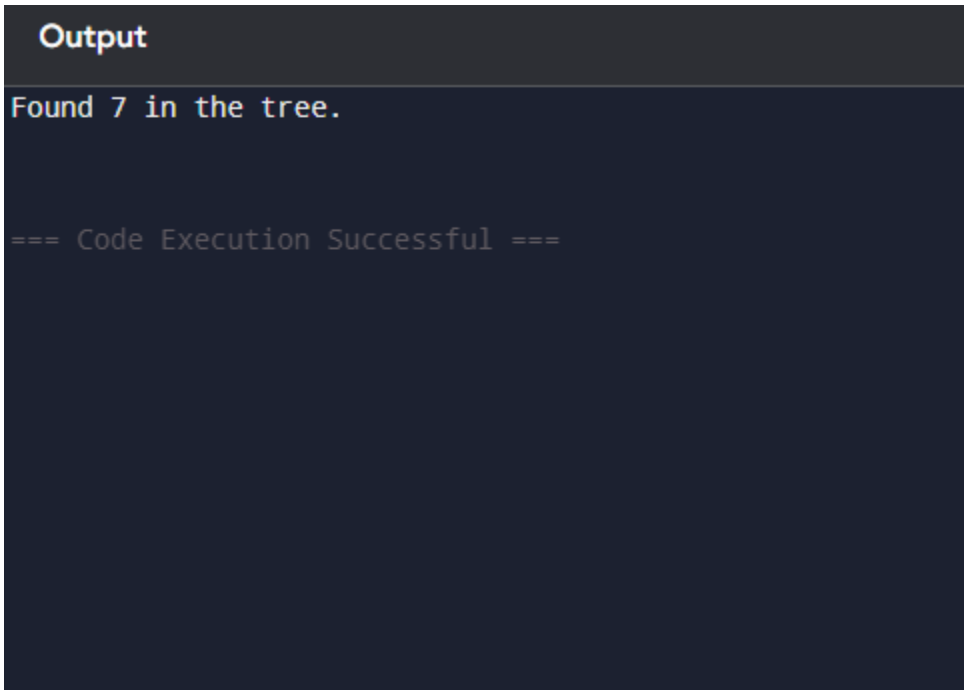
```
17. #include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node *left, *right;
};

struct Node* createNode(int val) {
   struct Node* node = (struct Node*)malloc(sizeof(struct Node));
   node->data = val;
   node->left = node->right = NULL;
   return node;
}

struct Node* insert(struct Node* root, int val) {
   if (root == NULL) return createNode(val);
   if (val < root->data)
      root->left = insert(root->left, val);
   else
      root->right = insert(root->right, val);
   return root;
}

int search(struct Node* root, int key) {
```

```
    if (root == NULL) return 0;
    if (root->data == key) return 1;
    if (key < root->data)
        return search(root->left, key);
    else
        return search(root->right, key);
}

int main() {
    struct Node* root = NULL;
    root = insert(root, 10);
    insert(root, 5);
    insert(root, 20);
    insert(root, 3);
    insert(root, 7);

    int key = 7;
    if (search(root, key))
        printf("Found %d in the tree.\n", key);
    else
        printf("%d not found in the tree.\n", key);

    return 0;
}
```


Output
Found 7 in the tree.
=== Code Execution Successful ===