**Name :: himanshu kumar**

**USN :: 1nt21is063**

**Sec :: A**

**AIM –** Binary Tree and its Traversals

**ALGORITHM –**

Preorder(tree) -

Visit root node of the tree

call preorder(left subtree) //recursion call

call preorder(right subtree) //recursion call

Inorder(tree) -

call inorder(left subtree) //recursion call

Visit the root node of the tree

call inorder(right subtree) // recursion call

Postorder(tree) -

call postorder(left subtree) // recursion call

call postorder(right subtree) // recursion call

Visit the root node of the tree

**CODE -**

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*left, \*right;

} ;

struct node \*create()

{

struct node \*temp;

int data,choice;

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

printf("Press 0 to exit");

printf("\nPress 1 for new node");

printf("\nEnter your choice : ");

scanf("%d", &choice);

if(choice==0)

{

return 0;

}

else

{

printf("Enter the data:");

scanf("%d", &data);

temp->data = data;

printf("Enter the left child of %d \n", data);

temp->left = create();

printf("Enter the right child of %d \n", data);

temp->right = create();

return temp;

}

}

void inorder(struct node \*root){

if(root==NULL){

return;

}

else {

inorder(root->left);

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

inorder(root->right);

}

}

void postorder(struct node \*root){

if(root==NULL){

return;

}

else {

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

postorder(root->left);

postorder(root->right);

}

}

void preorder(struct node \*root){

if(root==NULL){

return;

}

else {

preorder(root->left);

preorder(root->right);

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

}

}

void main()

{

struct node \*root;

root = 0;

root = create();

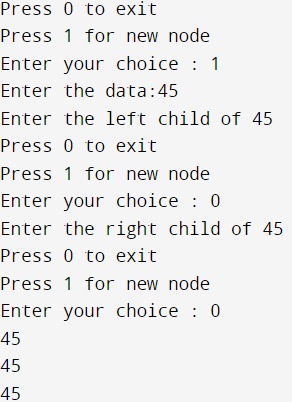
inorder(root);

preorder(root);

postorder(root);

}

**OUTPUT –**

****

Link:<https://github.com/himanshu120299/Lab-assignment.git>