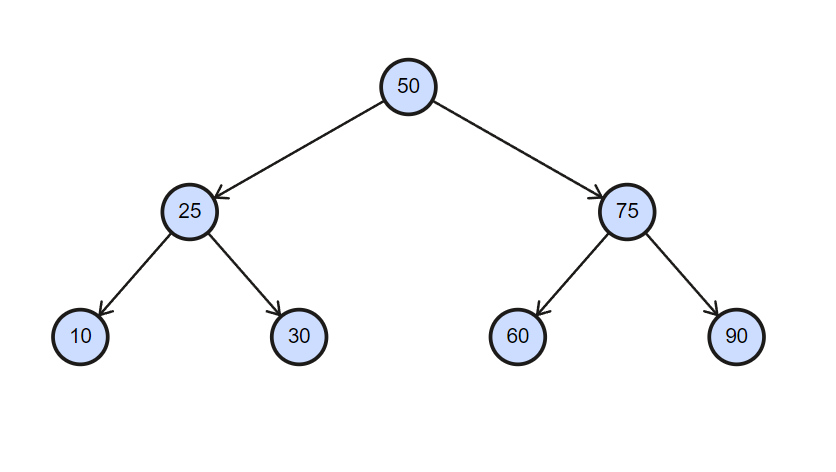
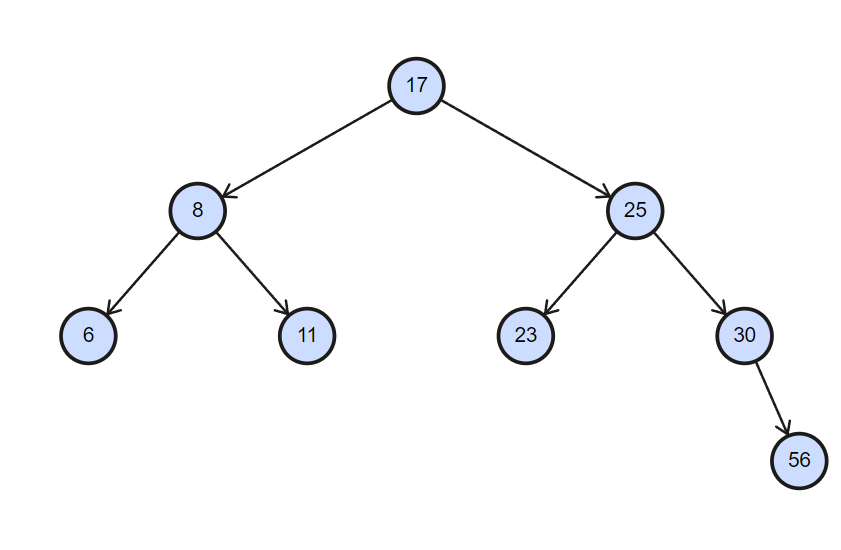
**01)**

**a)**

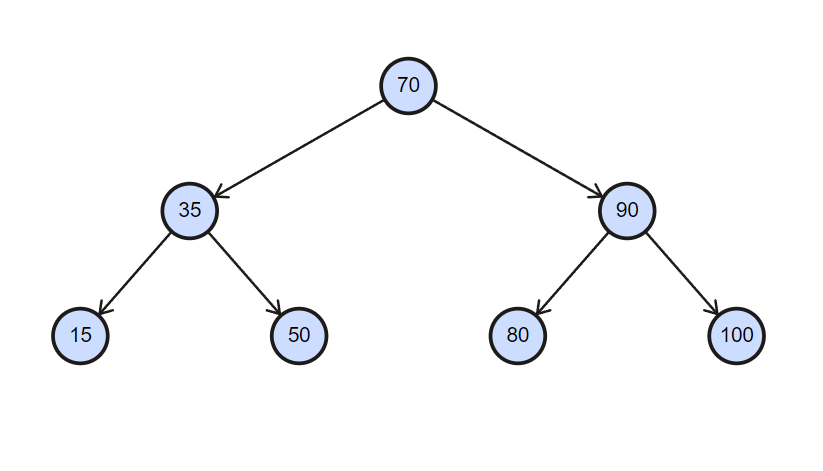
i)



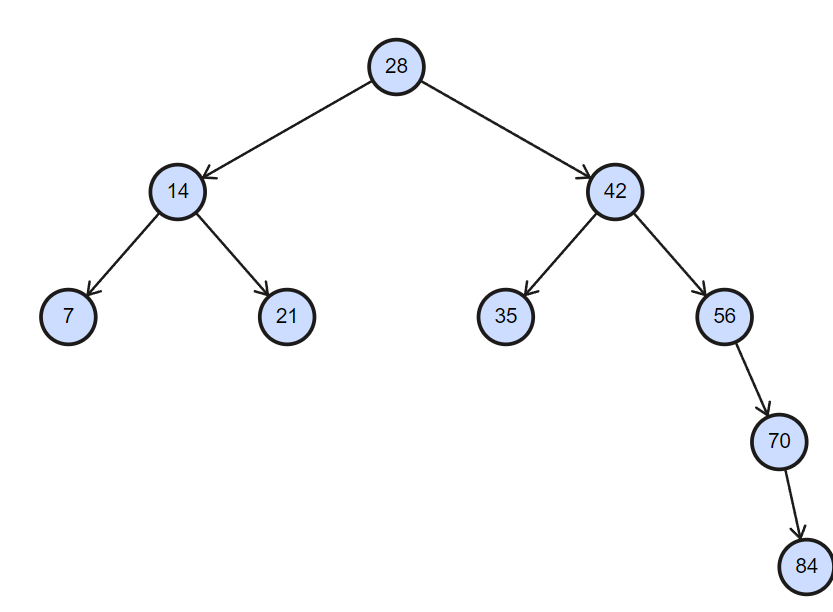
ii)



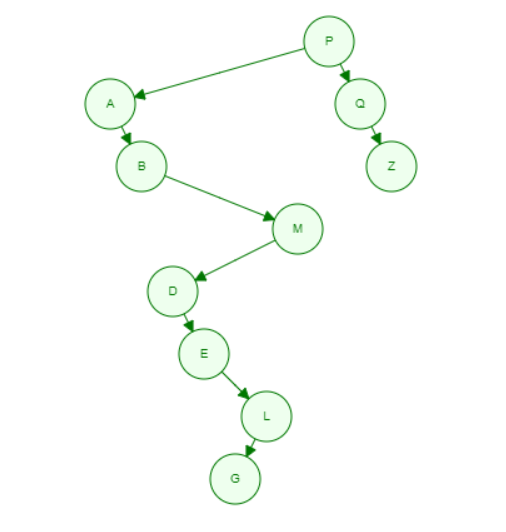
iii)



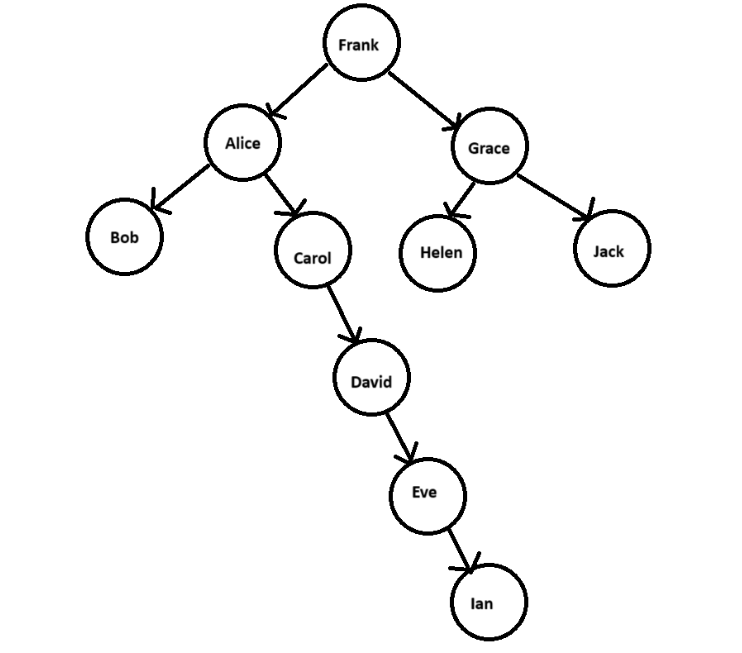
iv)



v)



vi)



**b)**

For Integers:

**IntNode Class:**

public class IntNode {  
 IntNode left,right;  
 int data;  
 public IntNode(int n){  
 left=null;  
 right=null;  
 data=n;  
 }  
}

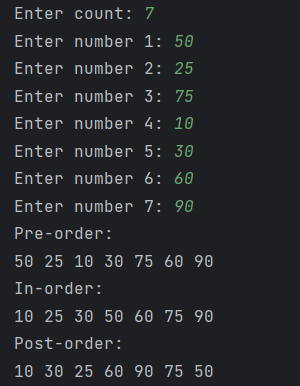
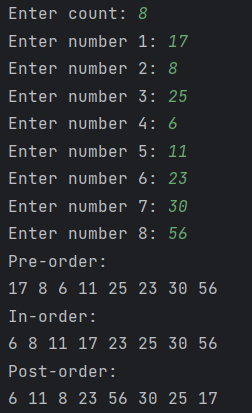
**IntTree Class:**

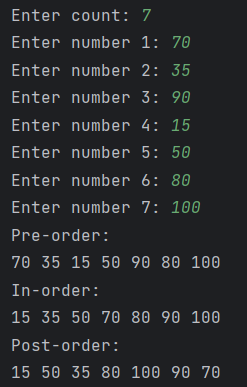
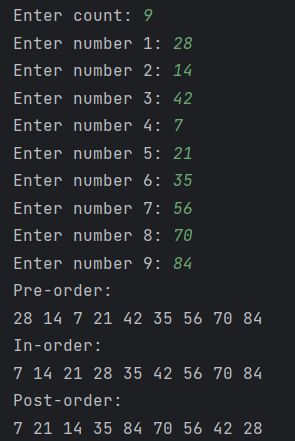
public class IntTree {  
 private IntNode root;  
  
 IntTree() {  
 root = null;  
 }  
  
 public void insert(int data) {  
 root = insert(root, data);  
 }  
  
 private IntNode insert(IntNode node, int data) {  
 if (node == null) {  
 node = new IntNode(data);  
 } else {  
 if (data <= node.data) {  
 node.left = insert(node.left, data);  
 } else {  
 node.right = insert(node.right, data);  
 }  
 }  
 return node;  
 }  
  
 public void preOrder() {  
 preOrder(root);  
 }  
  
 private void preOrder(IntNode r) {  
 if (r != null) {  
 System.*out*.print(r.data + " ");  
 preOrder(r.left);  
 preOrder(r.right);  
 }  
 }  
  
 public void inOrder() {  
 inOrder(root);  
 }  
  
 private void inOrder(IntNode r) {  
 if (r != null) {  
 inOrder(r.left);  
 System.*out*.print(r.data + " ");  
 inOrder(r.right);  
 }  
 }  
  
 public void postOrder() {  
 postOrder(root);  
 }  
  
 private void postOrder(IntNode r) {  
 if (r != null) {  
 postOrder(r.left);  
 postOrder(r.right);  
 System.*out*.print(r.data + " ");  
 }  
 }  
}

**NumberTree Class:**

import java.util.Scanner;  
  
public class NumberTree {  
 public static void main(String[] args) {  
 IntTree tree = new IntTree();  
 Scanner input=new Scanner(System.*in*);  
 System.*out*.print("Enter count: ");  
 int count=input.nextInt();  
 for (int i=1;i<=count;i++){  
 System.*out*.print("Enter number "+i+": ");  
 int num=input.nextInt();  
 tree.insert(num);  
 }  
 System.*out*.println("Pre-order: ");  
 tree.preOrder();  
 System.*out*.println();  
 System.*out*.println("In-order: ");  
 tree.inOrder();  
 System.*out*.println();  
 System.*out*.println("Post-order: ");  
 tree.postOrder();  
 }  
}

**Outputs:**

For Characters:

**CharNode Class:**

public class CharNode {  
 CharNode left,right;  
 char data;  
 public CharNode(char c) {  
 left = null;  
 right = null;  
 data=c;  
 }  
}

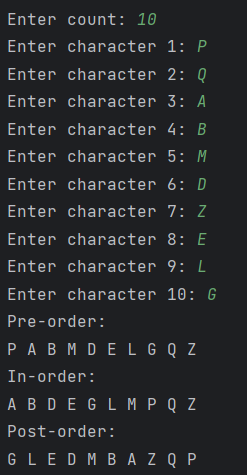
**CharTree Class:**

public class CharTree {  
 private CharNode root;  
 CharTree(){  
 root=null;  
 }  
 public void insert(char data){  
 root=insert(root,data);  
 }  
 private CharNode insert(CharNode node,char data){  
 if (node==null){  
 node=new CharNode(data);  
 }  
 else {  
 if (data<=node.data){  
 node.left=insert(node.left,data);  
 }  
 else {  
 node.right=insert(node.right,data);  
 }  
 }  
 return node;  
 }  
 public void preOrder(){  
 preOrder(root);  
 }  
 private void preOrder(CharNode r){  
 if (r!=null){  
 System.*out*.print(r.data+" ");  
 preOrder(r.left);  
 preOrder(r.right);  
 }  
 }  
 public void inOrder(){  
 inOrder(root);  
 }  
 private void inOrder(CharNode r){  
 if (r!=null){  
 inOrder(r.left);  
 System.*out*.print(r.data+" ");  
 inOrder(r.right);  
 }  
 }  
 public void postOrder(){  
 postOrder(root);  
 }  
 private void postOrder(CharNode r) {  
 if (r != null) {  
 postOrder(r.left);  
 postOrder(r.right);  
 System.*out*.print(r.data + " ");  
 }  
 }  
}

**LetterTree Class:**

import java.util.Scanner;  
  
public class LetterTree {  
 public static void main(String[] args) {  
 CharTree tree = new CharTree();  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.print("Enter count: ");  
 int count = input.nextInt();  
 for (int i = 1; i <= count; i++) {  
 System.*out*.print("Enter character " + i + ": ");  
 char ch = input.next().charAt(0);  
 tree.insert(ch);  
 }  
 System.*out*.println("Pre-order: ");  
 tree.preOrder();  
 System.*out*.println();  
 System.*out*.println("In-order: ");  
 tree.inOrder();  
 System.*out*.println();  
 System.*out*.println("Post-order: ");  
 tree.postOrder();  
 }  
}

**Output:**



For Strings:

**StringNode Class:**

public class StringNode {  
 StringNode left,right;  
 String data;  
 public StringNode(String str){  
 left=null;  
 right=null;  
 data=str;  
 }  
}

**StringTree Class:**

public class StringTree {  
 private StringNode root;  
 StringTree(){  
 root=null;  
 }  
 public void insert(String data){  
 root=insert(root,data);  
 }  
 private StringNode insert(StringNode node,String data){  
 if (node==null){  
 node=new StringNode(data);  
 }  
 else {  
 if (data.charAt(0)<=node.data.charAt(0)){  
 node.left=insert(node.left,data);  
 }  
 else {  
 node.right=insert(node.right,data);  
 }  
 }  
 return node;  
 }  
 public void preOrder(){  
 preOrder(root);  
 }  
 private void preOrder(StringNode r){  
 if (r!=null){  
 System.*out*.print(r.data+" ");  
 preOrder(r.left);  
 preOrder(r.right);  
 }  
 }  
 public void inOrder(){  
 inOrder(root);  
 }  
 private void inOrder(StringNode r){  
 if (r!=null){  
 inOrder(r.left);  
 System.*out*.print(r.data+" ");  
 inOrder(r.right);  
 }  
 }  
 public void postOrder(){  
 postOrder(root);  
 }  
 private void postOrder(StringNode r){  
 if (r!=null){  
 postOrder(r.left);  
 postOrder(r.right);  
 System.*out*.print(r.data+" ");  
 }  
 }  
}

**WordTree Class:**

import java.util.Scanner;  
  
public class WordTree {  
 public static void main(String[] args) {  
 StringTree t6=new StringTree();  
 Scanner input=new Scanner(System.*in*);  
 System.*out*.print("Enter count: ");  
 int count=input.nextInt();  
 input.nextLine();  
 for (int i=0;i<count;i++){  
 System.*out*.print("Enter string "+(i+1)+": ");  
 String str=input.nextLine();  
 t6.insert(str);  
 }  
 System.*out*.println("Pre-order: ");  
 t6.preOrder();  
 System.*out*.println();  
 System.*out*.println("In-order: ");  
 t6.inOrder();  
 System.*out*.println();  
 System.*out*.println("Post-order: ");  
 t6.postOrder();  
 }  
}

**Output:**

