#include<stdio.h>

#include<stdlib.h>

Struct Node

{

Int data;

Struct Node \*left;

Struct Node \*right;

};

Struct Node\* create\_node(int data)

{

Struct Node \*newnode= struct Node\* malloc(sizeof(struct Node)):

newnode->data=data;

newnode->left=NULL;

newnode->right=NULL;

return newnode;

}

Struct Node\* insert(struct Node \*root, int data)

{

if (root==NULL)

root=create\_node(data)

else if( data<root->data)

root->left=insert(root->left, data);

else if(data>root->data)

root->right=insert(root->right, data);

return root;

}

Void inorder(struct Node \*root)

{

Printf(“%d”, inorder(root->left));

Printf(“%d”, root->data);

Printf(“%d”,inorder( root->right));

}

Void preorder(struct Node \*root)

{

Printf(“%d”, root->data);

Printf(“%d”,preorder(root->left));

Printf(“%d”,preorder(root->right));

}

Void postorder(struct Node \*root)

{

Printf(“%d”,postorder(root->left));

Printf(“%d”, postorder(root->right));

Printf(“%d”,root->data);

}

int main()

{

Struct Node \*root=NULL;

insert(10);

insert(20);

insert(40);

insert(65);

insert(25);

insert(50);

printf(“Inorder traversal\n”);

inorder(root);

printf(“pre order traversal\n”);

preorder(root);

printf(“post order traversal\n”);

postorder(root);

return 0;

}