#include <stdio.h>

#include <stdlib.h>

struct Node {

int value;

struct Node\* left;

struct Node\* right;

};

void inorder(struct Node\* root) {

if (root != NULL) {

inorder(root->left);

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

inorder(root->right);

}

}

void preorder(struct Node\* root) {

if (root != NULL) {

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

preorder(root->left);

preorder(root->right);

}

}

void postorder(struct Node\* root) {

if (root != NULL) {

postorder(root->left);

postorder(root->right);

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

}

}

struct Node\* newNode(int value) {

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

node->value = value;

node->left = NULL;

node->right = NULL;

return node;

}

int main() {

int choice, value;

struct Node\* root = NULL;

printf("Enter root value: ");

scanf("%d", &value);

root = newNode(value);

while (1) {

printf("\n1. Insert left\n2. Insert right\n3. Inorder traversal\n4. Preorder traversal\n5. Postorder traversal\n6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter left child value: ");

scanf("%d", &value);

root->left = newNode(value);

break;

case 2:

printf("Enter right child value: ");

scanf("%d", &value);

root->right = newNode(value);

break;

case 3:

printf("Inorder: ");

inorder(root);

printf("\n");

break;

case 4:

printf("Preorder: ");

preorder(root);

printf("\n");

break;

case 5:

printf("Postorder: ");

postorder(root);

printf("\n");

break;

case 6:

exit(0);

default:

printf("Invalid choice!\n");

}

}

return 0;

}

Enter root value: 20

1. Insert left

2. Insert right

3. Inorder traversal

4. Preorder traversal

5. Postorder traversal

6. Exit

Enter your choice: 1

Enter left child value: 15

1. Insert left

2. Insert right

3. Inorder traversal

4. Preorder traversal

5. Postorder traversal

6. Exit

Enter your choice: 2

Enter right child value: 12

1. Insert left

2. Insert right

3. Inorder traversal

4. Preorder traversal

5. Postorder traversal

6. Exit

Enter your choice: 3

Inorder: 15 20 12

1. Insert left

2. Insert right

3. Inorder traversal

4. Preorder traversal

5. Postorder traversal

6. Exit

Enter your choice: 4

Preorder: 20 15 12

1. Insert left

2. Insert right

3. Inorder traversal

4. Preorder traversal

5. Postorder traversal

6. Exit

Enter your choice: 5

Postorder: 15 12 20

1. Insert left

2. Insert right

3. Inorder traversal

4. Preorder traversal

5. Postorder traversal

6. Exit