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

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

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

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

void preorder(struct Node* root) {
    if (root == NULL) return;
```

```c
    printf("%d ", root->data);

    preorder(root->left);

    preorder(root->right);

}


void postorder(struct Node* root) {

    if (root == NULL) return;

    postorder(root->left);

    postorder(root->right);

    printf("%d ", root->data);

}


int main() {

    struct Node* root = createNode(1);

    root->left = createNode(2);

    root->right = createNode(3);

    root->left->left = createNode(4);

    root->left->right = createNode(5);


    printf("Inorder: ");

    inorder(root);

    printf("\nPreorder: ");

    preorder(root);

    printf("\nPostorder: ");

    postorder(root);
```

```c
    printf("\n");


    return 0;

}
```

```
Output                                          Clear

Inorder: 4 2 5 1 3
Preorder: 1 2 4 5 3
Postorder: 4 5 2 3 1


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

\