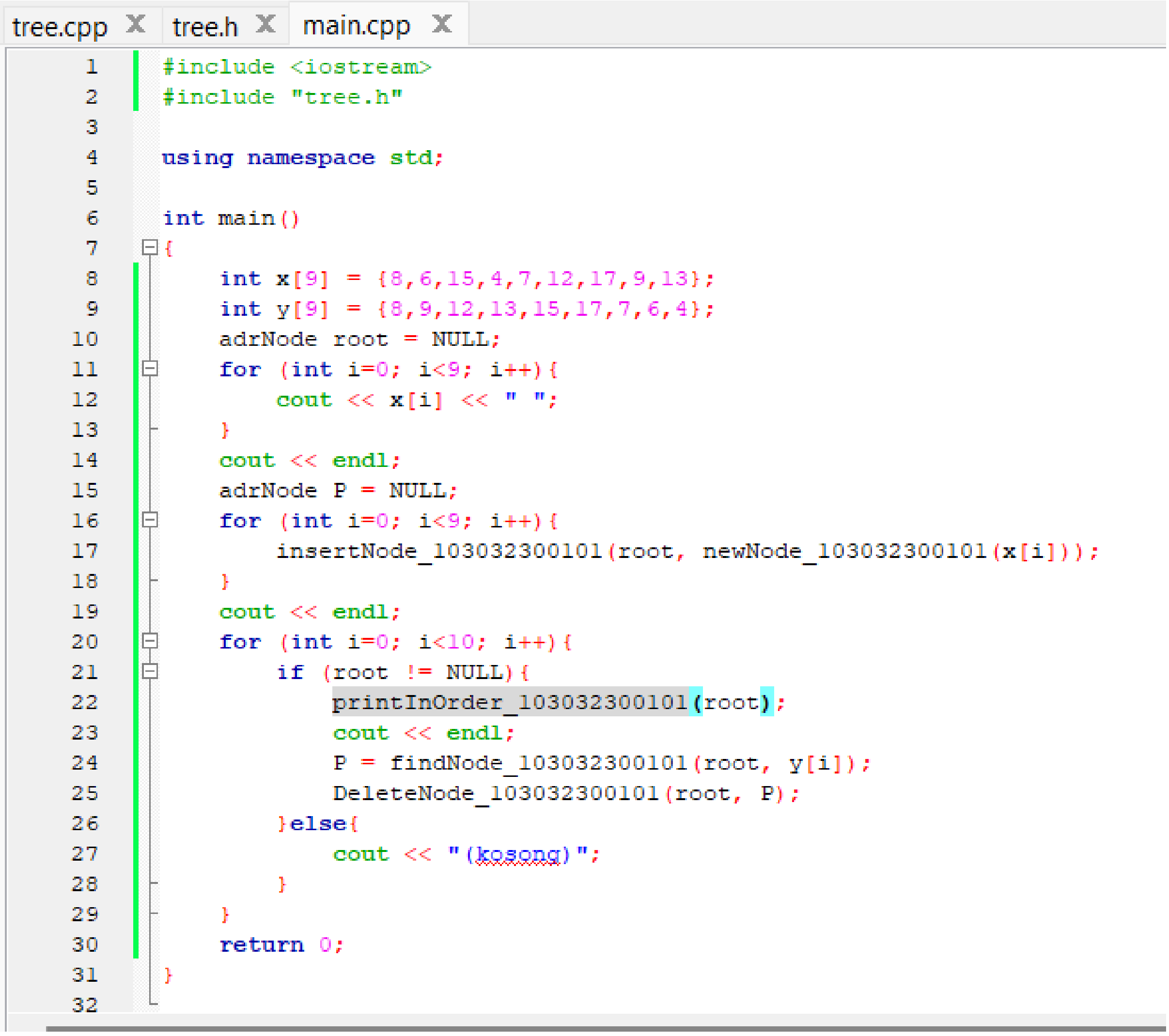
Made Naradeon Handika Pramesta
103032300101

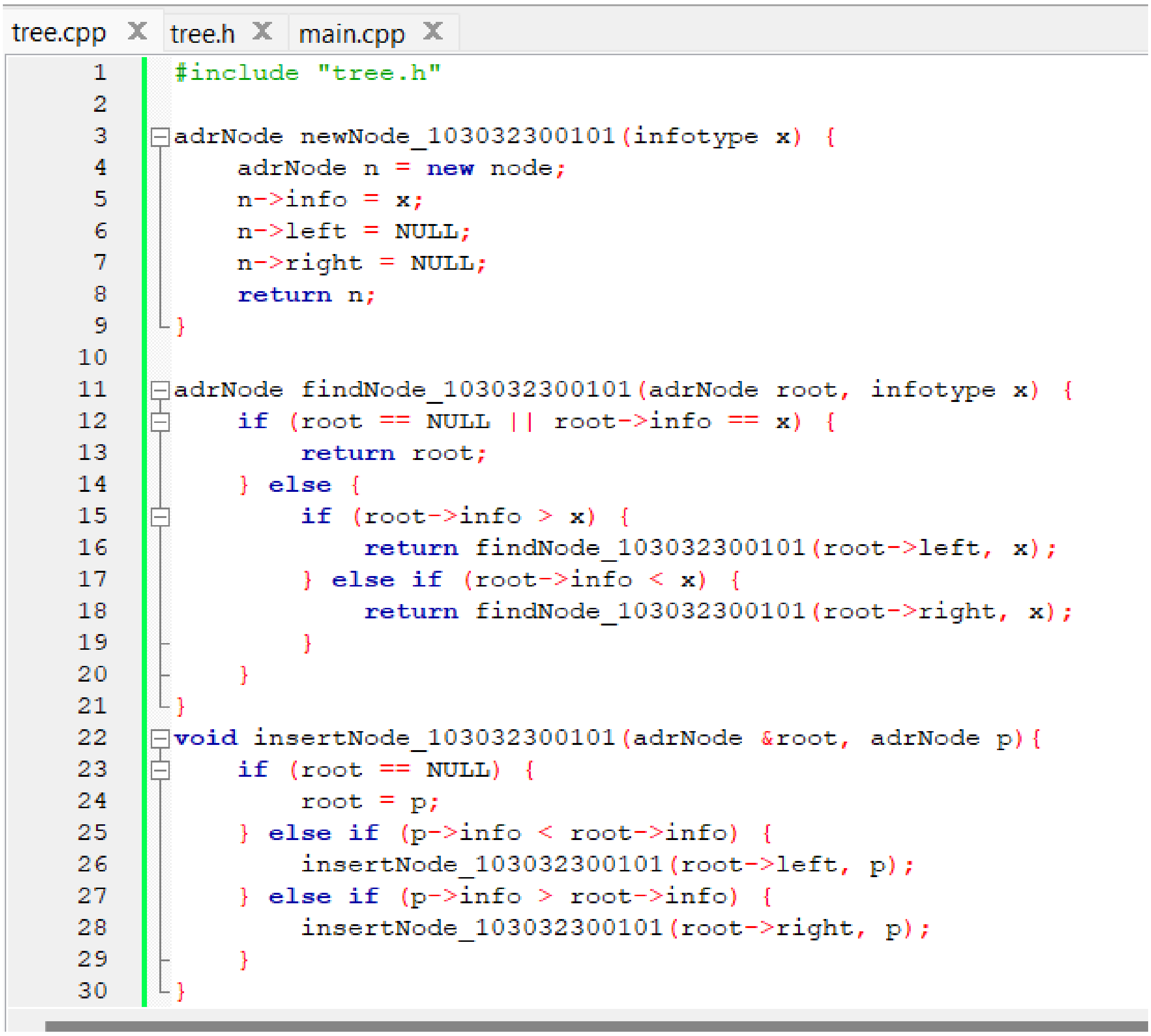
Main.cpp



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

using namespace std;

int main()
{
    int x[9] = {8,6,15,4,7,12,17,9,13};
    int y[9] = {8,9,12,13,15,17,7,6,4};
    adrNode root = NULL;
    for (int i=0; i<9; i++){
        cout << x[i] << " ";
    }
    cout << endl;
    adrNode P = NULL;
    for (int i=0; i<9; i++){
        insertNode_103032300101(root, newNode_103032300101(x[i]));
    }
    cout << endl;
    for (int i=0; i<10; i++){
        if (root != NULL){
            printInOrder_103032300101(root);
            cout << endl;
            P = findNode_103032300101(root, y[i]);
            DeleteNode_103032300101(root, P);
        }else{
            cout << "(kosong)";
        }
    }
    return 0;
}

```

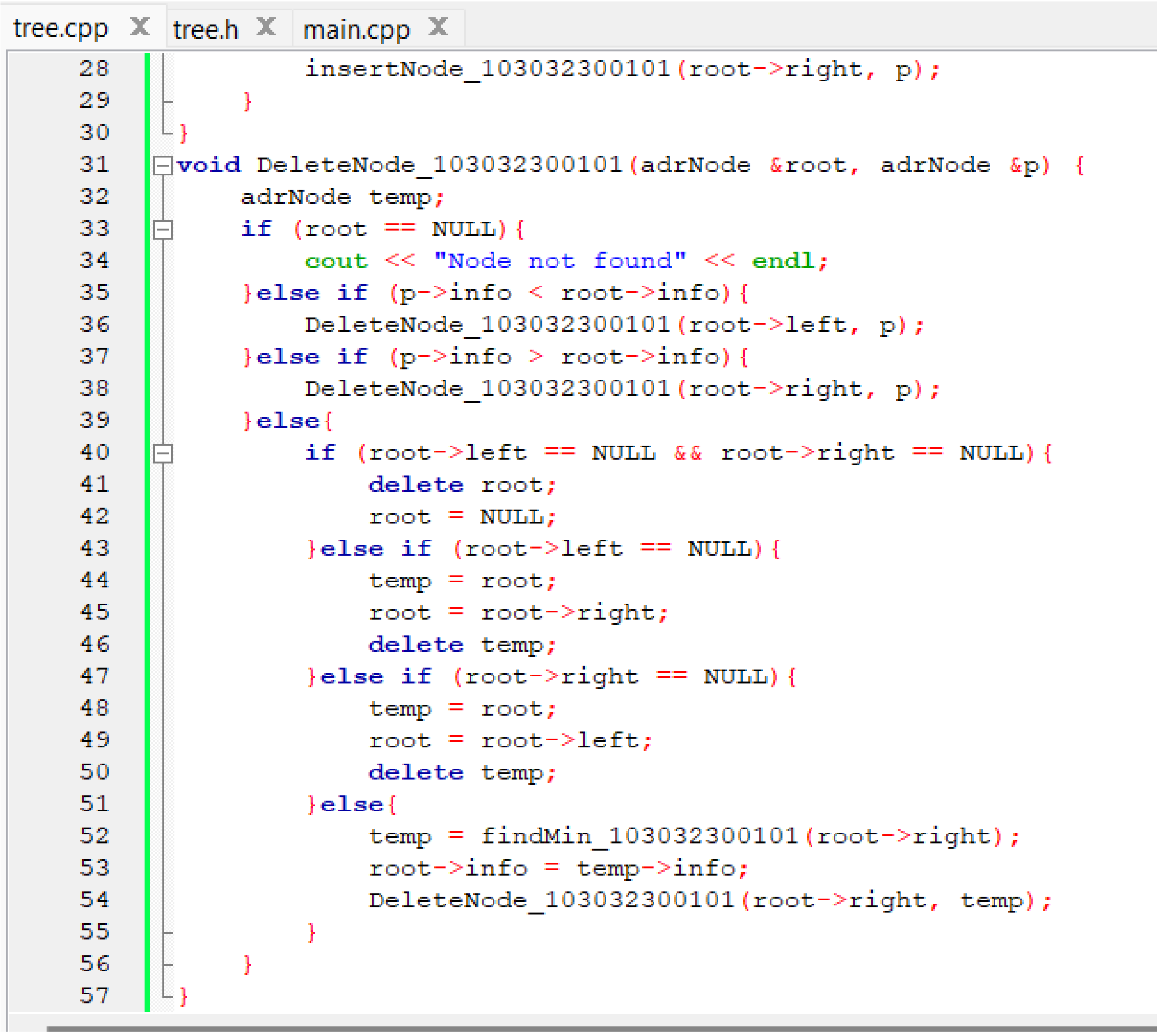
Tree.cpp

Made Naradeon Handika Pramesta
103032300101

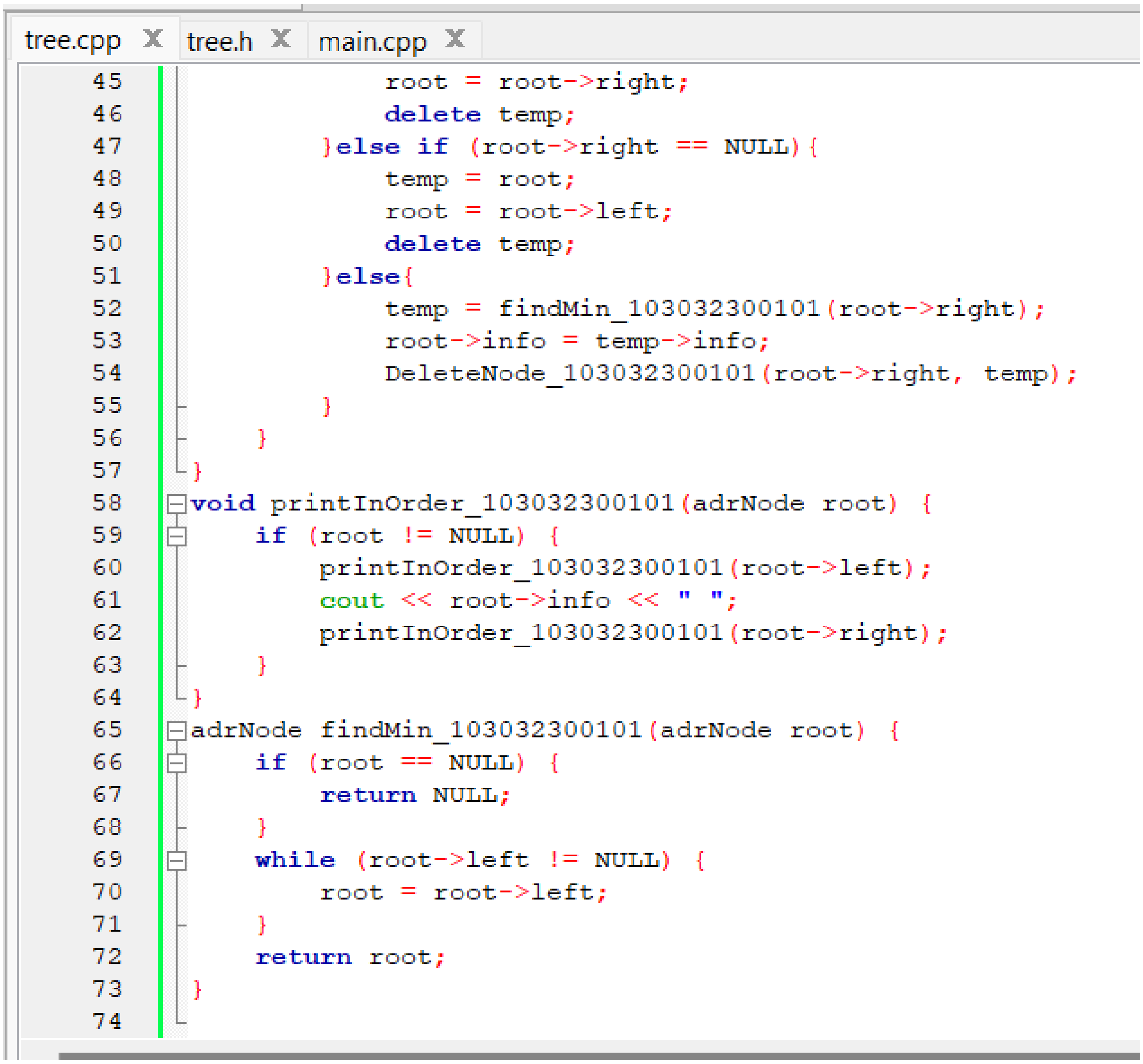


```
#include "tree.h"

adrNode newNode_103032300101(infotype x) {
    adrNode n = new node;
    n->info = x;
    n->left = NULL;
    n->right = NULL;
    return n;
}

adrNode findNode_103032300101(adrNode root, infotype x) {
    if (root == NULL || root->info == x) {
        return root;
    } else {
        if (root->info > x) {
            return findNode_103032300101(root->left, x);
        } else if (root->info < x) {
            return findNode_103032300101(root->right, x);
        }
    }
}
void insertNode_103032300101(adrNode &root, adrNode p){
    if (root == NULL) {
        root = p;
    } else if (p->info < root->info) {
        insertNode_103032300101(root->left, p);
    } else if (p->info > root->info) {
        insertNode_103032300101(root->right, p);
    }
}
```

Made Naradeon Handika Pramesta
103032300101



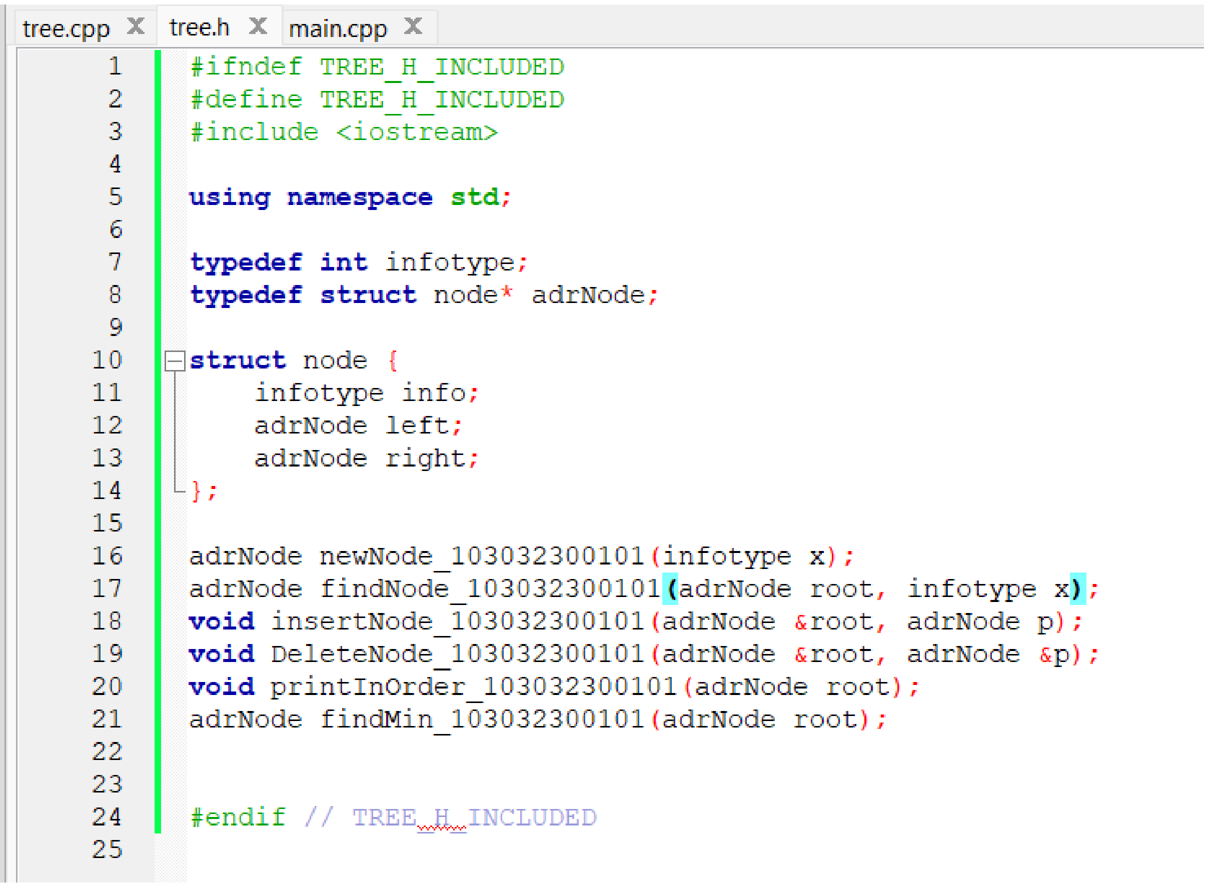
```
        insertNode_103032300101(root->right, p);
    }
}
void DeleteNode_103032300101(adrNode &root, adrNode &p) {
    adrNode temp;
    if (root == NULL){
        cout << "Node not found" << endl;
    }else if (p->info < root->info){
        DeleteNode_103032300101(root->left, p);
    }else if (p->info > root->info){
        DeleteNode_103032300101(root->right, p);
    }else{
        if (root->left == NULL && root->right == NULL){
            delete root;
            root = NULL;
        }else if (root->left == NULL){
            temp = root;
            root = root->right;
            delete temp;
        }else if (root->right == NULL){
            temp = root;
            root = root->left;
            delete temp;
        }else{
            temp = findMin_103032300101(root->right);
            root->info = temp->info;
            DeleteNode_103032300101(root->right, temp);
        }
    }
}
```



```
            root = root->right;
            delete temp;
        }else if (root->right == NULL){
            temp = root;
            root = root->left;
            delete temp;
        }else{
            temp = findMin_103032300101(root->right);
            root->info = temp->info;
            DeleteNode_103032300101(root->right, temp);
        }
    }
}
void printInOrder_103032300101(adrNode root) {
    if (root != NULL) {
        printInOrder_103032300101(root->left);
        cout << root->info << " ";
        printInOrder_103032300101(root->right);
    }
}
adrNode findMin_103032300101(adrNode root) {
    if (root == NULL) {
        return NULL;
    }
    while (root->left != NULL) {
        root = root->left;
    }
    return root;
}
```

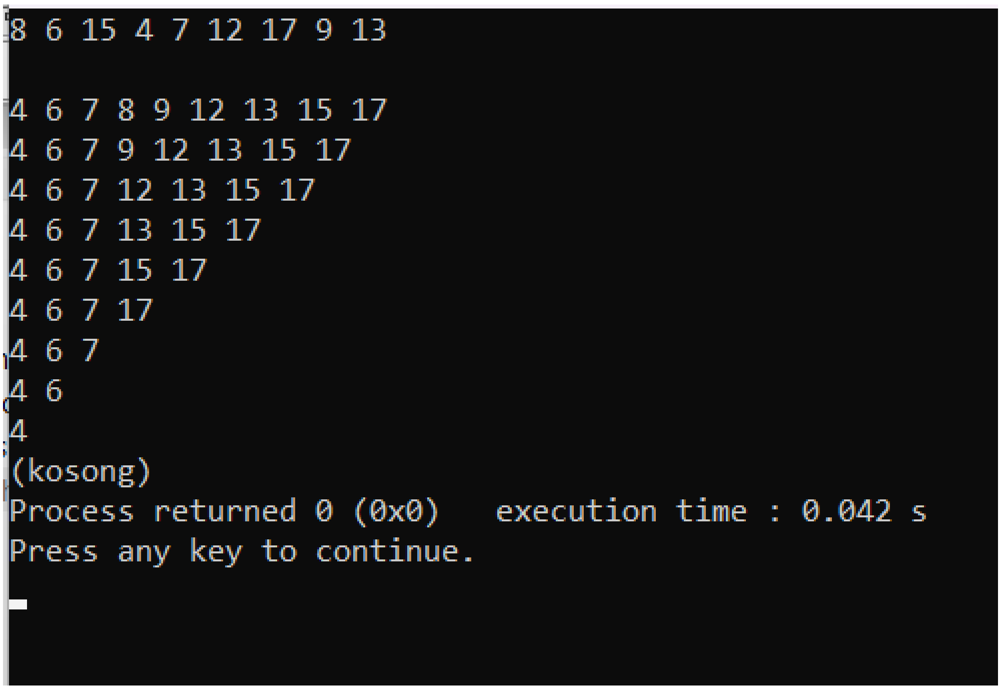
Tree.h

Made Naradeon Handika Pramesta
103032300101



```
#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_103032300101(infotype x);
adrNode findNode_103032300101(adrNode root, infotype x);
void insertNode_103032300101(adrNode &root, adrNode p);
void DeleteNode_103032300101(adrNode &root, adrNode &p);
void printInOrder_103032300101(adrNode root);
adrNode findMin_103032300101(adrNode root);


#endif // TREE_H_INCLUDED
```

Hasil :



```
8 6 15 4 7 12 17 9 13

4 6 7 8 9 12 13 15 17
4 6 7 9 12 13 15 17
4 6 7 12 13 15 17
4 6 7 13 15 17
4 6 7 15 17
4 6 7 17
4 6 7
4 6
4
(kosong)
Process returned 0 (0x0)   execution time : 0.042 s
Press any key to continue.
```