PROGRAM 14:

```c
#include<stdio.h>

#include<stdlib.h>

struct node{

 struct node* left;

 int data;

 struct node* right;

};

struct node* createTree();

void preOrder(struct node* );

void postOrder(struct node* );

void inOrder(struct node* );

int main(){

 struct node* root = NULL;

 root = createTree();

 printf("the pre order is:\n");

 preOrder(root);


 printf("the post order is:\n");

 postOrder(root);

 printf("the inoder is :\n");

 inOrder(root);

}

struct node* createTree(){

 struct node* newnode = NULL;

 newnode = (struct node*)malloc(sizeof(struct node));

 int data;

 printf("enter the data(-1 for no node)");

 scanf("%d",&data);

 if(data==-1)
```

```c
    return 0;
    newnode->data = data;
    printf("enter the left child of %d\n",data);
    newnode->left = createTree();
    /*
    here we are using the recursion
    storing the result of createTree() in new node left side
    */
    printf("enter the right child of %d\n",data);
    newnode->right = createTree();
    return newnode;
}
void preOrder(struct node* root){
if(root == NULL){
    return ;
    }
    printf("%d ",root->data);
    preOrder(root->left);
    preOrder(root->right);
}
void postOrder(struct node* root){
if(root==NULL){
return ;
    }
    postOrder(root->left);
    postOrder(root->right);
    printf("%d ",root->data);
}
void inOrder(struct node* root){
if(root == NULL){
return;
```
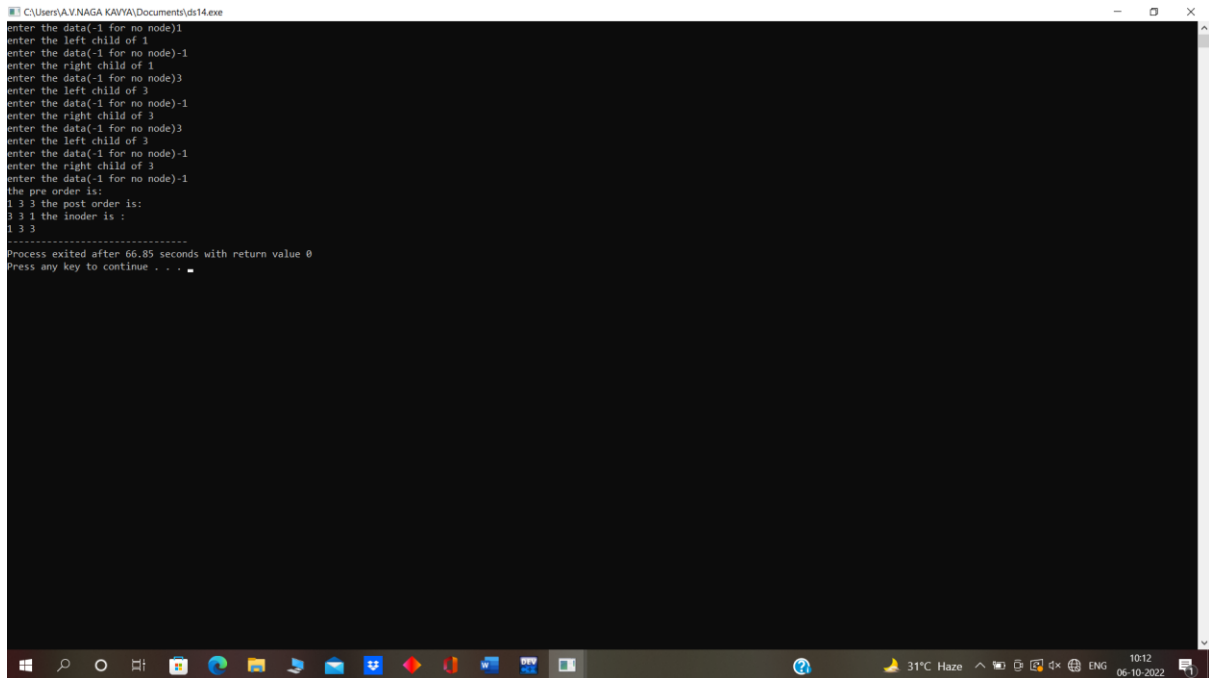
```
            }

            inOrder(root->left);

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

            inOrder(root->right);

        }
```

OUTPUT: