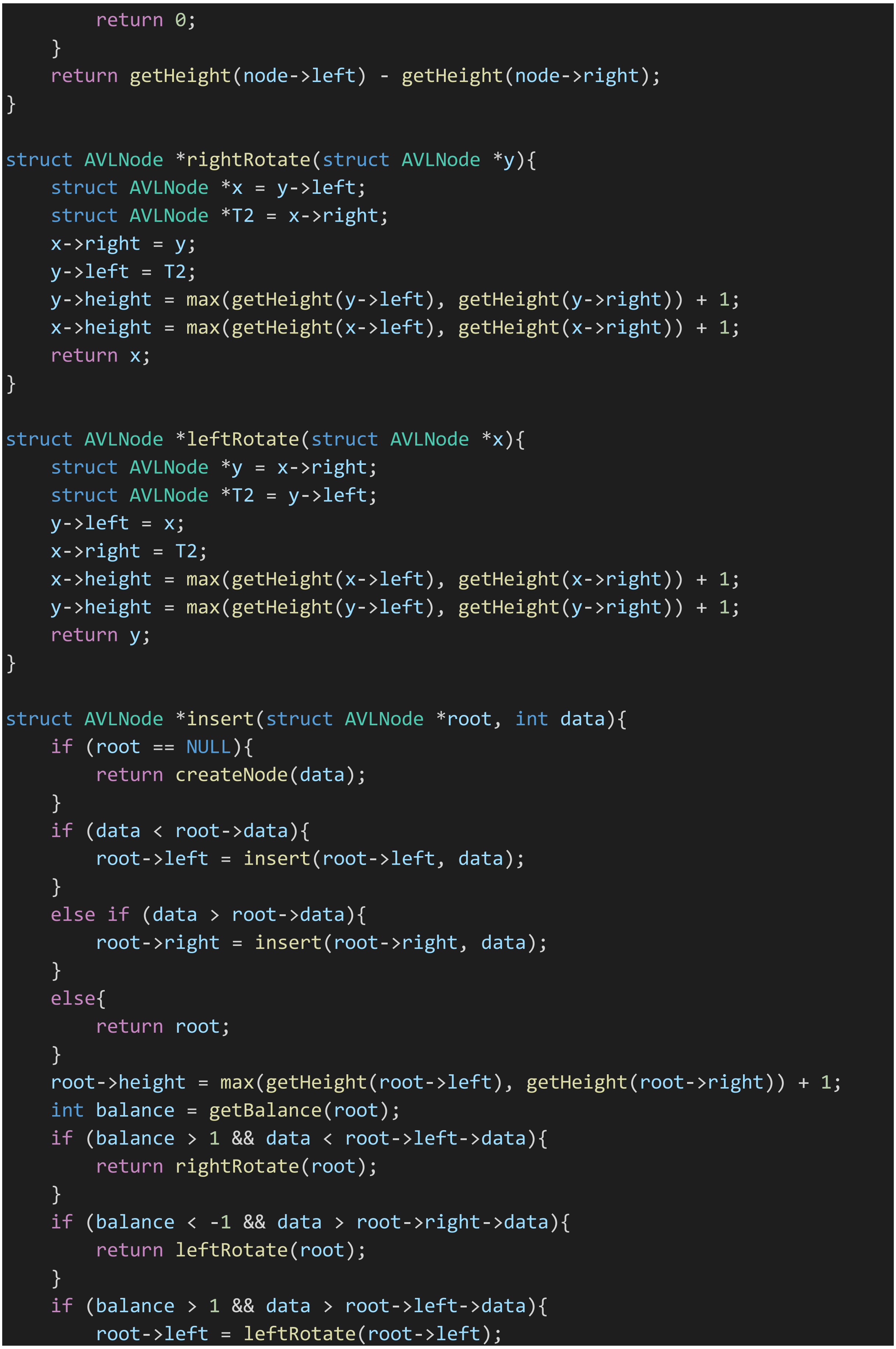
DS Lab Assignment 9

Pranav Joshi
CS-B Batch 2
Roll no: 43

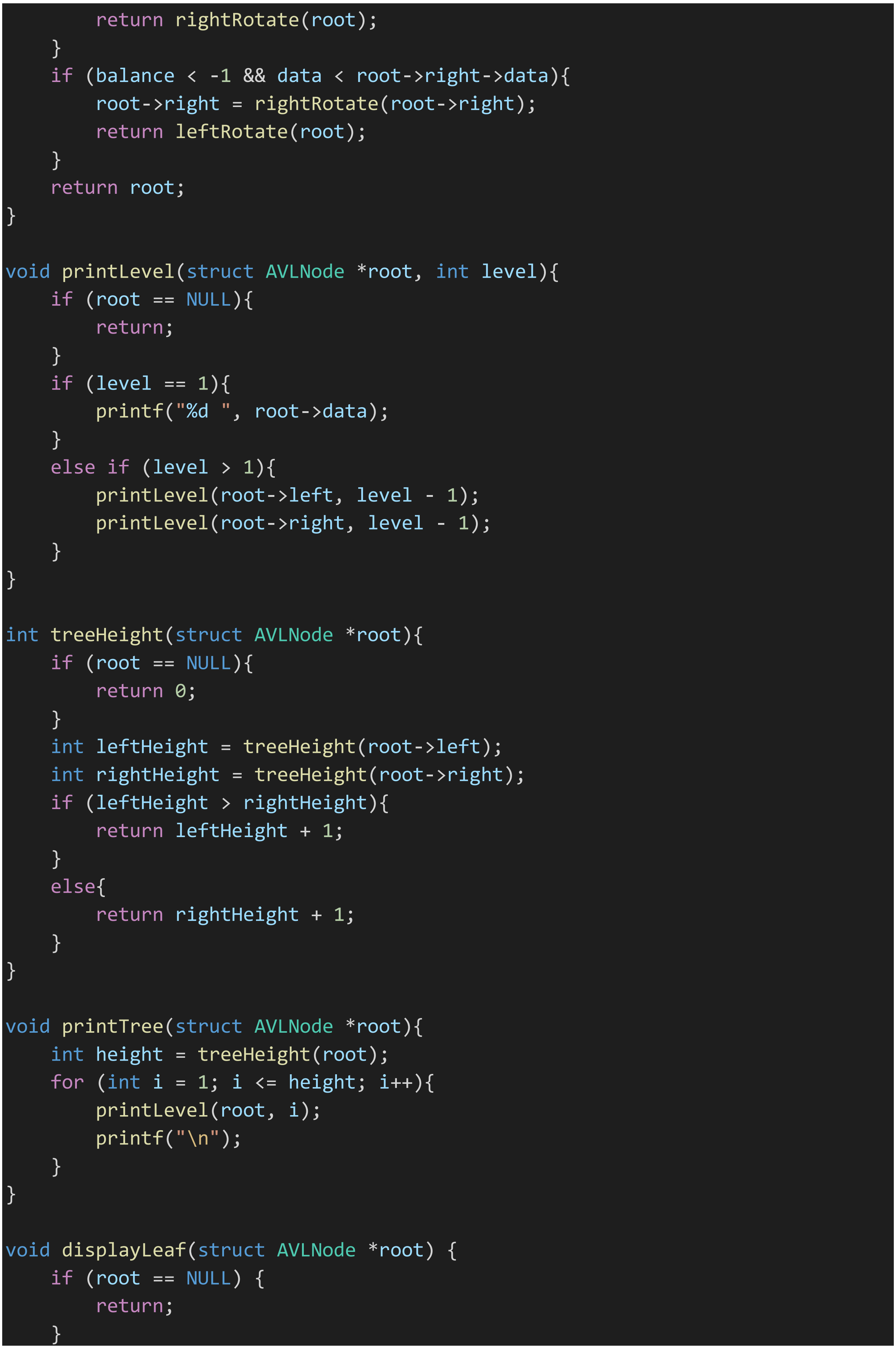
Title: WAP to Implement AVL tree and display it levelwise.

Code:

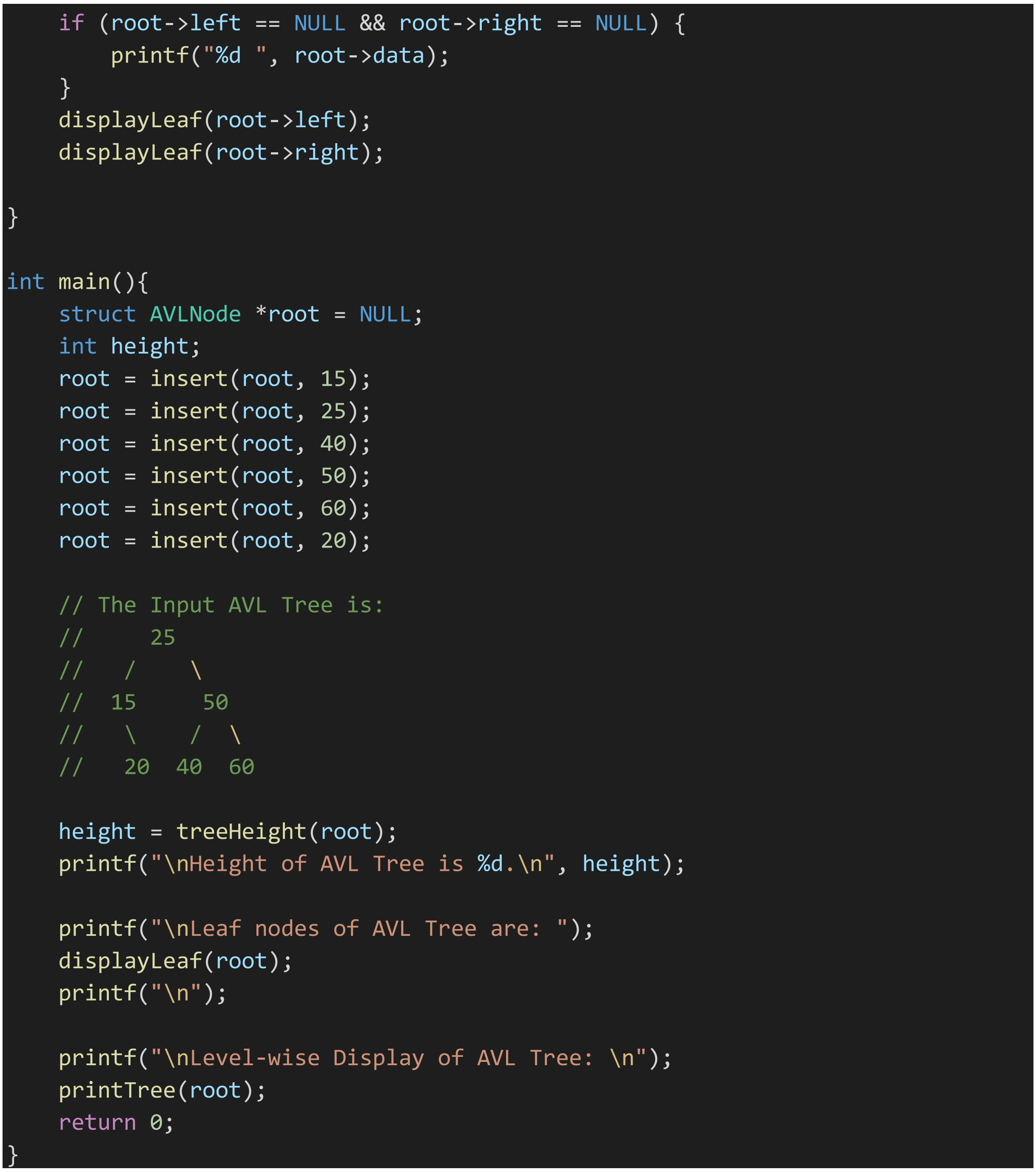
```
#include <stdio.h>
#include <stdlib.h>

struct AVLNode{
    int data;
    struct AVLNode *left;
    struct AVLNode *right;
    int height;
};

struct AVLNode *createNode(int data){
    struct AVLNode *newNode = (struct AVLNode *)malloc(sizeof(struct
AVLNode));
    newNode->data = data;
    newNode->left = NULL;
    newNode->right = NULL;
    newNode->height = 1;
    return newNode;
}

int getHeight(struct AVLNode *node){
    if (node == NULL){
        return 0;
    }
    return node->height;
}

int max(int a, int b){
    return (a > b) ? a : b;
}

int getBalance(struct AVLNode *node){
    if (node == NULL){
```



```
        return 0;
    }
    return getHeight(node->left) - getHeight(node->right);
}

struct AVLNode *rightRotate(struct AVLNode *y){
    struct AVLNode *x = y->left;
    struct AVLNode *T2 = x->right;
    x->right = y;
    y->left = T2;
    y->height = max(getHeight(y->left), getHeight(y->right)) + 1;
    x->height = max(getHeight(x->left), getHeight(x->right)) + 1;
    return x;
}

struct AVLNode *leftRotate(struct AVLNode *x){
    struct AVLNode *y = x->right;
    struct AVLNode *T2 = y->left;
    y->left = x;
    x->right = T2;
    x->height = max(getHeight(x->left), getHeight(x->right)) + 1;
    y->height = max(getHeight(y->left), getHeight(y->right)) + 1;
    return y;
}

struct AVLNode *insert(struct AVLNode *root, int data){
    if (root == NULL){
        return createNode(data);
    }
    if (data < root->data){
        root->left = insert(root->left, data);
    }
    else if (data > root->data){
        root->right = insert(root->right, data);
    }
    else{
        return root;
    }
    root->height = max(getHeight(root->left), getHeight(root->right)) + 1;
    int balance = getBalance(root);
    if (balance > 1 && data < root->left->data){
        return rightRotate(root);
    }
    if (balance < -1 && data > root->right->data){
        return leftRotate(root);
    }
    if (balance > 1 && data > root->left->data){
        root->left = leftRotate(root->left);
```

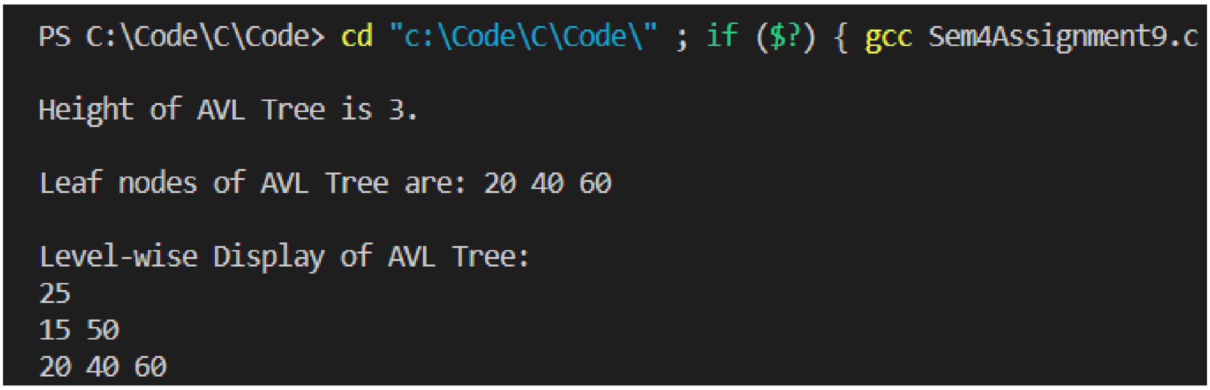


```
        return rightRotate(root);
    }
    if (balance < -1 && data < root->right->data){
        root->right = rightRotate(root->right);
        return leftRotate(root);
    }
    return root;
}

void printLevel(struct AVLNode *root, int level){
    if (root == NULL){
        return;
    }
    if (level == 1){
        printf("%d ", root->data);
    }
    else if (level > 1){
        printLevel(root->left, level - 1);
        printLevel(root->right, level - 1);
    }
}

int treeHeight(struct AVLNode *root){
    if (root == NULL){
        return 0;
    }
    int leftHeight = treeHeight(root->left);
    int rightHeight = treeHeight(root->right);
    if (leftHeight > rightHeight){
        return leftHeight + 1;
    }
    else{
        return rightHeight + 1;
    }
}

void printTree(struct AVLNode *root){
    int height = treeHeight(root);
    for (int i = 1; i <= height; i++){
        printLevel(root, i);
        printf("\n");
    }
}

void displayLeaf(struct AVLNode *root) {
    if (root == NULL) {
        return;
    }
```



```c
    if (root->left == NULL && root->right == NULL) {
        printf("%d ", root->data);
    }
    displayLeaf(root->left);
    displayLeaf(root->right);

}

int main(){
    struct AVLNode *root = NULL;
    int height;
    root = insert(root, 15);
    root = insert(root, 25);
    root = insert(root, 40);
    root = insert(root, 50);
    root = insert(root, 60);
    root = insert(root, 20);

    // The Input AVL Tree is:
    //      25
    //   /    \
    //  15     50
    //   \    /  \
    //   20  40  60

    height = treeHeight(root);
    printf("\nHeight of AVL Tree is %d.\n", height);

    printf("\nLeaf nodes of AVL Tree are: ");
    displayLeaf(root);
    printf("\n");

    printf("\nLevel-wise Display of AVL Tree: \n");
    printTree(root);
    return 0;
}
```

Output:



```
PS C:\Code\C\Code> cd "c:\Code\C\Code\" ; if ($?) { gcc Sem4Assignment9.c

Height of AVL Tree is 3.

Leaf nodes of AVL Tree are: 20 40 60

Level-wise Display of AVL Tree:
25
15 50
20 40 60
```