Duta Razaq Suhoyo

1301210280

IF 45 11

Tree.h

Tree.h | Tree.cpp | main.cpp

```
#ifndef TREE_H_INCLUDED
#define TREE_H_INCLUDED

#include <iostream>
using namespace std;

#define info(p) (p)->info
#define right(p) (p)->right
#define left(p) (p)->left
#define nil NULL

typedef int infotype;
typedef struct node *adrnode;

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

adrnode newnode_1301210280(infotype x);
adrnode findnode_1301210280(adrnode root, infotype x);
void insertnode_1301210280(adrnode &root, adrnode p);
void printpreorder_1301210280(adrnode root);
void printdescendant_1301210280(adrnode root, infotype x);
int sumnode_1301210280(adrnode root);
int countleaves_1301210280(adrnode root);
int heightTree_1301210280(adrnode root);

#endif // TREE_H_INCLUDED

```

Tree.cpp

Tree.h | Tree.cpp | main.cpp

```cpp
#include "Tree.h"

adrnode newnode_1301210280(infotype x){
    adrnode p = new node;
    info(p) = x;
    right(p) = nil;
    left(p) = nil;
}

adrnode findnode_1301210280(adrnode root, infotype x){
    if(root == nil){
        return nil;
    }else if(info(root) == x){
        return root;
    }else if(info(root) > x){
        return findnode_1301210280(left(root), x);
    }else{
        return findnode_1301210280(right(root), x);
    }
}

void insertnode_1301210280(adrnode &root, adrnode p){
    if(root == nil){
        root = p;
    }else if(info(p) < info(root)){
        insertnode_1301210280(left(root), p);
    }else{
        insertnode_1301210280(right(root), p);
    }
}

void printpreorder_1301210280(adrnode root){
    if(root != nil){
        cout << info(root) << " ";
        printpreorder_1301210280(left(root));
        printpreorder_1301210280(right(root));
    }
}
```

Tree.h | Tree.cpp | main.cpp

```cpp

void printdescendant_1301210280(adrnode root, infotype x){
    if(root != nil){
        printdescendant_1301210280(left(root), x);
        if(info(root) != x){
            cout << info(root) << " ";
        }
        printdescendant_1301210280(right(root), x);
    }
}

int sumnode_1301210280(adrnode root){
    if(root == nil){
        return 0;
    }else{
        return info(root) + sumnode_1301210280(left(root)) + sumnode_1301210280(right(root));
    }
}

int countleaves_1301210280(adrnode root){
    if(root == nil){
        return 0;
    }else if(left(root) == nil && right(root) == nil){
        return 1;
    }else{
        return countleaves_1301210280(left(root)) + countleaves_1301210280(right(root));
    }
}

int heightTree_1301210280(adrnode root){
    if(root == nil){
        return -1;
    }else{
        return max(heightTree_1301210280(left(root))+1, heightTree_1301210280(right(root))+1);
    }
}

```

Main.cpp

Tree.h | Tree.cpp | main.cpp

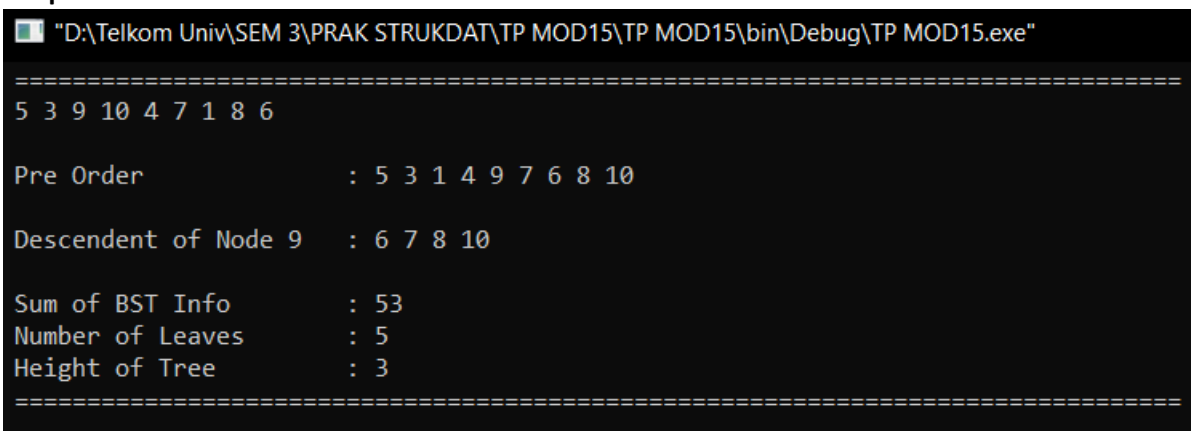
```cpp
#include <iostream>
#include "Tree.h"

using namespace std;

int main()
{
    cout<<" ================================================================================ "<<endl;
    int x[9] = {5,3,9,10,4,7,1,8,6};
    int i = 0;

    while(i<9){
        cout<<" "<<x[i];
        i++;
    }

    i = 0;
    adrnode p;
    adrnode root = nil;

    while(i<9){
        p = newnode_1301210280(x[i]);
        insertnode_1301210280(root, p);
        i++;
    }

    printf(" \n ");
    printf(" \n Pre Order\t\t: ");
    printpreorder_1301210280(root);

    printf(" \n ");
    printf(" \n Descendent of Node 9\t: ");
    p = findnode_1301210280(root, 9);
    printdescendant_1301210280(p,9);

    printf(" \n ");
    printf(" \n Sum of BST Info\t: ");
```

Tree.h | Tree.cpp | main.cpp

```cpp
    while(i<9){
        cout<<" "<<x[i];
        i++;
    }

    i = 0;
    adrnode p;
    adrnode root = nil;

    while(i<9){
        p = newnode_1301210280(x[i]);
        insertnode_1301210280(root, p);
        i++;
    }

    printf(" \n ");
    printf(" \n Pre Order\t\t: ");
    printpreorder_1301210280(root);

    printf(" \n ");
    printf(" \n Descendent of Node 9\t: ");
    p = findnode_1301210280(root, 9);
    printdescendant_1301210280(p,9);

    printf(" \n ");
    printf(" \n Sum of BST Info\t: ");
    cout<<sumnode_1301210280(root);

    printf(" \n Number of Leaves\t: ");
    cout<<countleaves_1301210280(root);

    printf(" \n Height of Tree\t\t: ");
    cout<<heightTree_1301210280(root)<<endl;

    cout<<" ================================================================================ "<<endl;
}
```

D:\Telkom Univ\SEM 3\PRAK STRUKDAT\TP MOD15\TP MOD15\main.cpp | C/C++ | Windows (CR+LF) | WINDO

Output


"D:\Telkom Univ\SEM 3\PRAK STRUKDAT\TP MOD15\TP MOD15\bin\Debug\TP MOD15.exe"
5 3 9 10 4 7 1 8 6
Pre Order : 5 3 1 4 9 7 6 8 10
Descendent of Node 9 : 6 7 8 10
Sum of BST Info : 53
Number of Leaves : 5
Height of Tree : 3