**Program No. 16**

Objective:-WAP for Implementation of Tree

#include<stdio.h>

#include<stdlib.h>

struct node\*head=NULL;

struct node

{

int data;

struct node\*llink;

struct node\*rlink;

};

struct node\*Aakash\_createnewnode()

{

struct node \*newNode;

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

printf("\nEnter new node=\n");

scanf("%d",&newNode->data);

newNode->llink=NULL;

newNode->rlink=NULL;

return newNode;

}

void insert(struct node\*ptr, struct node\*newNode)

{

char choice;

printf("Where you want to add new node:");

printf("\nFor left PRESS 'l' or 'L':");

printf("\nFor right PRESS 'r' or 'R':");

choice=getch();

if(choice=='R'||choice=='r')

{

if(ptr->rlink==NULL)

{

ptr->rlink=newNode;

}

else

{

insert(ptr->rlink,newNode);

}

}

else if(choice=='l'||choice=='L')

{

if(ptr->llink==NULL)

{

ptr->llink=newNode;

}

else

{

insert(ptr->llink,newNode);

}

}

}

void preorder(struct node\*ptr)

{

if(ptr!=NULL)

{

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

preorder(ptr->llink);

preorder(ptr->rlink);

}

}

void postorder(struct node\*ptr)

{

if(ptr!=NULL)

{

postorder(ptr->llink);

postorder(ptr->rlink);

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

}

}

void inorder(struct node\*ptr)

{

if(ptr!=NULL)

{

inorder(ptr->llink);

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

inorder(ptr->rlink);

}

}

main()

{

char mychoice;

struct node\*newNode,\*ptr;

ptr=NULL;

do

{

newNode=Aakash\_createnewnode();

if(ptr==NULL)

{

ptr=newNode;

}

else

{

insert(ptr,newNode);

}

printf("\nDo you want to add further leaf on tree Prees 'Y' & 'y' for yes and any key to exit");

mychoice=getch();

}

while(mychoice=='y'||mychoice=='Y');

printf("\n\nYour TREE is in Pre Order form\n\n");

preorder(ptr);

printf("\n\nYour TREE is in Post Order form\n\n");

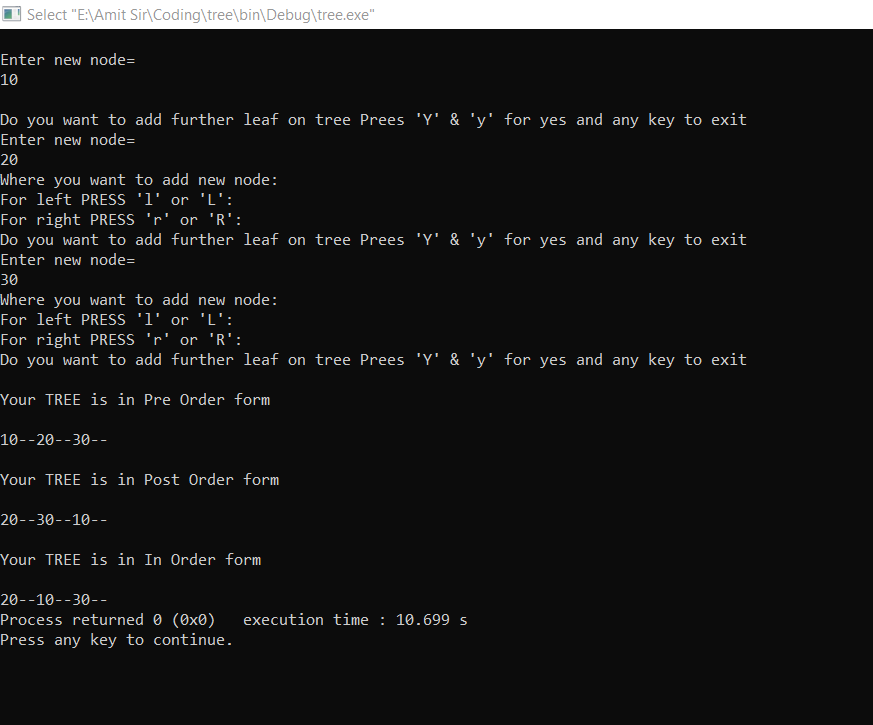
postorder(ptr);

printf("\n\nYour TREE is in In Order form\n\n");

inorder(ptr);

}

Output:-



Implementation of Tree