// Implementation of binary tree in C //

#include<stdio.h>

#include<stdlib.h>

Struct node

{

Int element;

Struct node\*left;

Struct node\*right;

};

Struct node\*createNode(int val)

{

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

Node->element=val;

Node->right=NULL;

Node->left=NULL;

Return(node);

}

Void traversepreorder(struct node\*root)

{

If(root==NULL)

Return;

Printf(“%d\t”,root->element);

Traversepreorder(root->left);

Traversepreorder(root->right);

}

Void traversepostorder(struct node\*root)

{

If(root==NULL)

Return;

Traversepostorder(root->left);

Traversepostorder(root->right);

Printf(“%d\t”,root->element);

}

Void traverseinorder(struct node\*root)

{

If(root==NULL)

Return;

Traverseinorder(root->right);

}

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);

Root->right->right->left=createnode(51);

Root->right->right->right=createnode(66);

Printf(“pre order traversal\n”);

Traversepreorder(root);

Printf(“\npostorder traversal\n”);

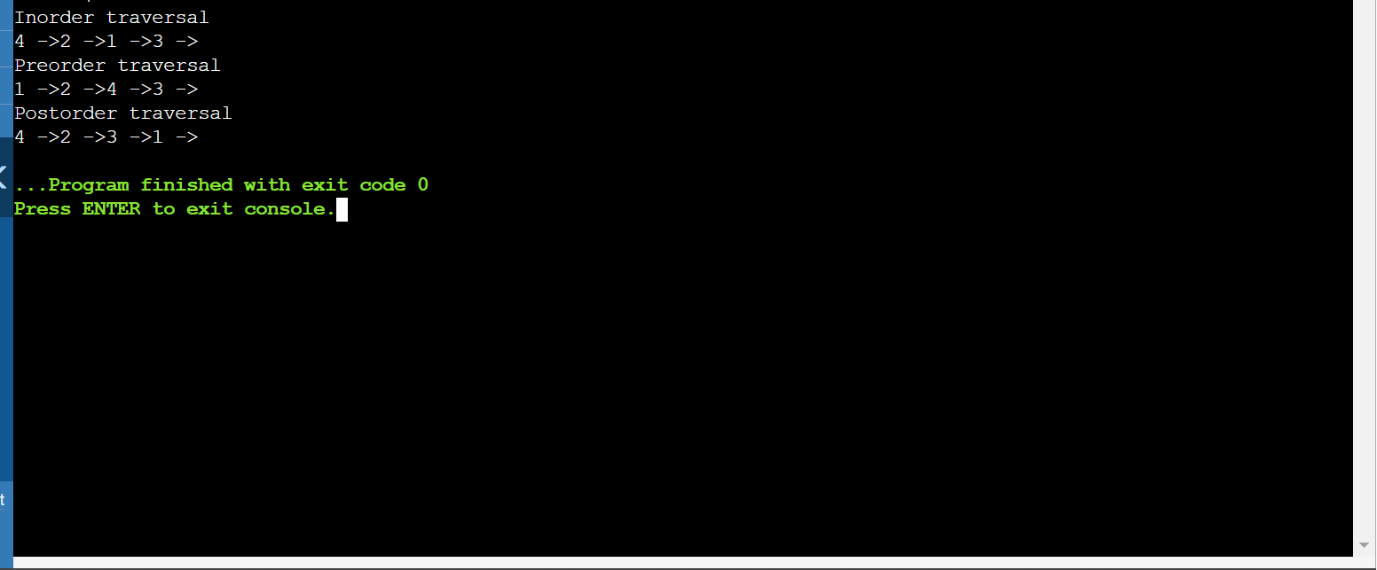
Traversepostorder(root);

Printf(“\inorder traversal\n”);

Traverseinorder(root);

}

Out put:



Github link :