#include <stdio.h>

#include <stdlib.h>

// Node structure

struct Node {

int data;

struct Node\* left;

struct Node\* right;

};

// Function to create new node

struct Node\* createNode(int value) {

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

newNode->data = value;

newNode->left = newNode->right = NULL;

return newNode;

}

// Inorder Traversal: Left -> Root -> Right

void inorder(struct Node\* root) {

if (root == NULL) return;

inorder(root->left);

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

inorder(root->right);

}

// Preorder Traversal: Root -> Left -> Right

void preorder(struct Node\* root) {

if (root == NULL) return;

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

preorder(root->left);

preorder(root->right);

}

// Postorder Traversal: Left -> Right -> Root

void postorder(struct Node\* root) {

if (root == NULL) return;

postorder(root->left);

postorder(root->right);

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

}

int main() {

// Manually creating a binary 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;

}  
