#include <stdio.h>

#include <stdlib.h>

typedef struct node {

    int value;

    struct node \*left;

    struct node \*right;

} node;

node \*create\_node(int value) {

    node \*new\_node = (node \*) malloc(sizeof(node));

    new\_node->value = value;

    new\_node->left = NULL;

    new\_node->right = NULL;

    return new\_node;

}

node \*insert(node \*root, int value) {

    if (root == NULL) {

        return create\_node(value);

    }    if (value < root->value) {

        root->left = insert(root->left, value);

    } else if (value > root->value) {

        root->right = insert(root->right, value);

    }    return root;

}

node \*search(node \*root, int value) {

    if (root == NULL || root->value == value) {

        return root;

    } else if (value < root->value) {

        return search(root->left, value);

    } else {

        return search(root->right, value);

    }

}

node \*delete\_node(node \*root, int value) {

    if (root == NULL) {

        return root;

    }    if (value < root->value) {

        root->left = delete\_node(root->left, value);

    } else if (value > root->value) {

        root->right = delete\_node(root->right, value);

    } else {

        if (root->left == NULL) {

            node \*temp = root->right;

            free(root);

            return temp;

        } else if (root->right == NULL) {

            node \*temp = root->left;

            free(root);

            return temp;

        }        node \*temp = root->right;

        while (temp && temp->left != NULL) {

            temp = temp->left;

        }        root->value = temp->value;

        root->right = delete\_node(root->right, temp->value);

    }    return root;

}

void in\_order(node \*root) {

    if (root != NULL) {

        in\_order(root->left);

        printf("%d ", root->value);

        in\_order(root->right);

    }

}

int main() {

    node \*root = NULL;

    root = insert(root, 4);

    insert(root, 2);

    insert(root, 1);

    insert(root, 3);

    insert(root, 6);

    insert(root, 5);

    insert(root, 7);

    printf("In-order traversal: ");

    in\_order(root);

    printf("\n");

    printf("Delete node 5\n");

    root = delete\_node(root, 5);

    printf("In-order traversal: ");

    in\_order(root);

    printf("\n");

    node \*search\_node = search(root, 7);

    if (search\_node == NULL){

        printf("Value not found\n");

    } else {

        printf("Value found: %d\n", search\_node->value);

    }

    search\_node = search(root, 5);

    if (search\_node == NULL){

        printf("Value not found \n");

    } else {

        printf("Value found: %d\n", search\_node->value);

    }

    return 0;

}

