1. Inorder :

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node *left;
    struct node *right;
};
struct node *createNode(int data) {
    struct node *newNode = (struct node *)malloc(sizeof(struct node));
    newNode->data = data;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}
void inorder(struct node *root) {
    if (root == NULL)
        return;
    inorder(root->left);
    printf("%d ", root->data);
    inorder(root->right);
}
int main() {
    struct node *root = createNode(10);
    root->left = createNode(5);
    root->right = createNode(15);
    root->left->left = createNode(2);
    root->left->right = createNode(7);
    root->right->left = createNode(12);
    root->right->right = createNode(20);
    printf("Inorder traversal of the tree is: ");
    inorder(root);
    return 0;
}
```

2. Preorder :

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node *left;
    struct node *right;
};
struct node *createNode(int data) {
    struct node *newNode = (struct node *)malloc(sizeof(struct node));
    newNode->data = data;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}
void preorder(struct node *root) {
    if (root == NULL)
        return;
    printf(" %d ", root->data);
    preorder(root->left);
    preorder(root->right);
}
int main() {
    struct node *root = createNode(10);
    root->left = createNode(20);
    root->right = createNode(30);
    root->left->left = createNode(40);
    root->left->right = createNode(50);
    root->right->left = createNode(60);
    root->right->right = createNode(70);
    printf("Preorder traversal of binary tree is: ");
    preorder(root);
    return 0;
}
```

3. Postorder :

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node *left;
    struct node *right;
};
struct node *newNode(int data) {
    struct node *temp = (struct node *)malloc(sizeof(struct node));
    temp->data = data;
    temp->left = NULL;
    temp->right = NULL;
    return temp;
}
void postorder(struct node *root) {
    if (root == NULL)
        return;
    postorder(root->left);
    postorder(root->right);
    printf("%d ", root->data);
}
int main() {
    struct node *root = newNode(10);
    root->left = newNode(20);
    root->right = newNode(30);
    root->left->left = newNode(40);
    root->left->right = newNode(50);
    root->right->left = newNode(60);
    root->right->right = newNode(70);
    printf("Postorder traversal of binary tree is: ");
    postorder(root);
    return 0;
}
```