

## CSA0317-DATA STRUCTURES

### Program 14

```
#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) {
        inorder(root->left);
        printf("%d ", root->data);
        inorder(root->right);
    }
}
```

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

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

```
int main() {  
    // Creating a sample tree  
    //    1  
    //   /\  
    //  2 3  
    // /\   
    // 4 5  
  
    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 traversal: ");
inorder(root);
printf("\n");
printf("Preorder traversal: ");
preorder(root);
printf("\n");
printf("Postorder traversal: ");
postorder(root);
printf("\n");
return 0;
}
```

Output:

Output
Inorder traversal: 4 2 5 1 3 Preorder traversal: 1 2 4 5 3 Postorder traversal: 4 5 2 3 1
=== Code Execution Successful ===

