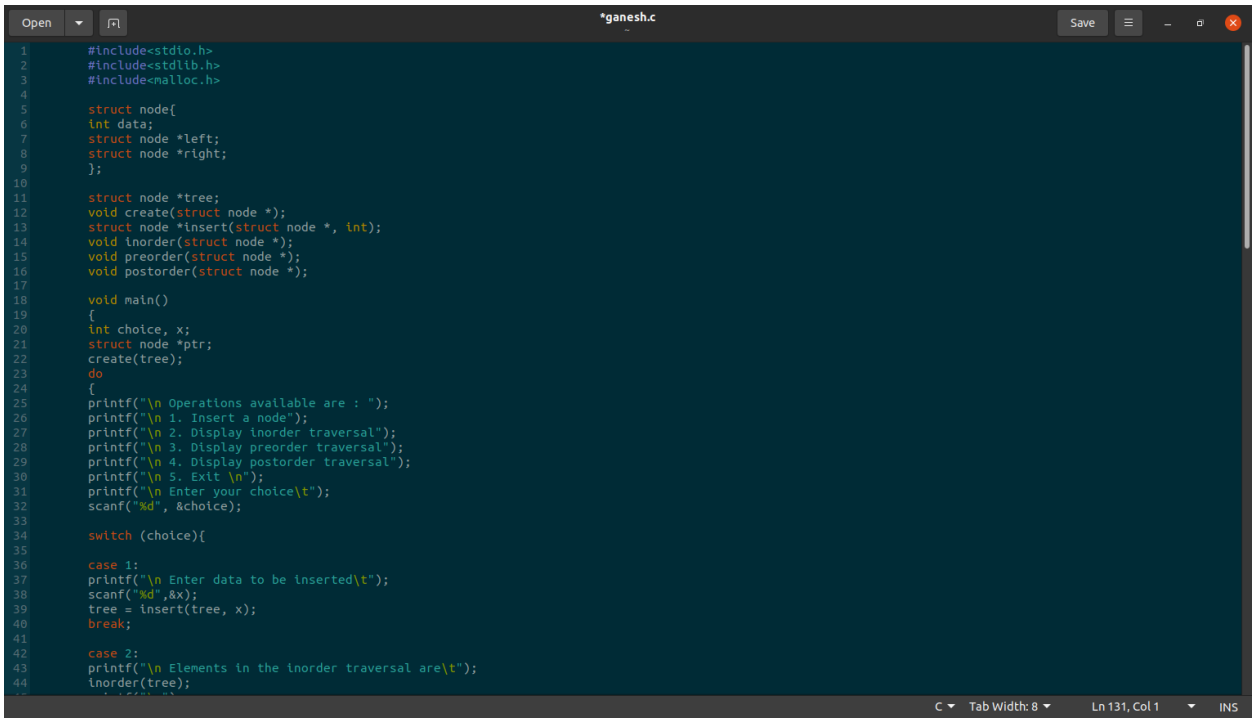
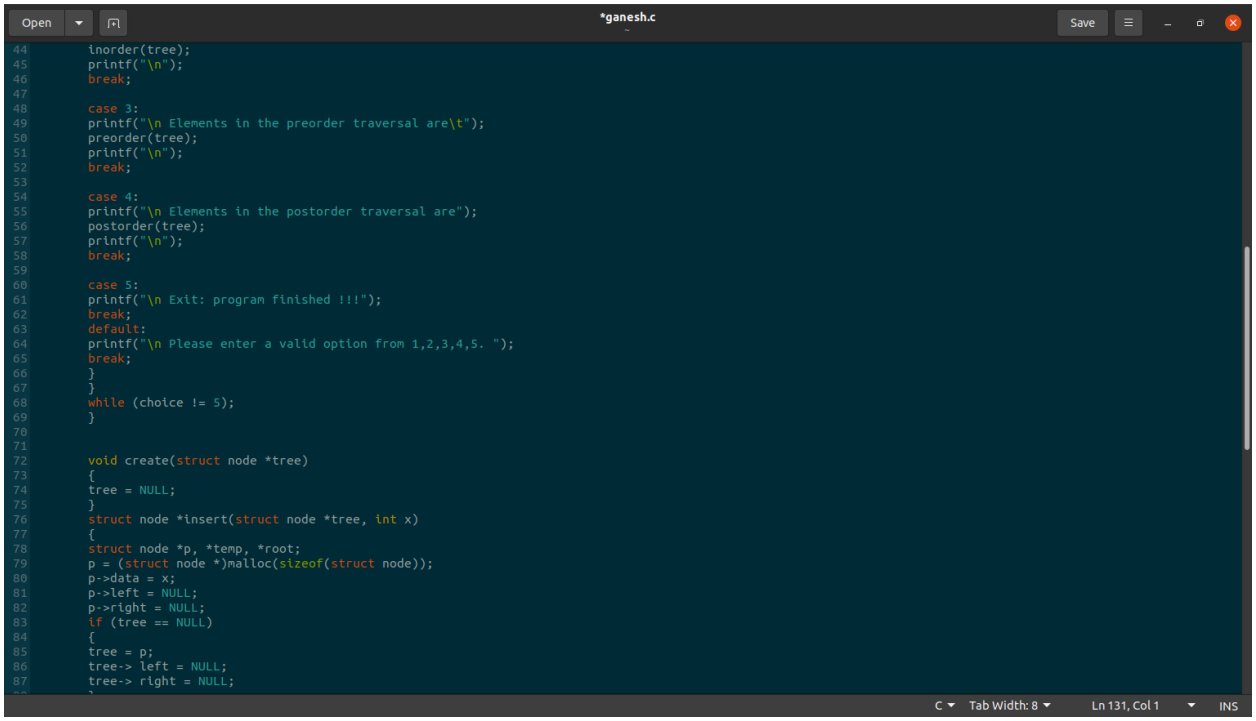
Name: Shreeganesh Vishwakarma
Class: Sy_IT Roll No.65
Experiment 06




```c
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>

struct node{
int data;
struct node *left;
struct node *right;
};

struct node *tree;
void create(struct node *);
struct node *insert(struct node *, int);
void inorder(struct node *);
void preorder(struct node *);
void postorder(struct node *);

void main()
{
int choice, x;
struct node *ptr;
create(tree);
do
{
printf("\n Operations available are : ");
printf("\n 1. Insert a node");
printf("\n 2. Display inorder traversal");
printf("\n 3. Display preorder traversal");
printf("\n 4. Display postorder traversal");
printf("\n 5. Exit \n");
printf("\n Enter your choice\t");
scanf("%d", &choice);

switch (choice){

case 1:
printf("\n Enter data to be inserted\t");
scanf("%d",&x);
tree = insert(tree, x);
break;

case 2:
printf("\n Elements in the inorder traversal are\t");
inorder(tree);
```




```c
inorder(tree);
printf("\n");
break;

case 3:
printf("\n Elements in the preorder traversal are\t");
preorder(tree);
printf("\n");
break;

case 4:
printf("\n Elements in the postorder traversal are");
postorder(tree);
printf("\n");
break;

case 5:
printf("\n Exit: program finished !!!");
break;
default:
printf("\n Please enter a valid option from 1,2,3,4,5. ");
break;
}
}
while (choice != 5);
}


void create(struct node *tree)
{
tree = NULL;
}
struct node *insert(struct node *tree, int x)
{
struct node *p, *temp, *root;
p = (struct node *)malloc(sizeof(struct node));
p->data = x;
p->left = NULL;
p->right = NULL;
if (tree == NULL)
{
tree = p;
tree-> left = NULL;
tree-> right = NULL;
```

Open *ganesh.c Save

```
87        tree-> right = NULL;
88        }
89        else
90        {
91        root = NULL;
92        temp = tree;
93        while (temp != NULL)
94        {
95        root = temp;
96        if (x < temp->data)
97        temp = temp->left;
98        else
99        temp = temp->right;
100       }
101       if(x < root->data)
102       root->left = p;
103       else
104       root->right = p;
105       }
106       return tree;
107       }
108
109
110       void inorder(struct node *tree)
111       {
112       if (tree != NULL)
113       {
114       inorder(tree->left);
115       printf("%d \t", tree->data);
116       inorder(tree->right);
117       }
118       }
119
120
121       void preorder(struct node *tree){
122       if (tree != NULL)
123       {
124       printf("%d \t", tree->data);
125       preorder(tree->left);
126       preorder(tree->right);
127       }
128       }
129
130
131       void postorder(struct node *tree){
```

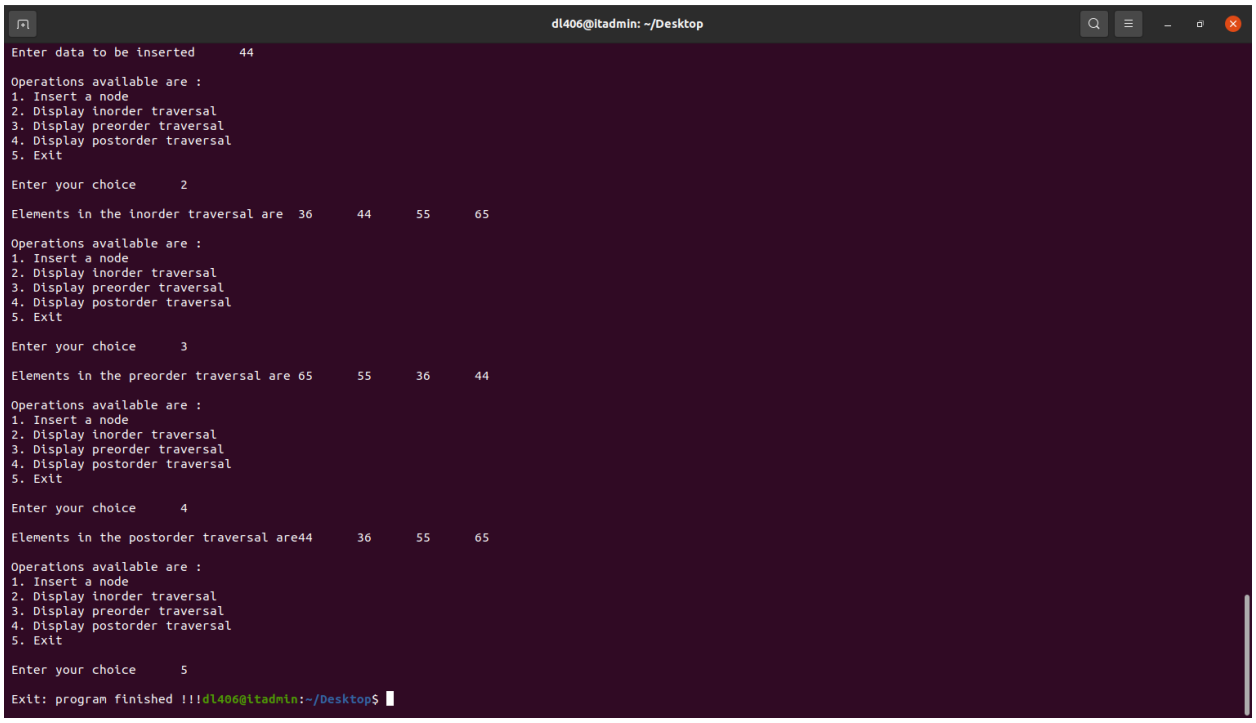
C Tab Width: 8 Ln 87, Col 29 INS

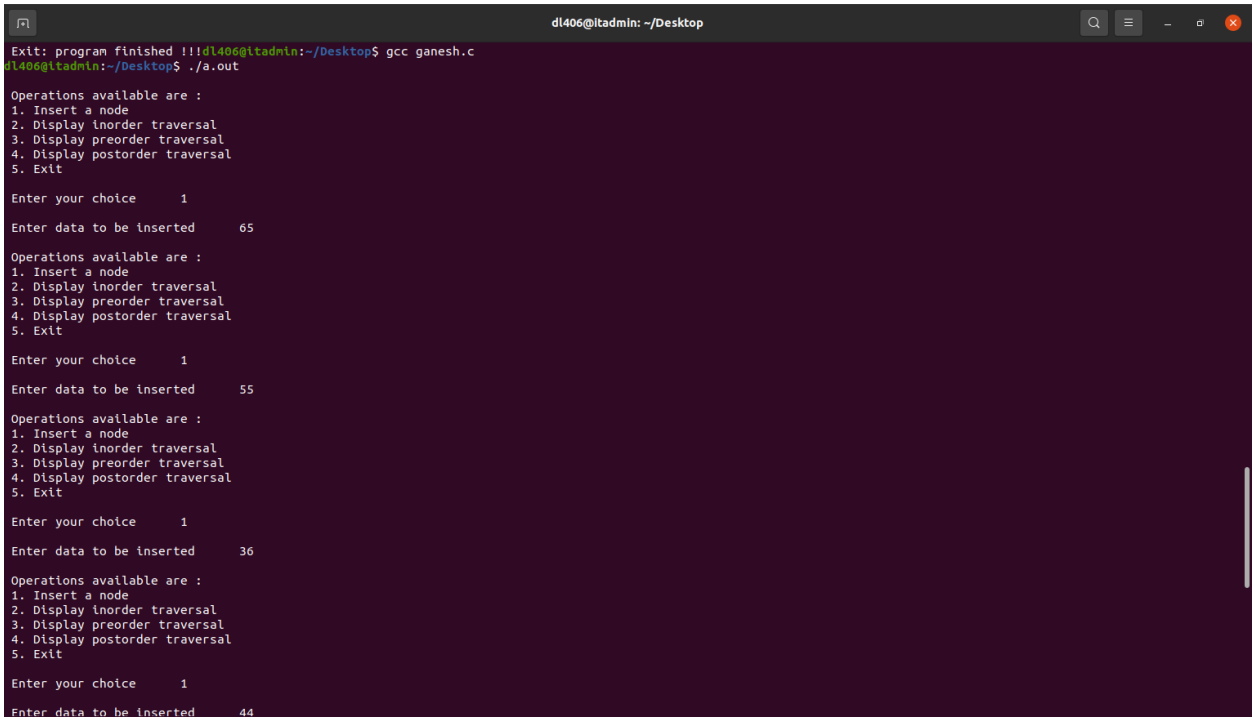
Open *ganesh.c Save

```
95        root = temp;
96        if (x < temp->data)
97        temp = temp->left;
98        else
99        temp = temp->right;
100       }
101       if(x < root->data)
102       root->left = p;
103       else
104       root->right = p;
105       }
106       return tree;
107       }
108
109
110       void inorder(struct node *tree)
111       {
112       if (tree != NULL)
113       {
114       inorder(tree->left);
115       printf("%d \t", tree->data);
116       inorder(tree->right);
117       }
118       }
119
120
121       void preorder(struct node *tree){
122       if (tree != NULL)
123       {
124       printf("%d \t", tree->data);
125       preorder(tree->left);
126       preorder(tree->right);
127       }
128       }
129
130
131       void postorder(struct node *tree){
132       if (tree != NULL)
133       {
134       postorder(tree->left);
135       postorder(tree->right);
136       printf("%d \t", tree->data);
137       }
138       }
```

C Tab Width: 8 Ln 87, Col 29 INS

Output:


dl406@itadmin: ~/Desktop
Enter data to be inserted 44
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit
Enter your choice 2
Elements in the inorder traversal are 36 44 55 65
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit
Enter your choice 3
Elements in the preorder traversal are 65 55 36 44
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit
Enter your choice 4
Elements in the postorder traversal are44 36 55 65
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit
Enter your choice 5
Exit: program finished !!!dl406@itadmin:~/Desktop$


dl406@itadmin: ~/Desktop
Exit: program finished !!!dl406@itadmin:~/Desktop$ gcc ganesh.c
dl406@itadmin:~/Desktop$ ./a.out
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit
Enter your choice 1
Enter data to be inserted 65
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit
Enter your choice 1
Enter data to be inserted 55
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit
Enter your choice 1
Enter data to be inserted 36
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit
Enter your choice 1
Enter data to be inserted 44