**Institute of Computer Technology**

**B. Tech. Computer Science and Engineering**

**Sub: DS**

**Course Code: 2CSE302**

**Practical – 13**

**Name: Jaymin Gondaliya**

**Enrollment No: 23162171007**

**Sem - 3**

**Branch: CS**

**Class: A**

**Batch: 32**

**Problem Definition-1:** To understand the concept of binary tree implementation: Create following mentioned tree and display the elements using any traversal algorithm (in order, pre order, post order)

**Code:**

*#include* <stdio.h>

*#include* <stdlib.h>

typedef struct Node {

    int data;

    struct Node\* left;

    struct Node\* right;

} Node;

Node\* createNode(int *data*) {

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

    newNode->data = *data*;

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

*return* newNode;

}

void inOrder(Node\* *root*) {

*if* (*root* == NULL) *return*;

    inOrder(*root*->left);

    printf("%d ", *root*->data);

    inOrder(*root*->right);

}

void preOrder(Node\* *root*) {

*if* (*root* == NULL) *return*;

    printf("%d ", *root*->data);

    preOrder(*root*->left);

    preOrder(*root*->right);

}

void postOrder(Node\* *root*) {

*if* (*root* == NULL) *return*;

    postOrder(*root*->left);

    postOrder(*root*->right);

    printf("%d ", *root*->data);

}

int main() {

    Node\* root = createNode(4);

    root->left = createNode(2);

    root->right = createNode(3);

    printf("In-order Traversal: ");

    inOrder(root);

    printf("\n");

    printf("Pre-order Traversal: ");

    preOrder(root);

    printf("\n");

    printf("Post-order Traversal: ");

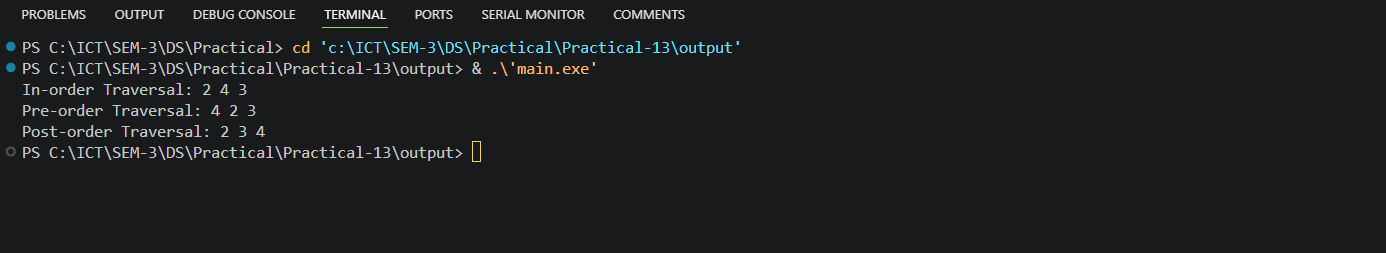
    postOrder(root);

    printf("\n");

*return* 0;

}

**Output:**

****