**Ex. No. 5 Binary Tree Traversal**

**Date:**

**Aim :**

To implement different types of traversal for the given binary tree

**Algorithm:**

1. Create a structure with key and 2 pointer variable left and right
2. 2. Read the node to be inserted.

If (root==NULL)

root=node

else if (root->key < node->key)

root->right=NULL

else

Root->left=node

1. For Inorder Traversal

Traverse Left subtree

Visit root

Traverse Right subtree

1. For Preorder Traversal

Visit root

Traverse Left subtree

Traverse Right subtree

1. For Postorder Traversal

Traverse Left subtree

Traverse Right subtree

Visit root

1. Stop

**Program:**

/\* Tree Traversal \*/

#include <stdio.h>

#include <stdlib.h>

typedef struct node

{

int data;

struct node \*left;

struct node \*right;

}node;

int count=1;

node \*insert(node \*tree,int digit)

{

if(tree == NULL)

{

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

tree->left = tree->right=NULL;

tree->data = digit;

count++;

}

else if(count%2 == 0)

tree->left = insert(tree->left, digit);

else

tree->right = insert(tree->right, digit);

return tree;

}

void preorder(node \*t)

{

if(t != NULL)

{

printf(" %d", t->data);

preorder(t->left);

preorder(t->right);

}

}

void postorder(node \*t)

{

if(t != NULL)

{

postorder(t->left);

postorder(t->right);

printf(" %d", t->data);

}

}

void inorder(node \*t)

{

if(t != NULL)

{

inorder(t->left);

printf(" %d", t->data);

inorder(t->right);

}

}

void main()

{

node \*root = NULL;

int digit;

puts("Enter integer:To quit enter 0");

scanf("%d", &digit);

while(digit != 0)

{

root=insert(root,digit);

scanf("%d",&digit);

}

printf("\nThe preorder traversal of tree is:\n");

preorder(root);

printf("\nThe inorder traversal of tree is:\n");

inorder(root);

printf("\nThe postorder traversal of tree is:\n");

postorder(root);

}

**Output:**

**Result:**

Thus three types of tree traversal was performed on the given binary tree