#include <stdio.h>

#include <stdlib.h>

struct Node {

    int data;

    struct Node\* left;

    struct Node\* right;

};

struct Node\* buildTree() {

    int data;

    struct Node\* newNode = NULL;

    printf("Enter the data (or 0 to exit): ");

    scanf("%d", &data);

    if (data != 0) {

        newNode = (struct Node\*)malloc(sizeof(struct Node));

        if (newNode == NULL) {

            printf("Memory allocation failed!\n");

            exit(1);

        }

        newNode->data = data;

        printf("Enter the left child of %d:\n", data);

        newNode->left = buildTree();

        printf("Enter the right child of %d:\n", data);

        newNode->right = buildTree();

    }

    return newNode;

}

void inorderTraversal(struct Node\* root) {

    if (root == NULL)

        return;

    inorderTraversal(root->left);

    printf("%d ", root->data);

    inorderTraversal(root->right);

}

void preorderTraversal(struct Node\* root) {

    if (root == NULL)

        return;

    printf("%d ", root->data);

    preorderTraversal(root->left);

    preorderTraversal(root->right);

}

void postorderTraversal(struct Node\* root) {

    if (root == NULL)

        return;

    postorderTraversal(root->left);

    postorderTraversal(root->right);

    printf("%d ", root->data);

}

int main() {

    printf("Binary Tree Traversal\n");

    printf("Build the binary tree:\n");

    struct Node\* root = buildTree();

    printf("\nInorder traversal: ");

    inorderTraversal(root);

    printf("\n");

    printf("Preorder traversal: ");

    preorderTraversal(root);

    printf("\n");

    printf("Postorder traversal: ");

    postorderTraversal(root);

    printf("\n");

    return 0;

}