

Main.cpp

```
main.cpp x Tree.cpp x Tree.h x
1  #include "Tree.h"
2  int main()
3  {
4      int x[9] = {8,6,15,4,7,12,17,9,13};
5      int y[9] = {8,9,12,13,15,17,7,6,4};
6      adrNode root = NULL;
7      for (int i=0; i<9; i++){
8          cout << x[i] << " ";
9      }
10     cout << endl;
11     adrNode P = NULL;
12     for (int i=0; i<9; i++){
13         insertNode_103032330054(root, newNode_103032330054(x[i]));
14     }
15     cout << endl;
16     for (int i=0; i<10; i++){
17         if (root != NULL){
18             printInOrder_103032330054(root);
19             cout << endl;
20             P = newNode_103032330054(y[i]);
21             DeleteNode_103032330054(root, P);
22         }else{
23             cout << "(kosong)";
24         }
25     }
26     return 0;
27 }
28
```

Tree.h

```
main.cpp x Tree.cpp x Tree.h x
1  #ifndef TREE_H_INCLUDED
2  #define TREE_H_INCLUDED
3  #include <iostream>
4  using namespace std;
5  typedef int infotype;
6  typedef struct elm *adrNode;
7  struct elm{
8      adrNode right;
9      adrNode left;
10     infotype info;
11 };
12 adrNode newNode_103032330054(infotype x);
13 void insertNode_103032330054(adrNode &root, adrNode p);
14 void DeleteNode_103032330054(adrNode &root, adrNode &p);
15 void printInOrder_103032330054(adrNode root);
16 adrNode findMin_103032330054(adrNode root);
17 adrNode findNode_103032330054(adrNode root, infotype x);
18
19 #endif // TREE_H_INCLUDED
20
```

Tree.cpp

```
main.cpp x Tree.cpp x Tree.h x
1  #include "Tree.h"
2  adrNode newNode_103032330054(infotype x){
3      adrNode P = new elm;
4      P->left = NULL;
5      P->right = NULL;
6      P->info = x;
7      return P;
8  }
9  void insertNode_103032330054(adrNode &root, adrNode p){
10     if (root == NULL){
11         root = p;
12     }else{
13         if (p->info > root->info){
14             insertNode_103032330054(root->right, p);
15         }else{
16             insertNode_103032330054(root->left, p);
17         }
18     }
19 }
20 adrNode findMin_103032330054(adrNode root){
21     if (root->left == NULL){
22         return root;
23     }else{
24         return findMin_103032330054(root->left);
25     }
26 }
```

```
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 }
```

Output

```
"D:\Kuliah\Semester 3\ST
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)
```