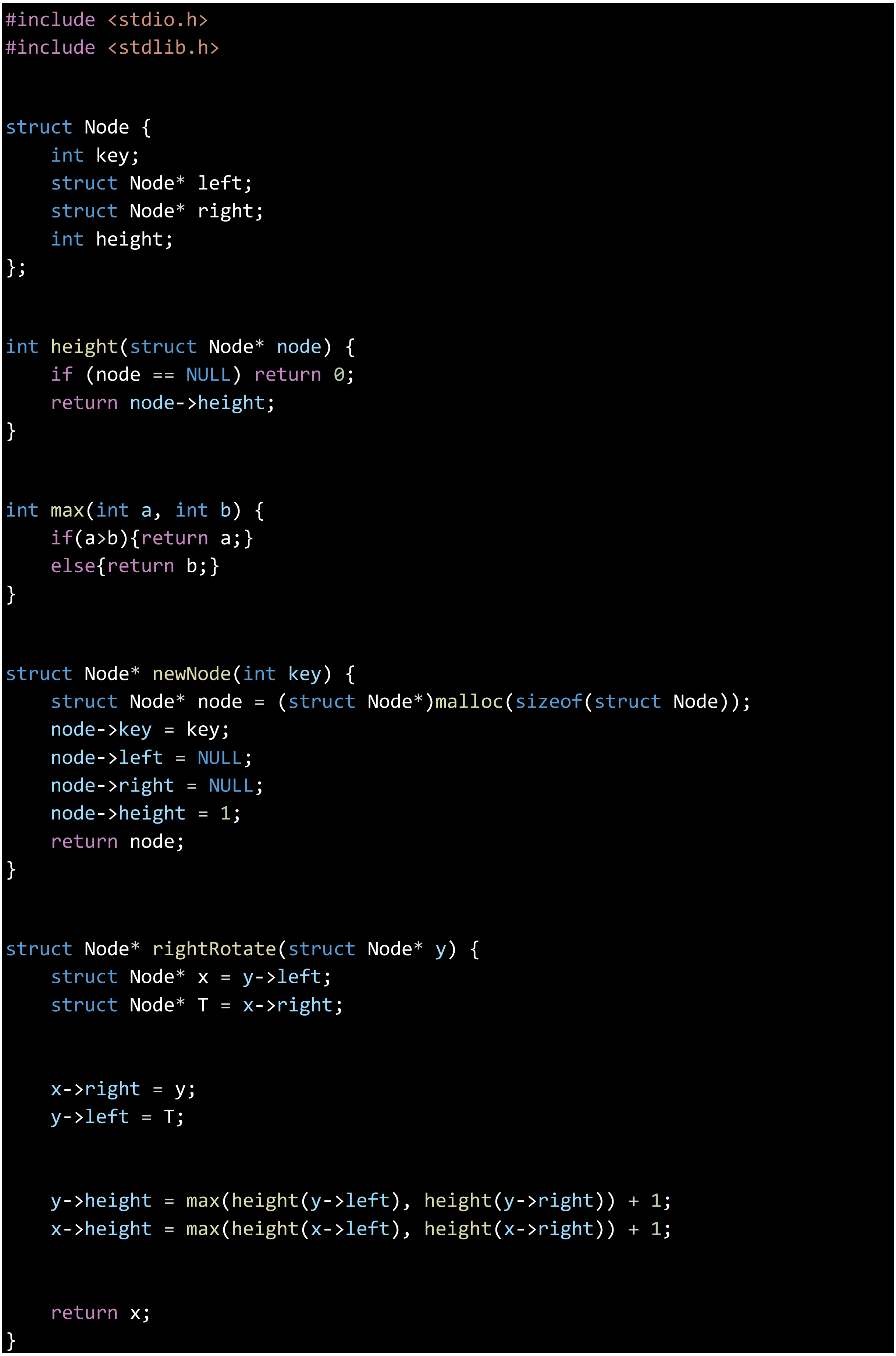
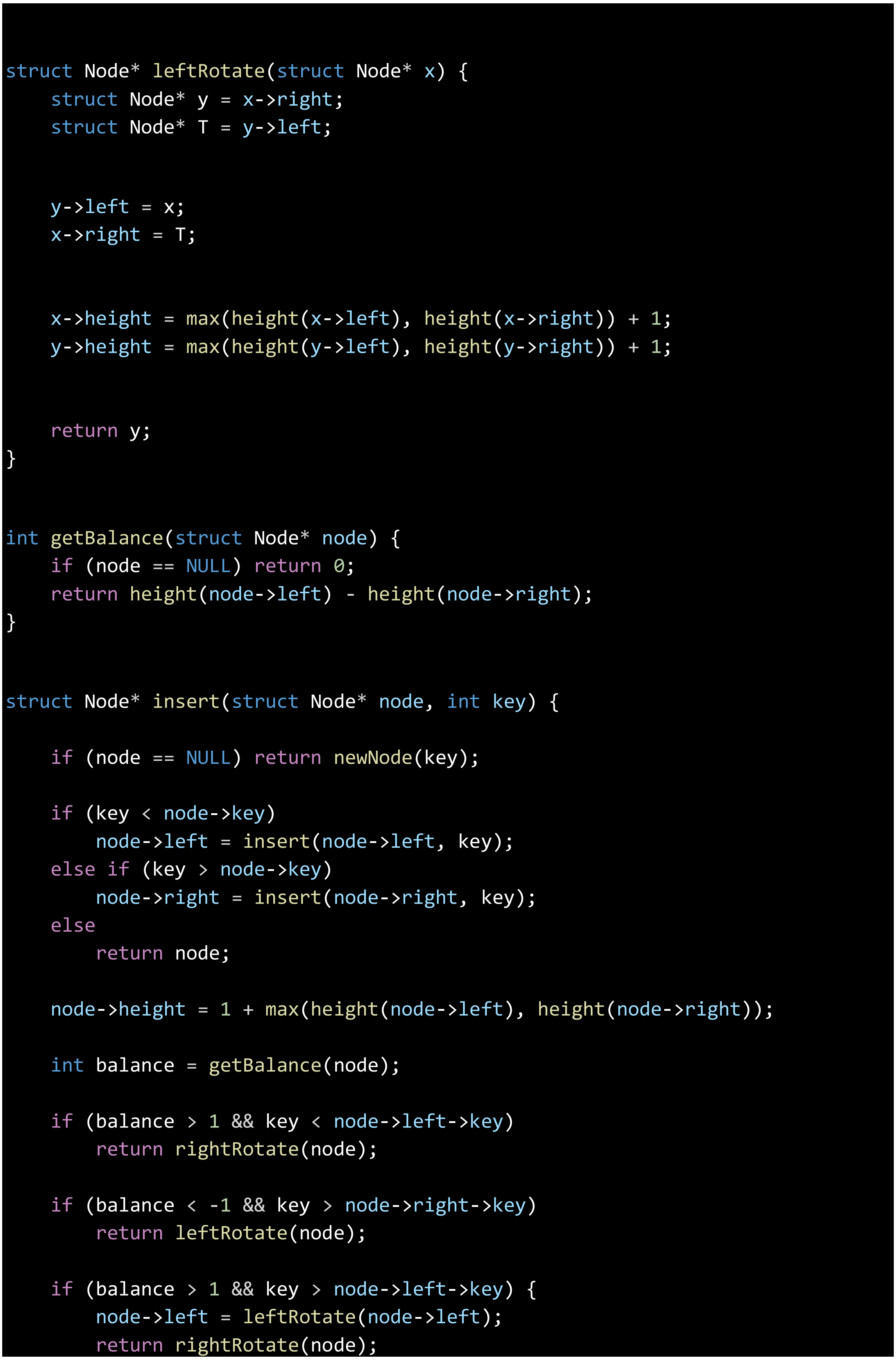
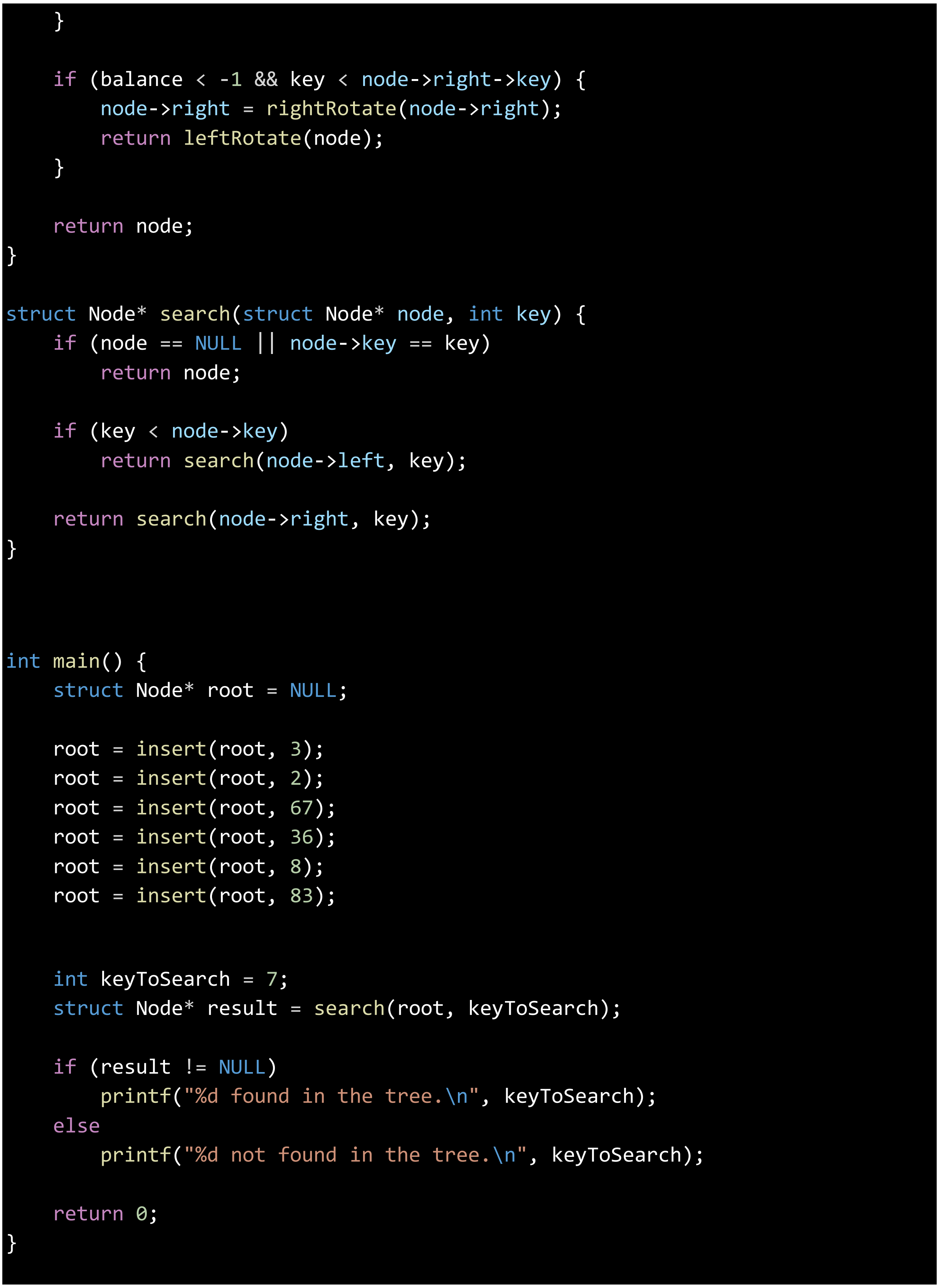


```c
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int key;
    struct Node* left;
    struct Node* right;
    int height;
};

int height(struct Node* node) {
    if (node == NULL) return 0;
    return node->height;
}

int max(int a, int b) {
    if(a>b){return a;}
    else{return b;}
}

struct Node* newNode(int key) {
    struct Node* node = (struct Node*)malloc(sizeof(struct Node));
    node->key = key;
    node->left = NULL;
    node->right = NULL;
    node->height = 1;
    return node;
}

struct Node* rightRotate(struct Node* y) {
    struct Node* x = y->left;
    struct Node* T = x->right;

    x->right = y;
    y->left = T;

    y->height = max(height(y->left), height(y->right)) + 1;
    x->height = max(height(x->left), height(x->right)) + 1;

    return x;
}
```



```c
struct Node* leftRotate(struct Node* x) {
    struct Node* y = x->right;
    struct Node* T = y->left;


    y->left = x;
    x->right = T;


    x->height = max(height(x->left), height(x->right)) + 1;
    y->height = max(height(y->left), height(y->right)) + 1;


    return y;
}


int getBalance(struct Node* node) {
    if (node == NULL) return 0;
    return height(node->left) - height(node->right);
}


struct Node* insert(struct Node* node, int key) {

    if (node == NULL) return newNode(key);

    if (key < node->key)
        node->left = insert(node->left, key);
    else if (key > node->key)
        node->right = insert(node->right, key);
    else
        return node;

    node->height = 1 + max(height(node->left), height(node->right));

    int balance = getBalance(node);

    if (balance > 1 && key < node->left->key)
        return rightRotate(node);

    if (balance < -1 && key > node->right->key)
        return leftRotate(node);

    if (balance > 1 && key > node->left->key) {
        node->left = leftRotate(node->left);
        return rightRotate(node);
```



```
    }

    if (balance < -1 && key < node->right->key) {
        node->right = rightRotate(node->right);
        return leftRotate(node);
    }

    return node;
}

struct Node* search(struct Node* node, int key) {
    if (node == NULL || node->key == key)
        return node;

    if (key < node->key)
        return search(node->left, key);

    return search(node->right, key);
}



int main() {
    struct Node* root = NULL;

    root = insert(root, 3);
    root = insert(root, 2);
    root = insert(root, 67);
    root = insert(root, 36);
    root = insert(root, 8);
    root = insert(root, 83);


    int keyToSearch = 7;
    struct Node* result = search(root, keyToSearch);

    if (result != NULL)
        printf("%d found in the tree.\n", keyToSearch);
    else
        printf("%d not found in the tree.\n", keyToSearch);

    return 0;
}
```