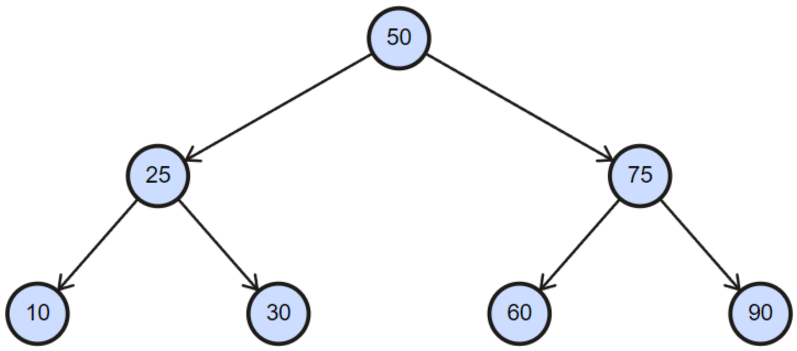
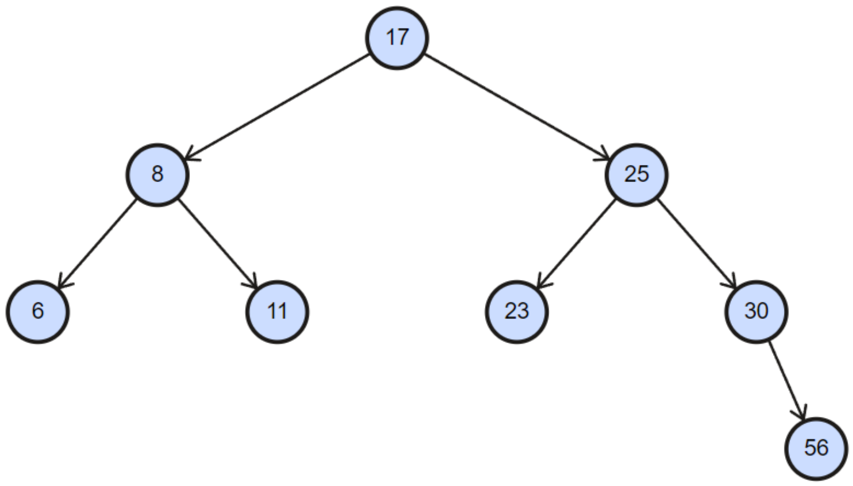
01)

a)

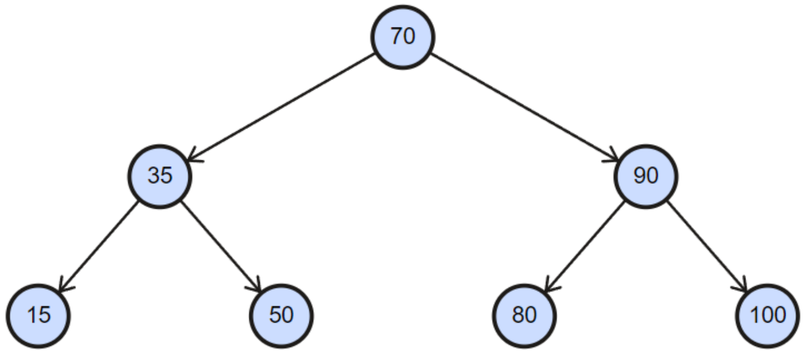
i)



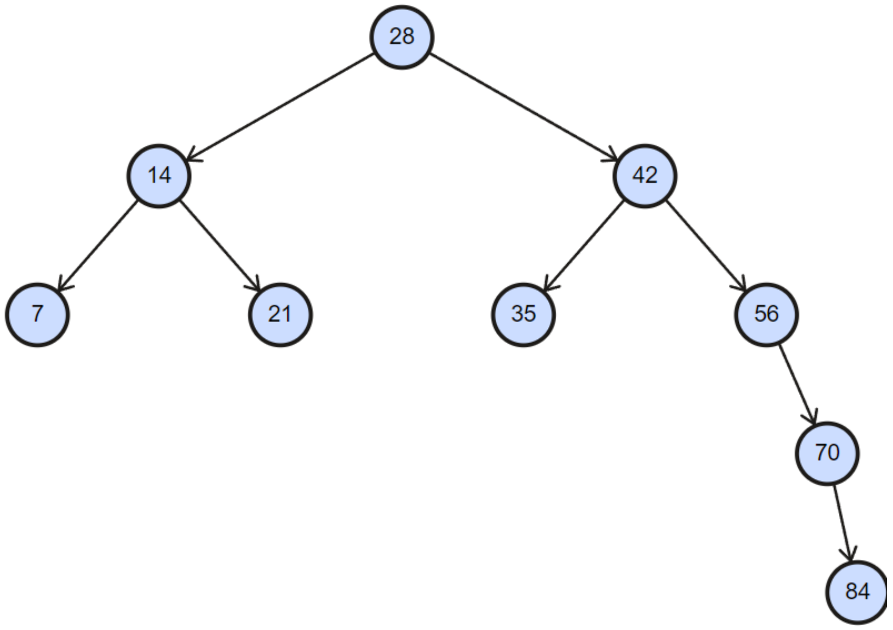
ii)



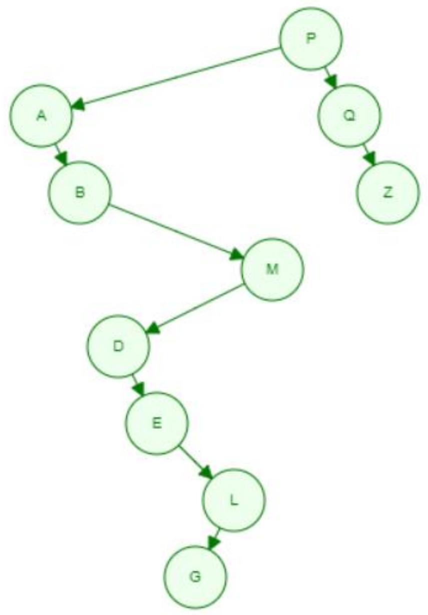
iii)



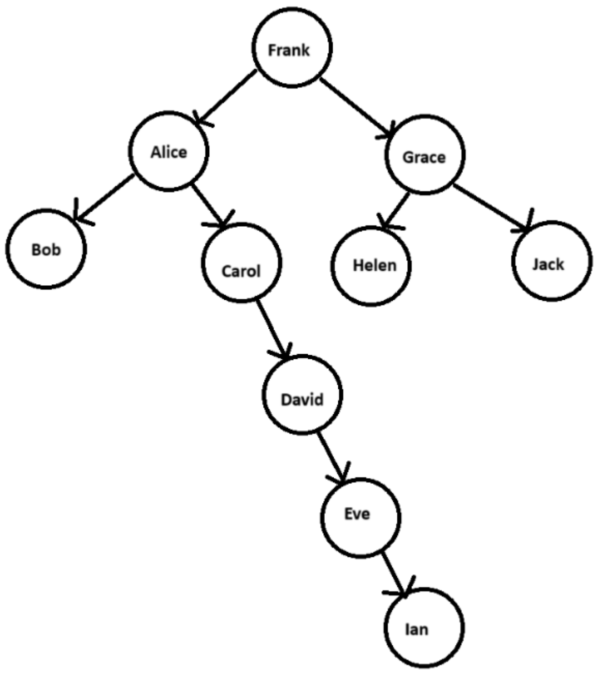
iv)



v)


P
A
Q
B
Z
M
D
E
L
G

vi)


Frank
Alice
Grace
Bob
Carol
Helen
Jack
David
Eve
Ian

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:

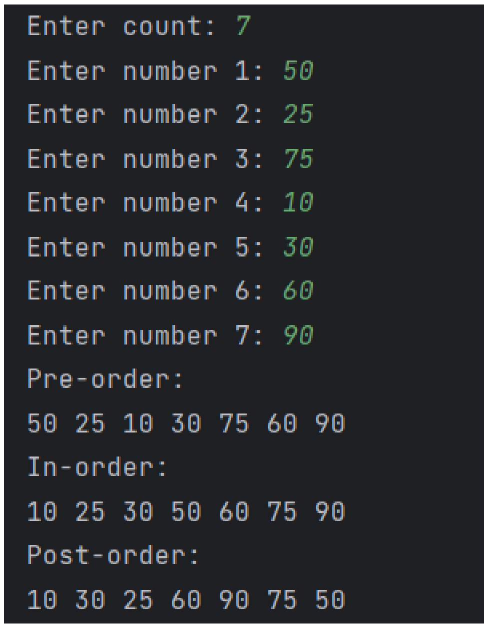
```
public class IntTree {
    private IntNode root;

    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);
        }
    }
```

```java
    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 + " ");
        }
    }
}
```

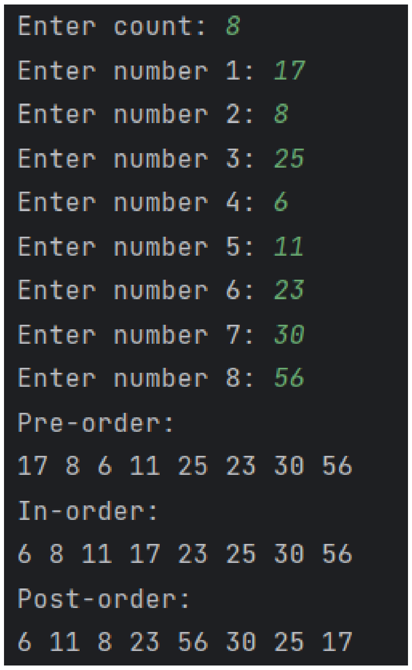
NumberTree Class:

```java
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();
    }
}
```

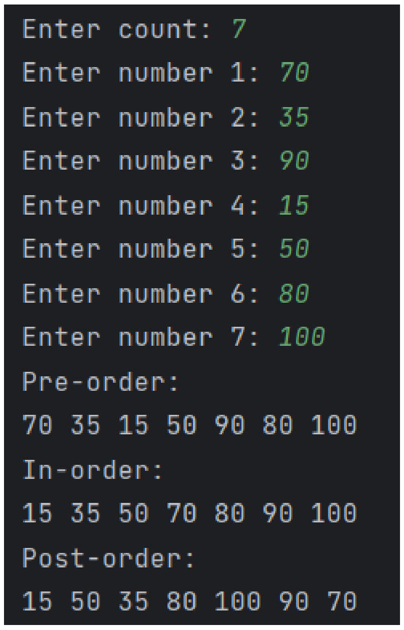
Outputs:



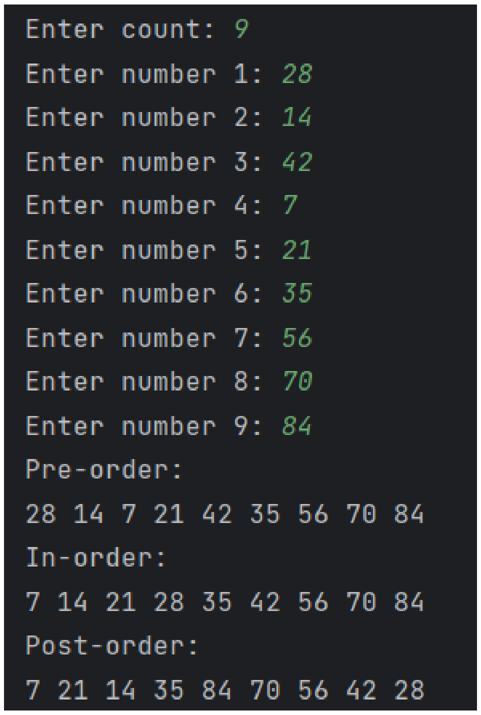
```
Enter count: 7
Enter number 1: 50
Enter number 2: 25
Enter number 3: 75
Enter number 4: 10
Enter number 5: 30
Enter number 6: 60
Enter number 7: 90
Pre-order:
50 25 10 30 75 60 90
In-order:
10 25 30 50 60 75 90
Post-order:
10 30 25 60 90 75 50
```



```
Enter count: 8
Enter number 1: 17
Enter number 2: 8
Enter number 3: 25
Enter number 4: 6
Enter number 5: 11
Enter number 6: 23
Enter number 7: 30
Enter number 8: 56
Pre-order:
17 8 6 11 25 23 30 56
In-order:
6 8 11 17 23 25 30 56
Post-order:
6 11 8 23 56 30 25 17
```



```
Enter count: 7
Enter number 1: 70
Enter number 2: 35
Enter number 3: 90
Enter number 4: 15
Enter number 5: 50
Enter number 6: 80
Enter number 7: 100
Pre-order:
70 35 15 50 90 80 100
In-order:
15 35 50 70 80 90 100
Post-order:
15 50 35 80 100 90 70
```



```
Enter count: 9
Enter number 1: 28
Enter number 2: 14
Enter number 3: 42
Enter number 4: 7
Enter number 5: 21
Enter number 6: 35
Enter number 7: 56
Enter number 8: 70
Enter number 9: 84
Pre-order:
28 14 7 21 42 35 56 70 84
In-order:
7 14 21 28 35 42 56 70 84
Post-order:
7 21 14 35 84 70 56 42 28
```

For Characters:

CharNode Class:

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

CharTree Class:

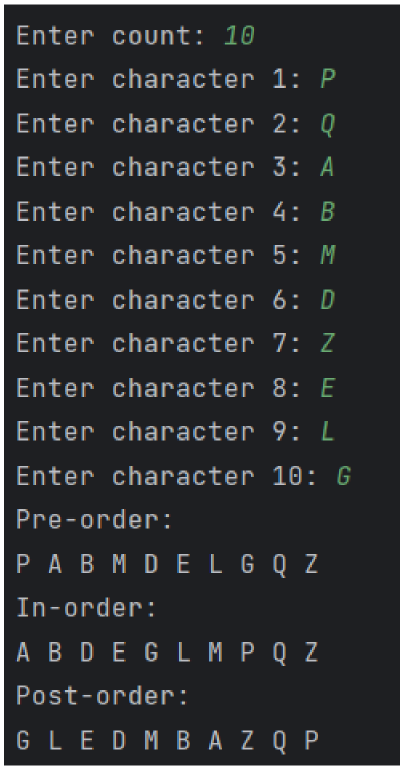
```
public class CharTree {
    private CharNode root;
    CharTree(){
        root=null;
    }
    public void insert(char data){
        root=insert(root,data);
    }
    private CharNode insert(CharNode node,char data){
        if (node==null){
            node=new CharNode(data);
        }
        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+" ");
```

```java
            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:

```java
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:



```
Enter count: 10
Enter character 1: P
Enter character 2: Q
Enter character 3: A
Enter character 4: B
Enter character 5: M
Enter character 6: D
Enter character 7: Z
Enter character 8: E
Enter character 9: L
Enter character 10: G
Pre-order:
P A B M D E L G Q Z
In-order:
A B D E G L M P Q Z
Post-order:
G L E D M B A Z Q P
```

For Strings:

StringNode Class:

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

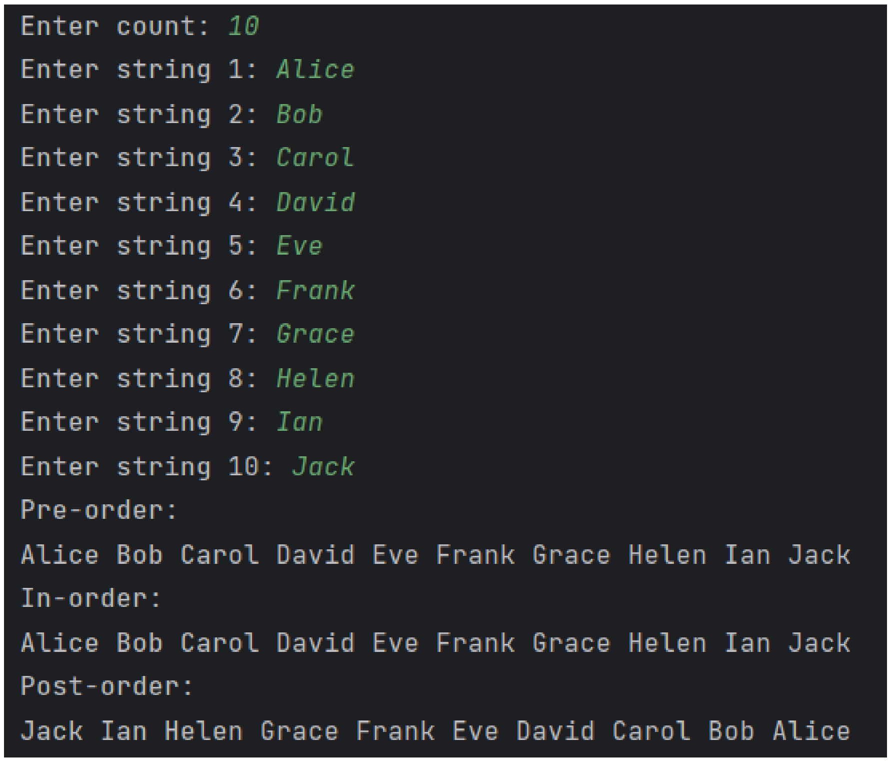
StringTree Class:

```java
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:

```java
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:



```
Enter count: 10
Enter string 1: Alice
Enter string 2: Bob
Enter string 3: Carol
Enter string 4: David
Enter string 5: Eve
Enter string 6: Frank
Enter string 7: Grace
Enter string 8: Helen
Enter string 9: Ian
Enter string 10: Jack
Pre-order:
Alice Bob Carol David Eve Frank Grace Helen Ian Jack
In-order:
Alice Bob Carol David Eve Frank Grace Helen Ian Jack
Post-order:
Jack Ian Helen Grace Frank Eve David Carol Bob Alice
```