Institute of Computer Technology

B. Tech Computer Science and Engineering

Subject: DS (2CSE302)

**PRACTICAL-16**

**AIM: - Implement the real-life scenario using binary tree.**

Cisco Systems, Inc. is an American multinational technology, which sells networking hardware, software, telecommunications equipment, and other high-technology services and products. Roshni is working at Cisco, Ahmedabad and she wants to capture all orders of telecommunications equipment in the form of a binary tree as each order is linked to the previous one in a parent-child relationship. Here, each node is having the constraint that it has either two children or zero. Kindly perform the below operation using C language:

1. **Create the binary tree of the given list of order-Id (define max-size as 7)**

**20, 15, 30, 25, 19, 31, 45**

1. **Print root node (level-0), level-1, and level-2 node.**
2. **Perform the below binary tree traversal operations:**

**Preorder traversal**

**Inorder traversal**

**Postorder traversal**

**Input:**

20 15 30 25 19 31 45

**Output:**

Root element of binary tree is: 20

Level-1 element of binary tree is: 15 30

Level-2 element of binary tree is: 25 19 31 45

Preorder traversal of binary tree is: 20 15 25 19 30 31 45

Inorder traversal of binary tree is: 25 15 19 20 31 30 45

Postorder traversal of binary tree is: 25 19 15 31 45 30 20

***SOLUTION***

#include <stdio.h>

#include <malloc.h>

struct node {

int yash;

struct node \*leftNode;

struct node \*rightNode;

};

struct node \*newNode(int value){

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

node->yash=value;

node->leftNode=NULL;

node->rightNode=NULL;

return node;

}

void printCurrentLevel(struct node \*root ,int lvl){

if(root==NULL){

return;

}

if(lvl==0){

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

}

if(lvl>0){

printCurrentLevel(root->leftNode,lvl-1);

printCurrentLevel(root->rightNode,lvl-1);

}

}

void PreOrder(struct node \*ptr)

{

if(ptr!=NULL)

{

printf(" %d ",ptr->yash);

PreOrder(ptr->leftNode);

PreOrder(ptr->rightNode);

}

}

void InOrder(struct node \*ptr)

{

if(ptr!=NULL)

{

InOrder(ptr->leftNode);

printf(" %d ",ptr->yash);

InOrder(ptr->rightNode);

}

}

void PostOrder(struct node \*ptr)

{

if(ptr!=NULL)

{

PostOrder(ptr->leftNode);

PostOrder(ptr->rightNode);

printf(" %d ",ptr->yash);

}

}

int Insert()

{

int rawdata=0;

scanf("%d",&rawdata);

return rawdata;

}

int main(){

printf("\n-----| I N P U T |-----\n");

struct node \*root= newNode(Insert());

root->leftNode=newNode(Insert());

root->rightNode=newNode(Insert());

root->leftNode->leftNode=newNode(Insert());

root->leftNode->rightNode=newNode(Insert());

root->rightNode->leftNode=newNode(Insert());

root->rightNode->rightNode=newNode(Insert());

printf("\n\n----------| O U T P U T |----------\n");

printf("Root element of binary tree is:\t\t");

printCurrentLevel(root,0);

printf("\nLevel-1 element of binary tree is:\t");

printCurrentLevel(root,1);

printf("\nLevel-2 element of binary tree is:\t");

printCurrentLevel(root,2);

printf("\n\nPreorder traversal of binary tree is:\t");

PreOrder(root);

printf("\nInorder traversal of binary tree is:\t");

InOrder(root);

printf("\nPostorder traversal of binary tree is:\t");

PostOrder(root);

printf("\n");

}

***OUTPUT***

