|  |  |
| --- | --- |
| **EX.No. 3** | **TREE TRAVERSAL OPERATIONS** |
| **Date : 03.01.2025** |

**AIM:**

To perform the Tree traversal operations like Inorder, Preorder and Post order

**ALGORITHM:**

STEP 1: Start the program

STEP 2: Declare a node using structure with left and right pointers

STEP 3: createNode( ): This function creates and allocates a space in memory for each node in a tree

STEP 4: preorder( ): This function prints the tree elements in a preorder using recursion

STEP 5: inorder( ): This function prints the elements in an inorder format using recursion

STEP 6: postorder( ): This function prints the elements in a postorder traversal using recursion

STEP 7: Stop the exeuction

**PROGRAM CODING:**

#include <stdio.h>

#include <stdlib.h>

struct node {

int element;

struct node\* left;

struct node\* right;

};

/\*To create a new node\*/

struct node\* createNode(int val)

{

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

Node->element = val;

Node->left = NULL;

Node->right = NULL;

return (Node);

}

/\*function to traverse the nodes of binary tree in preorder\*/

void preorder(struct node\* root)

{

if (root == NULL)

return;

printf("%d\t", root->element);

preorder(root->left);

preorder(root->right);

}

/\*function to traverse the nodes of binary tree in Inorder\*/

void inorder(struct node\* root)

{

if (root == NULL)

return;

inorder(root->left);

printf("%d\t", root->element);

inorder(root->right);

}

/\*function to traverse the nodes of binary tree in postorder\*/

void postorder(struct node\* root)

{

if (root == NULL)

return;

postorder(root->left);

postorder(root->right);

printf("%d\t", root->element);

}

int main()

{

struct node\* root = createNode(36);

root->left = createNode(26);

root->right = createNode(46);

root->left->left = createNode(21);

root->left->right = createNode(31);

root->left->left->left = createNode(11);

root->left->left->right = createNode(24);

root->right->left = createNode(41);

root->right->right = createNode(56);

printf("\n The Preorder traversal of given binary tree is -\n");

preorder(root);

printf("\n The Inorder traversal of given binary tree is -\n");

inorder(root);

printf("\n The Postorder traversal of given binary tree is -\n");

postorder(root);

return 0;

}

**RESULT:**

Thus the above program implemented the tree traversal operations successfully.