

Made Naradeon Handika Pramesta
103032300101

Main.cpp

```
tree.cpp X tree.h X main.cpp X
1  #include <iostream>
2  #include "tree.h"
3
4  using namespace std;
5
6  int main()
7  {
8      int x[9] = {8,6,15,4,7,12,17,9,13};
9      int y[9] = {8,9,12,13,15,17,7,6,4};
10     adrNode root = NULL;
11     for (int i=0; i<9; i++){
12         cout << x[i] << " ";
13     }
14     cout << endl;
15     adrNode P = NULL;
16     for (int i=0; i<9; i++){
17         insertNode_103032300101(root, newNode_103032300101(x[i]));
18     }
19     cout << endl;
20     for (int i=0; i<10; i++){
21         if (root != NULL){
22             printInOrder_103032300101(root);
23             cout << endl;
24             P = findNode_103032300101(root, y[i]);
25             DeleteNode_103032300101(root, P);
26         }else{
27             cout << "(kosong)";
28         }
29     }
30     return 0;
31 }
32
```

Tree.cpp

```
tree.cpp X tree.h X main.cpp X
1  #include "tree.h"
2
3  adrNode newNode_103032300101(infotype x) {
4      adrNode n = new node;
5      n->info = x;
6      n->left = NULL;
7      n->right = NULL;
8      return n;
9  }
10
11 adrNode findNode_103032300101(adrNode root, infotype x) {
12     if (root == NULL || root->info == x) {
13         return root;
14     } else {
15         if (root->info > x) {
16             return findNode_103032300101(root->left, x);
17         } else if (root->info < x) {
18             return findNode_103032300101(root->right, x);
19         }
20     }
21 }
22 void insertNode_103032300101(adrNode &root, adrNode p){
23     if (root == NULL) {
24         root = p;
25     } else if (p->info < root->info) {
26         insertNode_103032300101(root->left, p);
27     } else if (p->info > root->info) {
28         insertNode_103032300101(root->right, p);
29     }
30 }
```

```
tree.cpp X tree.h X main.cpp X
28         insertNode_103032300101(root->right, p);
29     }
30 }
31 void DeleteNode_103032300101(adrNode &root, adrNode &p) {
32     adrNode temp;
33     if (root == NULL) {
34         cout << "Node not found" << endl;
35     } else if (p->info < root->info) {
36         DeleteNode_103032300101(root->left, p);
37     } else if (p->info > root->info) {
38         DeleteNode_103032300101(root->right, p);
39     } else {
40         if (root->left == NULL && root->right == NULL) {
41             delete root;
42             root = NULL;
43         } else if (root->left == NULL) {
44             temp = root;
45             root = root->right;
46             delete temp;
47         } else if (root->right == NULL) {
48             temp = root;
49             root = root->left;
50             delete temp;
51         } else {
52             temp = findMin_103032300101(root->right);
53             root->info = temp->info;
54             DeleteNode_103032300101(root->right, temp);
55         }
56     }
57 }
```

```
tree.cpp X tree.h X main.cpp X
45         root = root->right;
46         delete temp;
47     }else if (root->right == NULL){
48         temp = root;
49         root = root->left;
50         delete temp;
51     }else{
52         temp = findMin_103032300101(root->right);
53         root->info = temp->info;
54         DeleteNode_103032300101(root->right, temp);
55     }
56 }
57 }
58 void printInOrder_103032300101(adNode root) {
59     if (root != NULL) {
60         printInOrder_103032300101(root->left);
61         cout << root->info << " ";
62         printInOrder_103032300101(root->right);
63     }
64 }
65 adNode findMin_103032300101(adNode root) {
66     if (root == NULL) {
67         return NULL;
68     }
69     while (root->left != NULL) {
70         root = root->left;
71     }
72     return root;
73 }
74 }
```

Tree.h

Made Naradeon Handika Pramesta
103032300101

```
tree.cpp X tree.h X main.cpp X
1  #ifndef TREE_H_INCLUDED
2  #define TREE_H_INCLUDED
3  #include <iostream>
4
5  using namespace std;
6
7  typedef int infotype;
8  typedef struct node* adrNode;
9
10 struct node {
11     infotype info;
12     adrNode left;
13     adrNode right;
14 };
15
16 adrNode newNode_103032300101(infotype x);
17 adrNode findNode_103032300101(adrNode root, infotype x);
18 void insertNode_103032300101(adrNode &root, adrNode p);
19 void DeleteNode_103032300101(adrNode &root, adrNode &p);
20 void printInOrder_103032300101(adrNode root);
21 adrNode findMin_103032300101(adrNode root);
22
23
24 #endif // TREE_H_INCLUDED
25
```

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.
_
```