Tugas pendahuluan modul 15

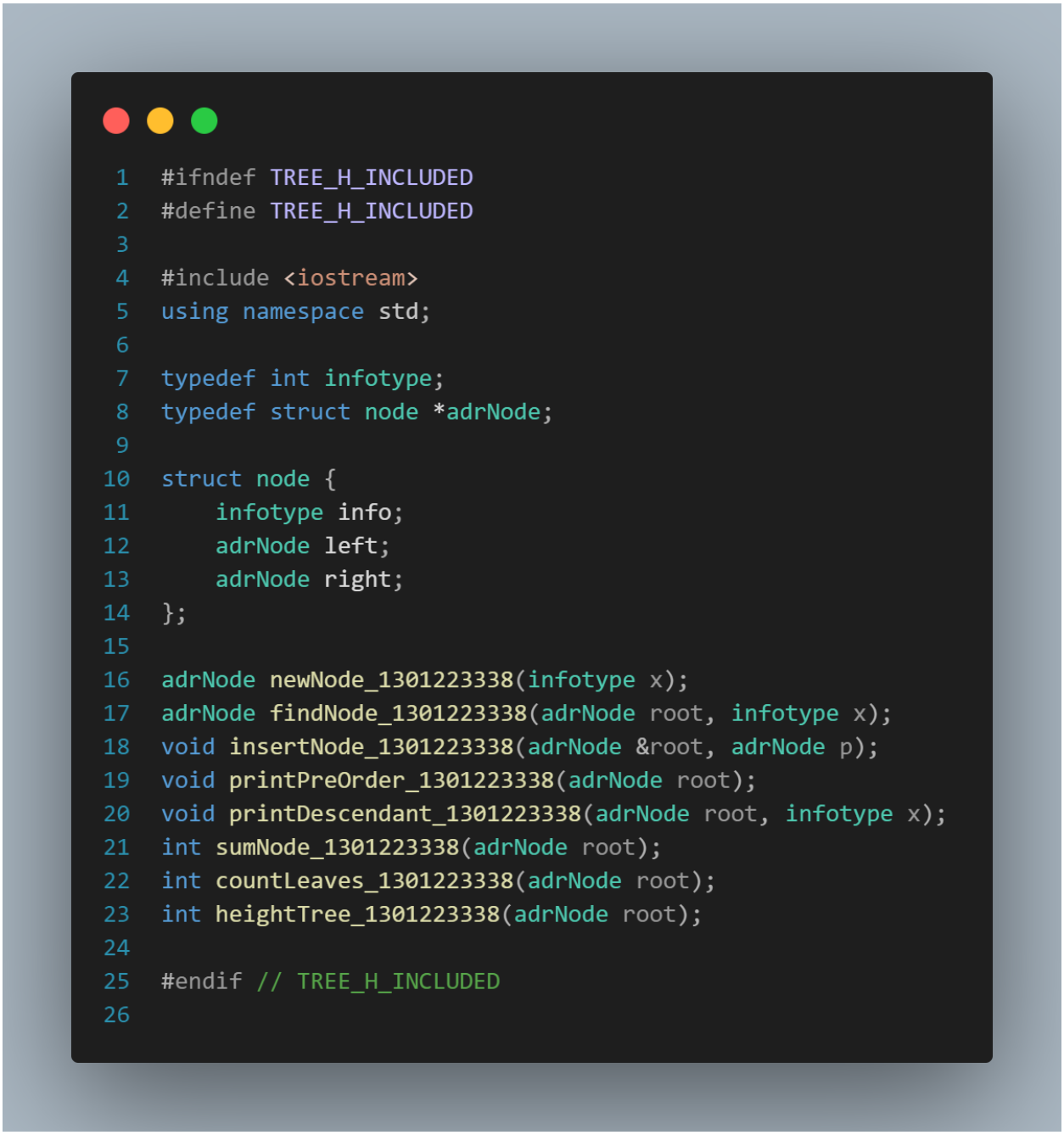
Nama : Helmi Efendi Lubis

NIM : 1301223338

Kelas : IF-46-08

Link Github : https://github.com/helmiel/StrukturDataTP/tree/main/TP15

Header



```h
#ifndef TREE_H_INCLUDED
#define TREE_H_INCLUDED

#include <iostream>
using namespace std;

typedef int infotype;
typedef struct node *adrNode;

struct node {
    infotype info;
    adrNode left;
    adrNode right;
};

adrNode newNode_1301223338(infotype x);
adrNode findNode_1301223338(adrNode root, infotype x);
void insertNode_1301223338(adrNode &root, adrNode p);
void printPreOrder_1301223338(adrNode root);
void printDescendant_1301223338(adrNode root, infotype x);
int sumNode_1301223338(adrNode root);
int countLeaves_1301223338(adrNode root);
int heightTree_1301223338(adrNode root);

#endif // TREE_H_INCLUDED

```

Source

```cpp
#include "Tree.h"

adrNode newNode_1301223338(infotype x){
    adrNode p = new node;
    p->info = x;
    p->left = NULL;
    p->right = NULL;
    return p;
}

adrNode findNode_1301223338(adrNode root, infotype x){
    if (root == NULL || root->info == x) {
        return root;
    }
    if (x < root->info) {
        return findNode_1301223338(root->left, x);
    }
    else {
        return findNode_1301223338(root->right, x);
    }
}

void insertNode_1301223338(adrNode &root, adrNode p){
    if(root == NULL){
        root = p;
    }else {
        if(p->info < root->info){
            insertNode_1301223338(root->left, p);
        }else {
            insertNode_1301223338(root->right, p);
        }
    }
}

void printPreOrder_1301223338(adrNode root){
    if(root != NULL){
        cout << root->info << " ";
        printPreOrder_1301223338(root->left);
        printPreOrder_1301223338(root->right);
    }

}

void printDescendant_1301223338(adrNode root, infotype x){
    if (root != NULL) {
        if (root->info == x) {
            printPreOrder_1301223338(root->left);
            printPreOrder_1301223338(root->right);
        } else if (x < root->info) {
            printDescendant_1301223338(root->left, x);
        } else {
            printDescendant_1301223338(root->right, x);
        }
    }
}

int sumNode_1301223338(adrNode root){
    if(root == NULL){
        return 0;
    }else {
        return root->info + sumNode_1301223338(root->left) + sumNode_1301223338(root->right);
    }
}

int countLeaves_1301223338(adrNode root){
    if(root == NULL){
        return 0;
    }else if (root->left == NULL && root->right == NULL){
        return 1;
    }else {
        return countLeaves_1301223338(root->left) + countLeaves_1301223338(root->right);
    }
}

int heightTree_1301223338(adrNode root){
    if(root == NULL){
        return 0;
    }else {
        int leftHeight = heightTree_1301223338(root->left);
        int rightHeight = heightTree_1301223338(root->right);
        if (leftHeight >= rightHeight){
            return leftHeight + 1;
        }else{
            return rightHeight + 1;
        }
    }
}

```

Main

```cpp
#include <iostream>
#include "Tree.h"

using namespace std;

int main()
{
    adrNode root = NULL;
    adrNode p;
    int x[9] = {5,3,9,10,4,7,1,8,6};
    int length = sizeof(x) / sizeof(x[0]);
    cout << "================================================================================" << endl;
    for(int i = 0; i < length; i++){
        cout << x[i] << " ";
        p = newNode_1301223338(x[i]);
        insertNode_1301223338(root,p);
    }
    cout << "\n\nPre Order\t: "; printPreOrder_1301223338(root); cout << endl;
    cout << "\nDescendent of Node 9\t: "; printDescendant_1301223338(root, 9); cout << endl;
    cout << "\nSum of BST info: " << sumNode_1301223338(root) << endl;
    cout << "Number of Leaves: " << countLeaves_1301223338(root) << endl;
    cout << "Height of tree: " << heightTree_1301223338(root) << endl;
    cout << "================================================================================" << endl;
    return 0;
}

```

Output

```
================================================================================
5 3 9 10 4 7 1 8 6

Pre Order       : 5 3 1 4 9 7 6 8 10

Descendent of Node 9    : 7 6 8 10

Sum of BST info: 53
Number of Leaves: 5
Height of tree: 4
================================================================================

Process returned 0 (0x0)   execution time : 0.272 s
Press any key to continue.
```