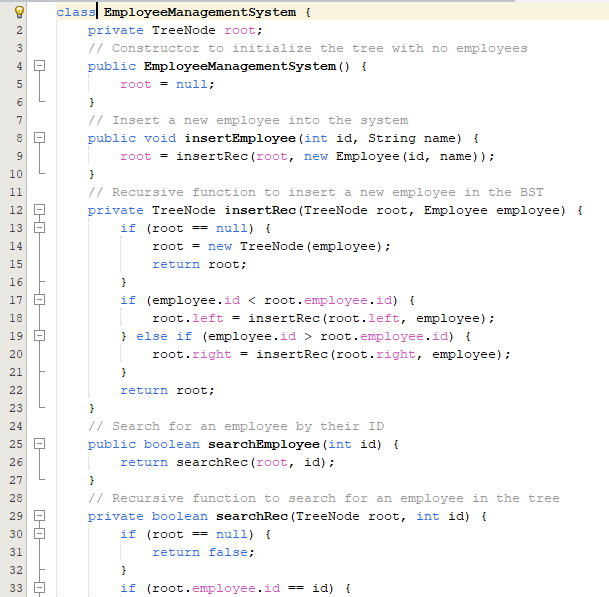


1. Create a task for employee management system. In an employee management system, each employee is represented by an ID, and you need to maintain an efficient system that allows you to:
2. Insert a new employee into the system.
3. Search for an employee by their ID.
4. Remove an employee from the system.
5. Display the employees in different sorted orders

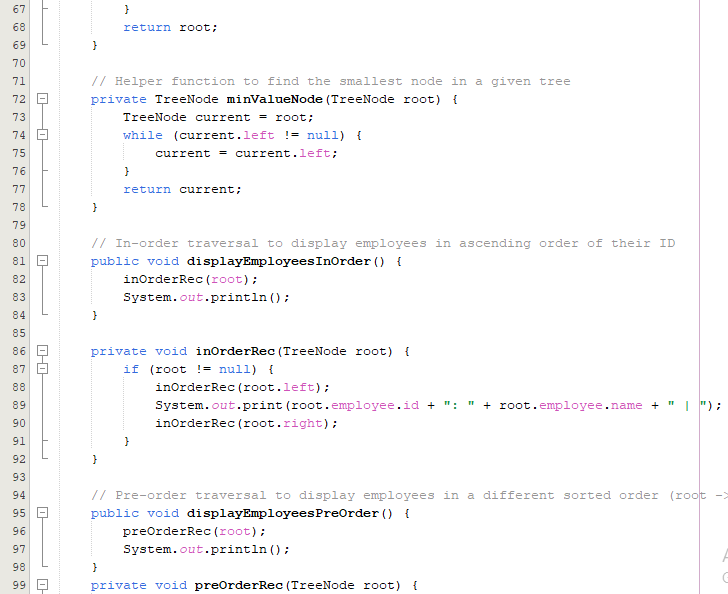
CODE



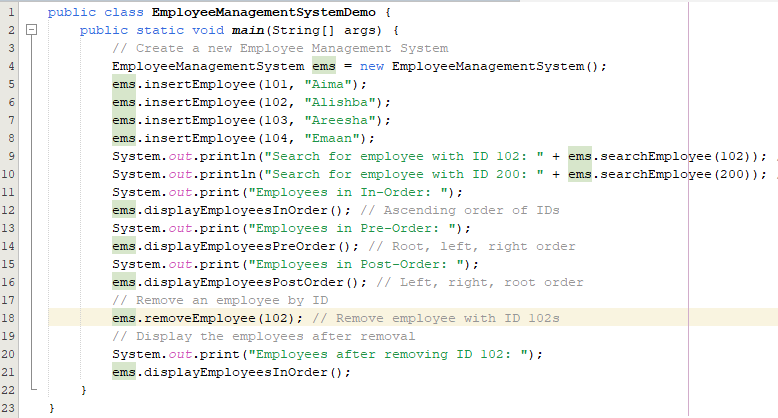












OUTPUT

