

```

import java.io.*;
import java.util.*;
public class Lab1{
    public static void main( String[] args){
        System.out.println("");
        System.out.println("Use of AVL tree with random numbers.");
        System.out.println("");
        AvlRandom();
        System.out.println("");
        System.out.println("Use of Binary Search Tree with random numbers.");
        System.out.println("");
        BSTRandom();
        System.out.println("");
        System.out.println("Use of AVL tree with ascending numbers.");
        System.out.println("");
        AvlAscend();
        System.out.println("");
        System.out.println("Use of Binary Search Tree with ascending numbers.");
        System.out.println("");
        BSTAscend();
        System.out.println("");
        System.out.println("Use of AVL tree with descending numbers.");
        System.out.println("");
        AvlDescend();
        System.out.println("");
        System.out.println("Use of Binary Search Tree with descending numbers.");
        System.out.println("");
        BSTDescend();
    }
    public static void AvlRandom(){
        AvlTree MyAVL = new AvlTree();

        String s = "";
        String newInfo = "";
        try {
            BufferedReader MyBufferedReader = new BufferedReader(new FileReader("Random.txt"));
            s = MyBufferedReader.readLine() ;
            while (s != null) {
                newInfo = newInfo + s + " ";
                s = MyBufferedReader.readLine() ;
            }
        } catch (IOException e) { System.out.println(e) ; }
        //System.out.println(newInfo);
        StringTokenizer st = new StringTokenizer(newInfo);
        Comparable x;

        for(int i = 0 ; i < 50; i++){
            if (st.hasMoreTokens()){
                x = (Comparable)(new Integer(Integer.parseInt(st.nextToken())));
                MyAVL.insert(x);
            }
        }
        MyAVL.printTree();
        System.out.println("");
        System.out.println("There are " + MyAVL.numberOfNodes() + " Nodes. ");
        System.out.println("The height is " + MyAVL.getHeight() + ". ");
    } //end AvlRandom
    public static void BSTRandom(){
        BST MyBTree = new BST();

        String s = "";
        String newInfo = "";
        try {
            BufferedReader MyBufferedReader = new BufferedReader(new FileReader("Random.txt"));
            s = MyBufferedReader.readLine() ;
            while (s != null) {
                newInfo = newInfo + s + " ";
                s = MyBufferedReader.readLine() ;
            }
        } catch (IOException e) { System.out.println(e) ; }
        //System.out.println(newInfo);
    }
}

```

```

StringTokenizer st = new StringTokenizer(newInfo);
Comparable x;

for(int i = 0 ; i < 50; i++){
    if (st.hasMoreTokens()){
        x = (Comparable)(new Integer(Integer.parseInt(st.nextToken())));
        MyBTree.insert(x);
    }
}
MyBTree.printTree();
System.out.println("");
System.out.println("There are " + MyBTree.numberOfNodes() + " Nodes.");
System.out.println("The height is " + MyBTree.getHeight() + ".");
} //end BSTRandom
public static void AvlAscend(){
    AvlTree MyAVL = new AvlTree();

    String s = "";
    String newInfo = "";
    try {
        BufferedReader MyBufferedReader = new BufferedReader(new FileReader("Ascend.txt"));
        s = MyBufferedReader.readLine();
        while (s != null) {
            newInfo = newInfo + s + " ";
            s = MyBufferedReader.readLine();
        }
    } catch (IOException e) { System.out.println(e); }
    //System.out.println(newInfo);
    StringTokenizer st = new StringTokenizer(newInfo);
    Comparable x;

    for(int i = 0 ; i < 50; i++){
        if (st.hasMoreTokens()){
            x = (Comparable)(new Integer(Integer.parseInt(st.nextToken())));
            MyAVL.insert(x);
        }
    }
    MyAVL.printTree();
    System.out.println("");
    System.out.println("There are " + MyAVL.numberOfNodes() + " Nodes.");
    System.out.println("The height is " + MyAVL.getHeight() + ".");
} //end AvlAscend
public static void AvlDescend(){
    AvlTree MyAVL = new AvlTree();

    String s = "";
    String newInfo = "";
    try {
        BufferedReader MyBufferedReader = new BufferedReader(new FileReader("Descend.txt"));
        s = MyBufferedReader.readLine();
        while (s != null) {
            newInfo = newInfo + s + " ";
            s = MyBufferedReader.readLine();
        }
    } catch (IOException e) { System.out.println(e); }
    //System.out.println(newInfo);
    StringTokenizer st = new StringTokenizer(newInfo);
    Comparable x;

    for(int i = 0 ; i < 50; i++){
        if (st.hasMoreTokens()){
            x = (Comparable)(new Integer(Integer.parseInt(st.nextToken())));
            MyAVL.insert(x);
        }
    }
    MyAVL.printTree();
    System.out.println("");
    System.out.println("There are " + MyAVL.numberOfNodes() + " Nodes.");
    System.out.println("The height is " + MyAVL.getHeight() + ".");
} //end AvlDescend
public static void BSTAscend(){

```

```

    BST MyBTree = new BST();

    String s = "";
    String newInfo = "";
    try {
        BufferedReader MyBufferedReader = new BufferedReader(new FileReader("Ascend.txt"));
        s = MyBufferedReader.readLine() ;
        while (s != null) {
            newInfo = newInfo + s + " ";
            s = MyBufferedReader.readLine() ;
        }
    }catch (IOException e) { System.out.println(e) ; }
    //System.out.println(newInfo);
    StringTokenizer st = new StringTokenizer(newInfo);
    Comparable x;

    for(int i = 0 ; i < 50; i++){
        if (st.hasMoreTokens()){
            x = (Comparable)(new Integer(Integer.parseInt(st.nextToken())));
            MyBTree.insert(x);
        }
    }
    MyBTree.printTree();
    System.out.println("");
    System.out.println("There are " + MyBTree.numberOfNodes() + " Nodes.");
    System.out.println("The height is " + MyBTree.getHeight() + ".");
} //end BSTAscend
public static void BSTDescend(){
    BST MyBTree = new BST();

    String s = "";
    String newInfo = "";
    try {
        BufferedReader MyBufferedReader = new BufferedReader(new FileReader("Descend.txt"));
        s = MyBufferedReader.readLine() ;
        while (s != null) {
            newInfo = newInfo + s + " ";
            s = MyBufferedReader.readLine() ;
        }
    }catch (IOException e) { System.out.println(e) ; }
    //System.out.println(newInfo);
    StringTokenizer st = new StringTokenizer(newInfo);
    Comparable x;

    for(int i = 0 ; i < 50; i++){
        if (st.hasMoreTokens()){
            x = (Comparable)(new Integer(Integer.parseInt(st.nextToken())));
            MyBTree.insert(x);
        }
    }
    MyBTree.printTree();
    System.out.println("");
    System.out.println("There are " + MyBTree.numberOfNodes() + " Nodes.");
    System.out.println("The height is " + MyBTree.getHeight() + ".");
} //end BSTDescend
} //end Lab1

```