Deore Mufrad Hendrady
1301223029

Header.h

Source.cpp

header.h | main.cpp | source.cpp

source.cpp > countLeaves(adrNode)

```cpp
#include "header.h"
adrNode newNode_1301223029(infotype x){
    adrNode p;
    p = new node;
    p->info= x;
    p->kanan = NULL;
    p->kiri = NULL;
    return p;
}
adrNode findNode_1301223029(adrNode root, infotype x){
    if (root == NULL || root ->info == x){
        return root;
    }

    if (x > root->info) {
        return findNode_1301223029(root->kanan,x);
    }else {
        return findNode_1301223029(root->kiri,x);
    }
}
void insertNode_1301223029(adrNode &root, adrNode p){
    if(root == NULL ){
        root = p;
    }else {
        if (p ->info > root->info) {
            insertNode_1301223029(root->kanan, p);
        }else {
            insertNode_1301223029(root->kiri, p);
        }
    }
}
void printPreOrder_1301223029(adrNode root){
    if (root == NULL){
        return;
    }

    cout << root->info << " " ;
    printPreOrder_1301223029(root->kiri );
    printPreOrder_1301223029(root->kanan);

}
void printDescendant_1301223029(adrNode root, infotype x){
    adrNode p;
    p = findNode_1301223029(root , x);
    printPreOrder_1301223029(p);

}
int sumNode_1301223029(adrNode root){
```

adrNode root


source.cpp
source.cpp > countLeaves(adrNode)
}
void printDescendant_1301223029(adrNode root, infotype x){
adrNode p;
p = findNode_1301223029(root , x);
printPreOrder_1301223029(p);
}
int sumNode_1301223029(adrNode root){
int hasil = 0;
if (root == NULL){
return 0;
}
return root->info + sumNode_1301223029(root->kiri) + sumNode_1301223029(root->ka
}
int countLeaves(adrNode root){
if (root == NULL){
return 0;
}
if (root->kiri == NULL && root->kanan == NULL){
return 1;
}else {
return countLeaves(root->kiri) + countLeaves(root->kanan);
}
}
int heightTree_1301223029(adrNode root){
if (root == NULL){
return 0;
}
return max(heightTree_1301223029(root->kiri), heightTree_1301223029(root->kanan)

main.cpp

header.h | main.cpp | source.cpp

main.cpp > main()

```cpp
#include "header.h"


int main(){
    int x[9] = {5,3,9,10,4,7,1,8,6};

    adrNode root;
    root = NULL;

    for (int i = 0 ; i<= 9 -1 ; i++){
        cout << x[i] << " ";
        insertNode_1301223029(root, newNode_1301223029(x[i]));
    }

    cout << endl;

    cout << "Preorder : ";
    printPreOrder_1301223029(root);
    cout << endl;

    cout << "descendent of Node 9 : ";
    printDescendant_1301223029(root,9);
    cout << endl;

    cout << "Sum of BST Info : " << sumNode_1301223029(root)<<endl;

    cout << "Number of Leaves : " << countLeaves(root)<<endl;

    cout << "Height of Tree : " << heightTree_1301223029(root) << endl;
    return 0;
}
```

output

```
~/STD-TP/TP-15 on main !1 ?1
./myprogram
5 3 9 10 4 7 1 8 6
Preorder : 5 3 1 4 9 7 6 8 10
descendent of Node 9 : 9 7 6 8 10
Sum of BST Info : 53
Number of Leaves : 5
Height of Tree : 3
```