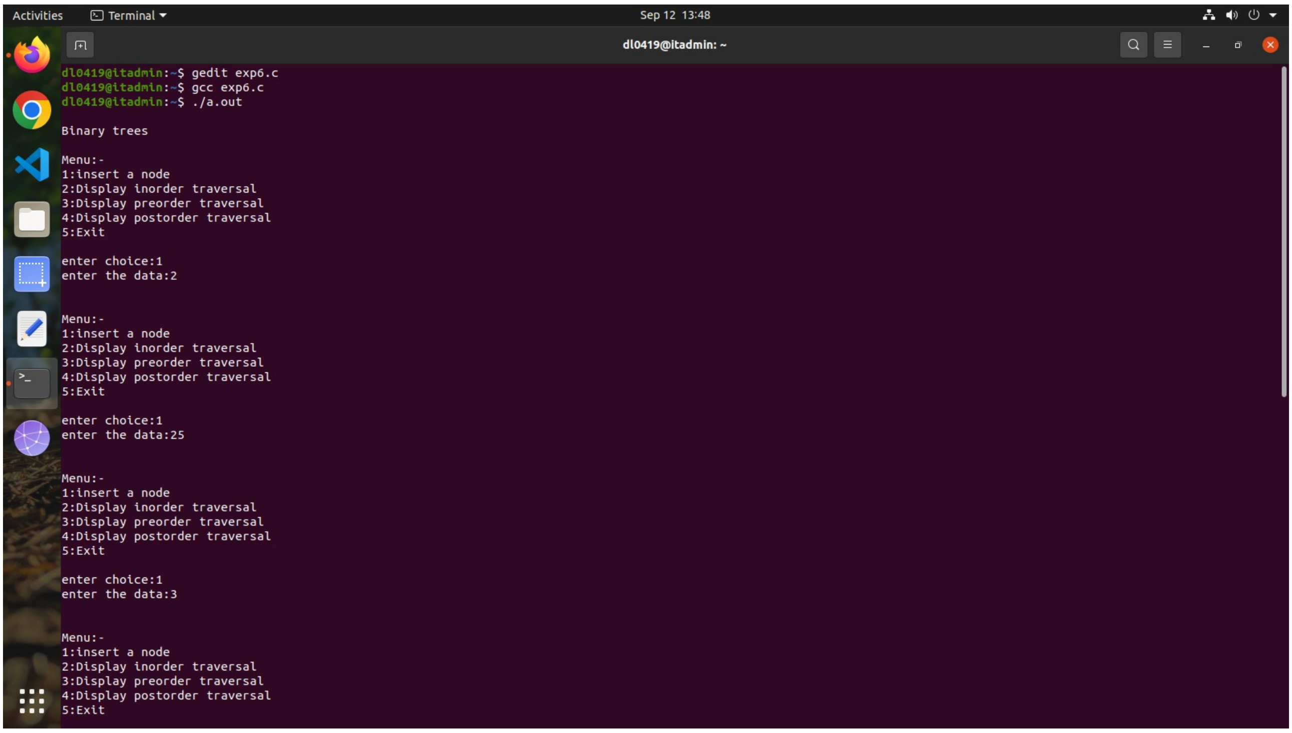
# EXPERIMENT NO.:06

**Aim:**

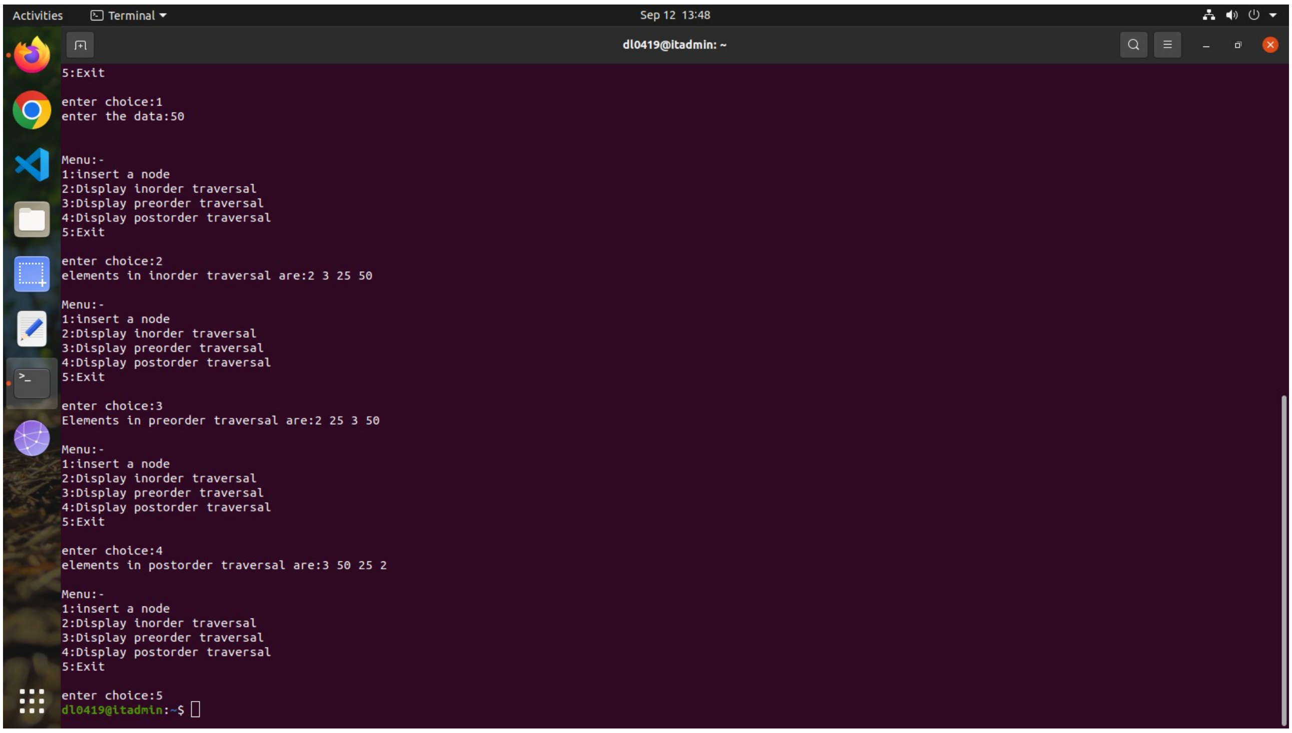
**Program:**

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

struct node{
    int data;
    struct node*L;
    struct node*R;
};
struct node*tree;

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

void create(){
    tree=NULL;
}

struct node*insert(int x){
    struct node*p,*temp,*root;
    p=(struct node*)malloc(sizeof(struct node));
    p->data=x;
    p->L=NULL;
    p->R=NULL;

    if(tree==NULL){
        tree=p;
        tree->L=NULL;
        tree->R=NULL;
    }
    else{
        root=NULL;
        temp=tree;
        while(temp!=NULL){
            root=temp;
            if(x<temp->data)
            temp=temp->L;
            else
            temp=temp->R;
        }
        if(x<root->data)
        root->L=p;
        else
        root->R=p;
    }
    return tree;
}

void inorder(struct node*tree){
    if(tree!=NULL){
        inorder (tree->L);
        printf ("%d ", tree->data);
        inorder (tree->R);
    }
}
void preorder(struct node*tree){
    if(tree!=NULL){
        printf ("%d ", tree->data);
        preorder (tree->L);
        preorder (tree->R);
    }
}
void postorder(struct node*tree){
    if(tree!=NULL){
        postorder (tree->L);
        postorder (tree->R);
        printf ("%d ", tree->data);
    }
}

int main(){
    printf("\nBinary trees");
    int ch,x;
    create();
    do{
        printf("\n\nMenu:-\n1:insert a node\n2:Display inorder traversal\n3:Display preorder traversal\n4:Display postorder traversal\n5:Exit\n");
        printf("\nenter choice:");
        scanf("%d",&ch);
        switch(ch){
            case 1:
                    printf("enter the data:");
                    scanf("%d",&x);
                    tree=insert(x);
                    break;
            case 2:
                printf("elements in inorder traversal are:");
                inorder(tree);
                break;
            case 3:
                    printf("Elements in preorder traversal are:");
                    preorder(tree);
                    break;
            case 4:
                    printf("elements in postorder traversal are:");
                    postorder(tree);
                    break;
            case 5:
                    printf("Exiting");
                    break;
            default:
                    printf("Wrong input");

        }

    }while(ch!=5);
    return 0;
}
```

**Output:**



```
dl0419@itadmin:~$ gedit exp6.c
dl0419@itadmin:~$ gcc exp6.c
dl0419@itadmin:~$ ./a.out

Binary trees

Menu:-
1:insert a node
2:Display inorder traversal
3:Display preorder traversal
4:Display postorder traversal
5:Exit

enter choice:1
enter the data:2


Menu:-
1:insert a node
2:Display inorder traversal
3:Display preorder traversal
4:Display postorder traversal
5:Exit

enter choice:1
enter the data:25


Menu:-
1:insert a node
2:Display inorder traversal
3:Display preorder traversal
4:Display postorder traversal
5:Exit

enter choice:1
enter the data:3


Menu:-
1:insert a node
2:Display inorder traversal
3:Display preorder traversal
4:Display postorder traversal
5:Exit
```



```
5:Exit

enter choice:1
enter the data:50


Menu:-
1:insert a node
2:Display inorder traversal
3:Display preorder traversal
4:Display postorder traversal
5:Exit

enter choice:2
elements in inorder traversal are:2 3 25 50

Menu:-
1:insert a node
2:Display inorder traversal
3:Display preorder traversal
4:Display postorder traversal
5:Exit

enter choice:3
Elements in preorder traversal are:2 25 3 50

Menu:-
1:insert a node
2:Display inorder traversal
3:Display preorder traversal
4:Display postorder traversal
5:Exit

enter choice:4
elements in postorder traversal are:3 50 25 2

Menu:-
1:insert a node
2:Display inorder traversal
3:Display preorder traversal
4:Display postorder traversal
5:Exit

enter choice:5
dl0419@itadmin:~$ 
```