DAY8 LAB

1. Write a C program to implement a binary tree and perform in-order, pre-order, and post-order traversal.

```
#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() {
    struct Node* root = createNode(1);
    root->left = createNode(2);
    root->right = createNode(3);
    root->left->left = createNode(4);
    root->left->right = createNode(5);

    printf("In-order traversal: ");
    inOrder(root);
    printf("\n");

    printf("Pre-order traversal: ");
    preOrder(root);
    printf("\n");

    printf("Post-order traversal: ");
    postOrder(root);
    printf("\n");
    return 0;
}
```

OUTPUT:

```
In-order traversal: 4 2 5 1 3
Pre-order traversal: 1 2 4 5 3
Post-order traversal: 4 5 2 3 1
```