```c
//Author:DEADPOOL
//User@DEADPOOL
//Device name:LAPTOP-MGJPSU5N
//**************************
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#include<windows.h>
typedef struct tnode{
    int value;
    struct tnode *left;
    struct tnode *right;
}tnode;
tnode* create_tnode(int value){
    tnode* new_node =malloc(sizeof(tnode));
    if (new_node!=NULL){
        new_node->left=NULL;
        new_node->right=NULL;
        new_node->value=value;
    }
    return new_node;
}

int int_max(int a, int b)
{
    return (a > b)? a : b;
}
//calculating the height of tree in a recursive function
int height(tnode *N){
    if (N == NULL)
        return 0;
    return 1 + int_max(height(N->left),height(N->right));// recursive call
}
//counting the number of leaf nodes in a recursive function
int count_leaf_node(tnode* root){

    if (root==NULL)
        return 0;
    else if (root->left==NULL&&root->right==NULL)
        return 1;
    return (count_leaf_node(root->left)+count_leaf_node(root->right));//recursive
call


}

tnode* insert(tnode* node, int value){
    /* 1.  Perform the normal BST insertion */
    if (node == NULL)
        return(create_tnode(value));

    if (value < node->value)
        node->left  = insert(node->left, value);
    else if (value > node->value)
        node->right = insert(node->right, value);
    else // Equal values are not allowed in BST
        return node;


}

//functions to print a tree (AVL tree)
//this delay function will helps to view the output data properly
void delay(unsigned int mseconds){
    clock_t goal = mseconds + clock();
    while (goal > clock());
```

```
 66  }
 67
 68  void print_format(int num_of_char){
 69          delay(45);
 70          printf("%c",219);
 71      for(int i=0;i<num_of_char;i++){
 72          delay(90);
 73          printf("        %c",219);
 74      }
 75      printf("%c%c%c%c",254,254,254,254);
 76  }
 77
 78  // pre-order traversal
 79  void pre_order_print_tree(tnode* root,int level){
 80      if (root==NULL){
 81          print_format(level);
 82          printf("...\n");
 83          return;
 84      }
 85      print_format(level);
 86      printf("%d(L:%d)\n",root->value,level);
 87      print_format(level);
 88      printf(" Left\n");
 89      pre_order_print_tree(root->left,level+1);
 90      print_format(level);
 91      printf("Right\n");
 92      pre_order_print_tree(root->right,level+1);
 93  }
 94
 95
 96  int main(){
 97      int choice=1,value;
 98     tnode* root=NULL;
 99     while (1){
100          if (choice==1){
101              system("cls");
102              printf("\nEnter the value : ");
103              scanf("%d",&value);
104              root=insert(root,value);
105              //calling height function and count leaf node function
106              printf("\nHEIGHT : %d \nNUMBER OF LEAF NODES : %d\n",height(root),
count_leaf_node(root));
107              printf("\n\n\n PreOrder view \n");
108              pre_order_print_tree(root,0);
109          }
110          else{
111              return 0;
112          }
113      printf("\n Enter 1 to insert another node :");
114      scanf("%d",&choice);
115  }
116
117  return 0;
118  }
```