**Preorder**

#include <iostream >

using namespace std;

struct Node {

int data;

struct Node \*left, \*right;

Node (int v) {

data = v;

left = right = NULL;

}

};

// pre-order traversal

void printPreorder (struct Node\* node) {

if (node == NULL)

return;

cout << node->data << " ";

printPreorder (node->left);

printPreorder (node->right);

}

int main () {

struct Node\* root = new Node (1);

root->left = new Node (2);

root->right = new Node (3);

root->left->left = new Node (4);

root->left->right = new Node (5);

root->right->right = new Node (6);

cout << "Preorder traversal of binary tree is: ";

printPreorder (root);

return 0;

}

**Post -order**

#include < iostream >

using namespace std;

struct Node {

int data;

struct Node \*left, \*right;

Node (int v) {

data = v;

left = right = NULL;

}

};

// post-order traversal

void printPostorder (struct Node\* node) {

if (node == NULL)

return;

printPostorder (node->left);

printPostorder (node->right);

cout << node->data << " ";

}

int main () {

struct Node\* root = new Node (1);

root->left = new Node (2);

root->right = new Node (3);

root->left->left = new Node (4);

root->left->right = new Node (5);

root->right->right = new Node (6);

cout << "Postorder traversal of binary tree is: \\n";

printPostorder (root);

return 0;

}

**In-order**

#include < iostream >

using namespace std;

struct Node {

int data;

struct Node \*left, \*right;

Node (int v) {

data = v;

left = right = NULL;

}

};

// in-order traversal

void printInorder (struct Node\* node) {

if (node == NULL)

return;

printInorder (node->left);

cout << node->data << " ";

printInorder (node->right);

}

int main () {

struct Node\* root = new Node (1);

root->left = new Node (2);

root->right = new Node (3);

root->left->left = new Node (4);

root->left->right = new Node (5);

root->right->right = new Node (6);

cout << "Inorder traversal of binary tree is: \\n";

printInorder (root);

return 0;

}

``