Main.cpp

main.cpp | Tree.cpp | Tree.h

```cpp
#include "Tree.h"
int main()
{
    int x[9] = {8,6,15,4,7,12,17,9,13};
    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_103032330054(root, newNode_103032330054(x[i]));
    }
    printInOrder_103032330054(root);
    cout << endl;
    cout << mostLeftNode_103032330054(root)->info << endl;
    cout << mostRightNode_103032330054(root)->info << endl;
    return 0;
}
```

Tree.cpp

main.cpp | Tree.cpp | Tree.h

```cpp
#include "Tree.h"
adrNode newNode_103032330054(infotype x){
    adrNode P = new elm;
    P->left = NULL;
    P->right = NULL;
    P->info = x;
    return P;
}
void insertNode_103032330054(adrNode &root, adrNode p)
    if (root == NULL){
        root = p;
    }else{
        if (p->info > root->info){
            insertNode_103032330054(root->right, p);
        }else{
            insertNode_103032330054(root->left, p);
        }
    }
}
adrNode findMin_103032330054(adrNode root){
    if (root->left == NULL){
        return root;
    }else{
        return findMin_103032330054(root->left);
    }
}
```

```
27  void DeleteNode_103032330054(adrNode &root, adrNode &p){
28      adrNode temp;
29      if (root == NULL){
30          cout << "Node not found" << endl;
31      }else if (p->info < root->info){
32          DeleteNode_103032330054(root->left, p);
33      }else if (p->info > root->info){
34          DeleteNode_103032330054(root->right, p);
35      }else{
36          if (root->left == NULL && root->right == NULL){
37              delete root;
38              root = NULL;
39          }else if (root->left == NULL){
40              temp = root;
41              root = root->right;
42              delete temp;
43          }else if (root->right == NULL){
44              temp = root;
45              root = root->left;
46              delete temp;
47          }else{
48              temp = findMin_103032330054(root->right);
49              root->info = temp->info;
50              DeleteNode_103032330054(root->right, temp);
51          }
52      }
53  }
54  void printInOrder_103032330054(adrNode root){
55      if (root != NULL){
56          printInOrder_103032330054(root->left);
57          cout << root->info << " ";
58          printInOrder_103032330054(root->right);
59      }
60  }
```

```
61  adrNode mostLeftNode_103032330054(adrNode root){
62      if (root->left == NULL){
63          return root;
64      }else{
65          return mostLeftNode_103032330054(root->left);
66      }
67  }
68  adrNode mostRightNode_103032330054(adrNode root){
69      if (root->right == NULL){
70          return root;
71      }else{
72          return mostRightNode_103032330054(root->right);
73      }
74  }
```

Tree/h

main.cpp Tree.cpp Tree.h

```h
#ifndef TREE_H_INCLUDED
#define TREE_H_INCLUDED
#include <iostream>
using namespace std;
typedef int infotype;
typedef struct elm *adrNode;
struct elm{
    adrNode right;
    adrNode left;
    infotype info;
};
adrNode newNode_103032330054(infotype x);
void insertNode_103032330054(adrNode &root, adrNode p);
void DeleteNode_103032330054(adrNode &root, adrNode &p);
void printInOrder_103032330054(adrNode root);
adrNode findMin_103032330054(adrNode root);
adrNode mostLeftNode_103032330054(adrNode root);
adrNode mostRightNode_103032330054(adrNode root);
#endif // TREE_H_INCLUDED

```

Output

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

Process returned 0 (0x0)   execution time : 0.067 s
Press any key to continue.
```