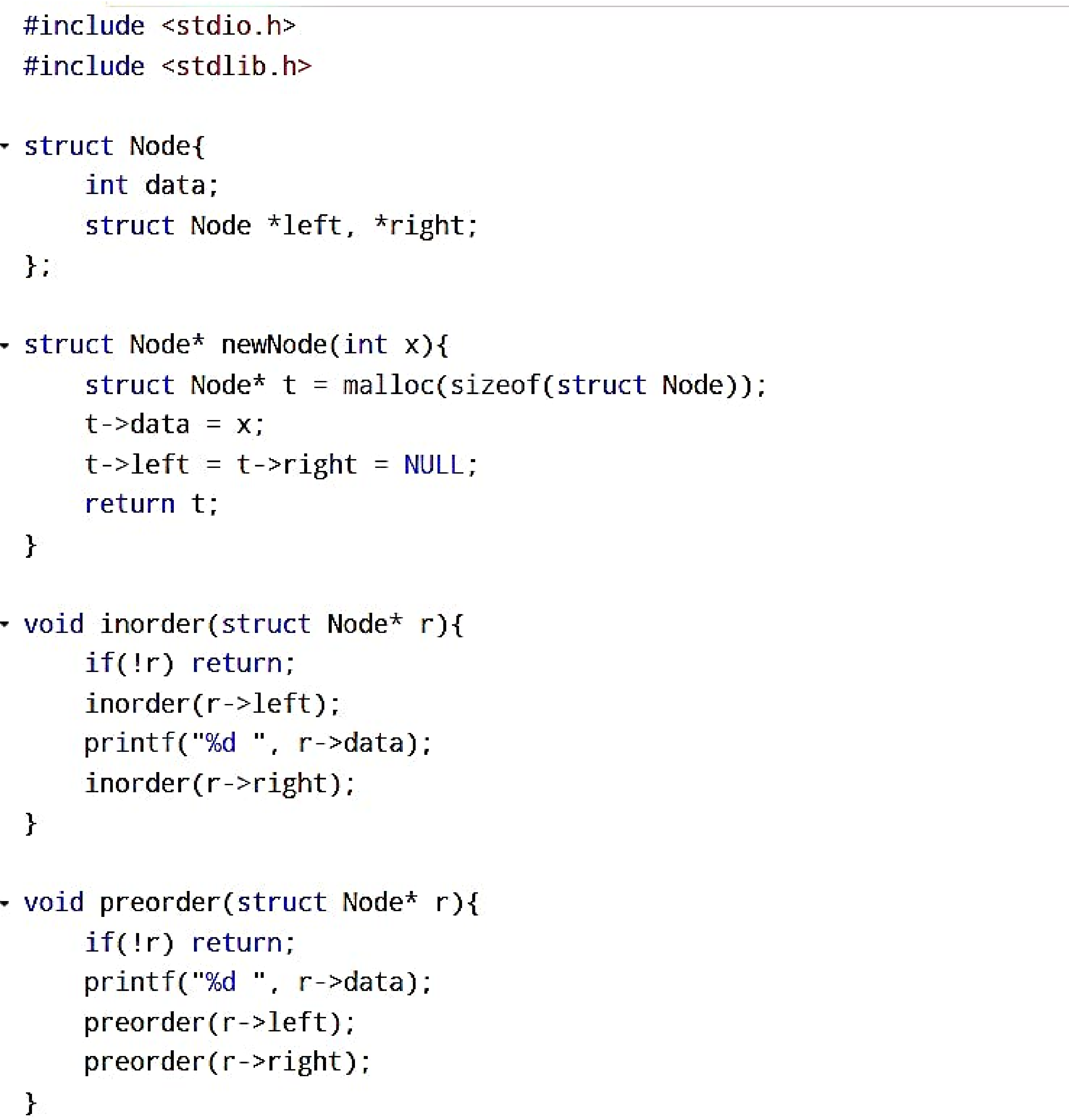


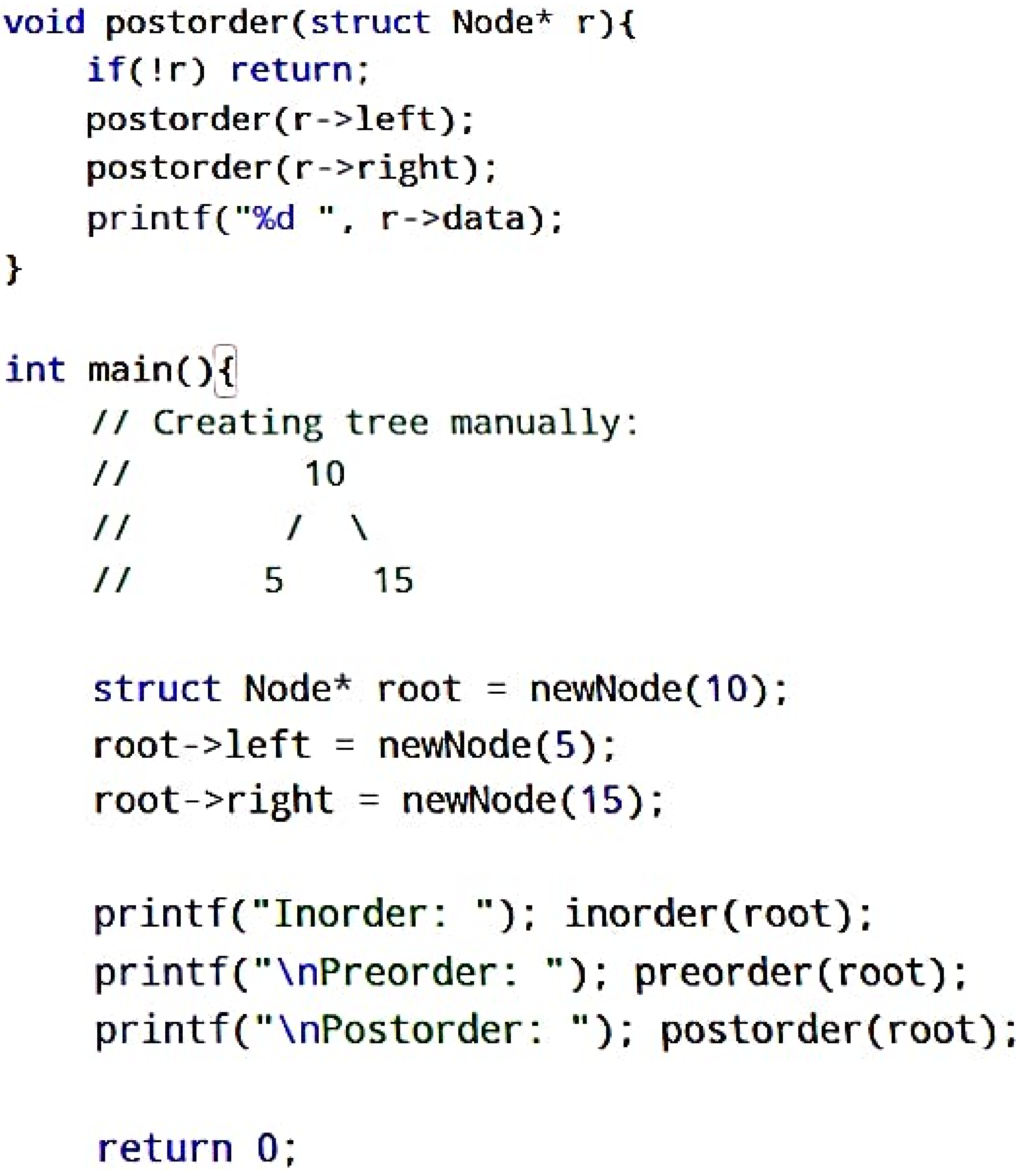
```c
#include <stdio.h>
#include <stdlib.h>

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

struct Node* newNode(int x){
    struct Node* t = malloc(sizeof(struct Node));
    t->data = x;
    t->left = t->right = NULL;
    return t;
}

void inorder(struct Node* r){
    if(!r) return;
    inorder(r->left);
    printf("%d ", r->data);
    inorder(r->right);
}

void preorder(struct Node* r){
    if(!r) return;
    printf("%d ", r->data);
    preorder(r->left);
    preorder(r->right);
}
```



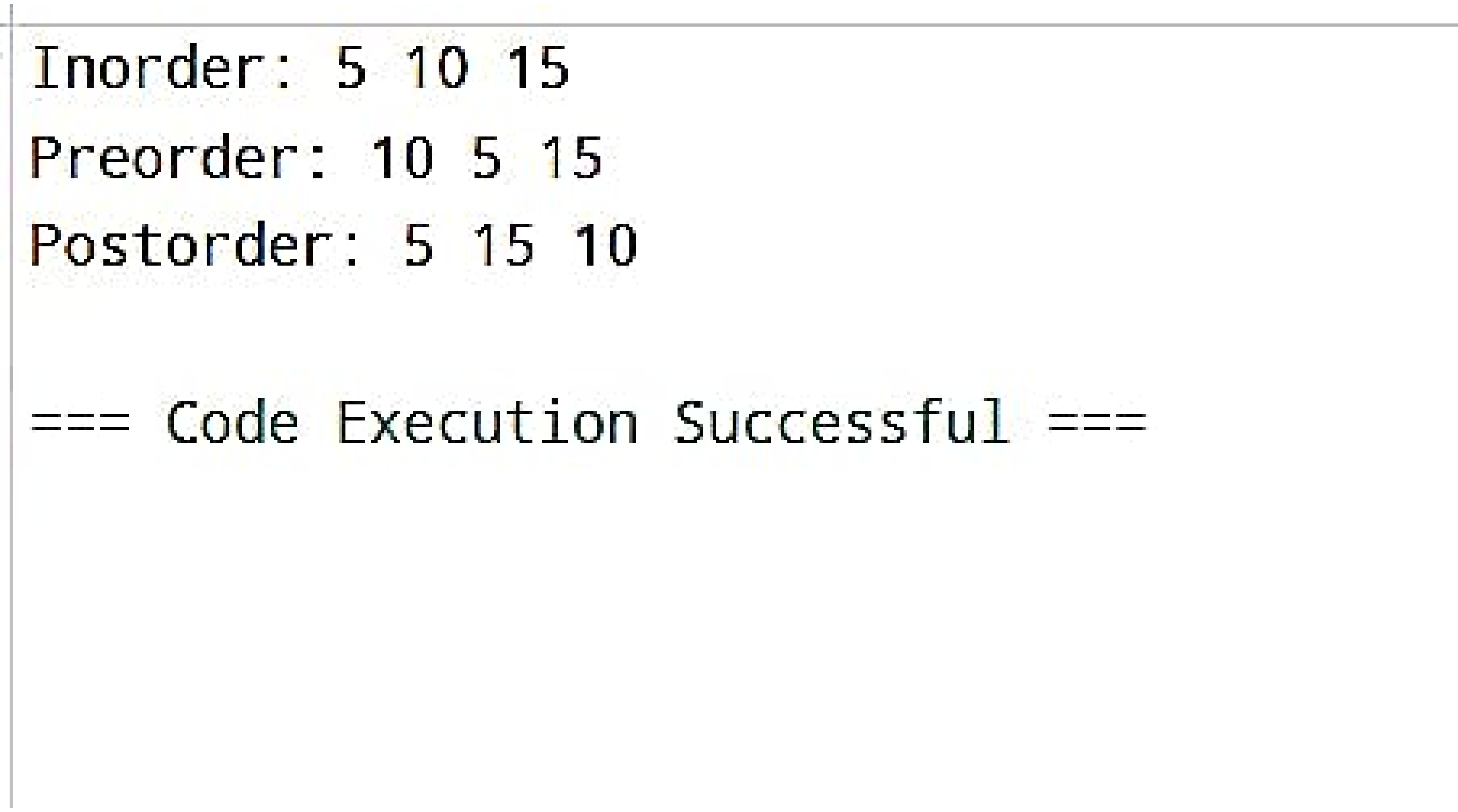
```c
void postorder(struct Node* r){
    if(!r) return;
    postorder(r->left);
    postorder(r->right);
    printf("%d ", r->data);
}

int main(){
    // Creating tree manually:
    //        10
    //       /  \
    //      5    15

    struct Node* root = newNode(10);
    root->left = newNode(5);
    root->right = newNode(15);

    printf("Inorder: "); inorder(root);
    printf("\nPreorder: "); preorder(root);
    printf("\nPostorder: "); postorder(root);

    return 0;
```



```
Inorder: 5 10 15
Preorder: 10 5 15
Postorder: 5 15 10

=== Code Execution Successful ===
```