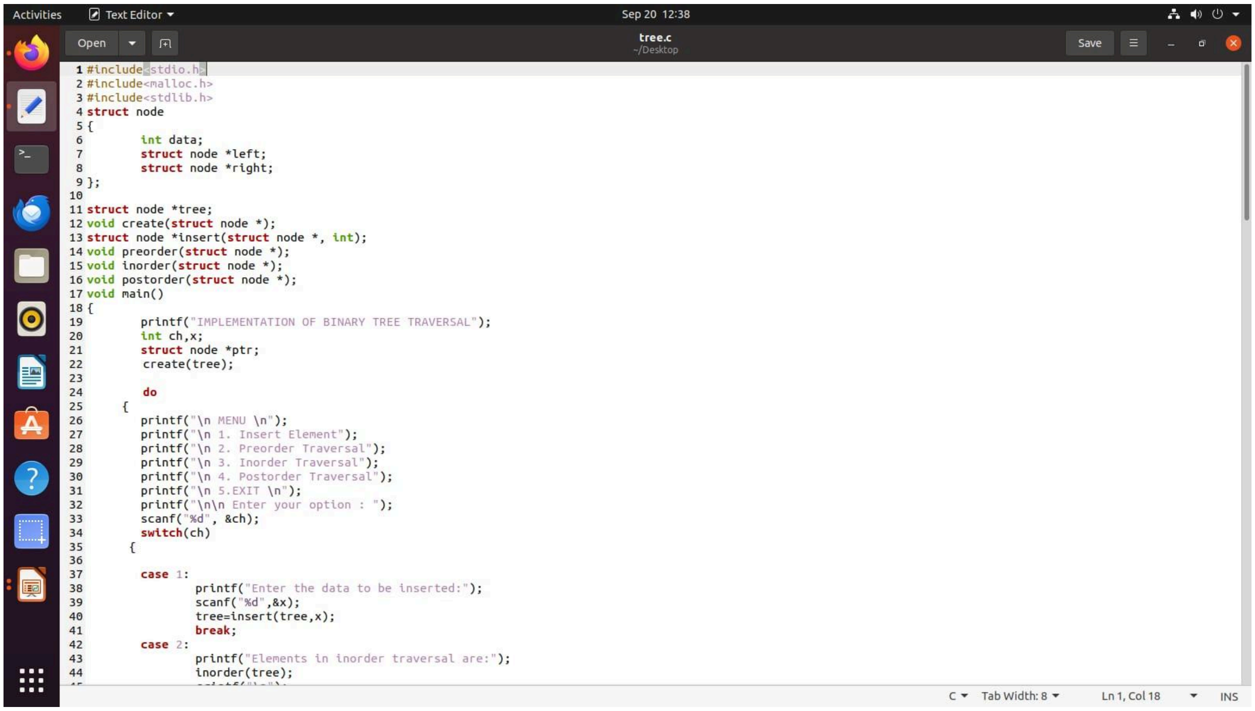
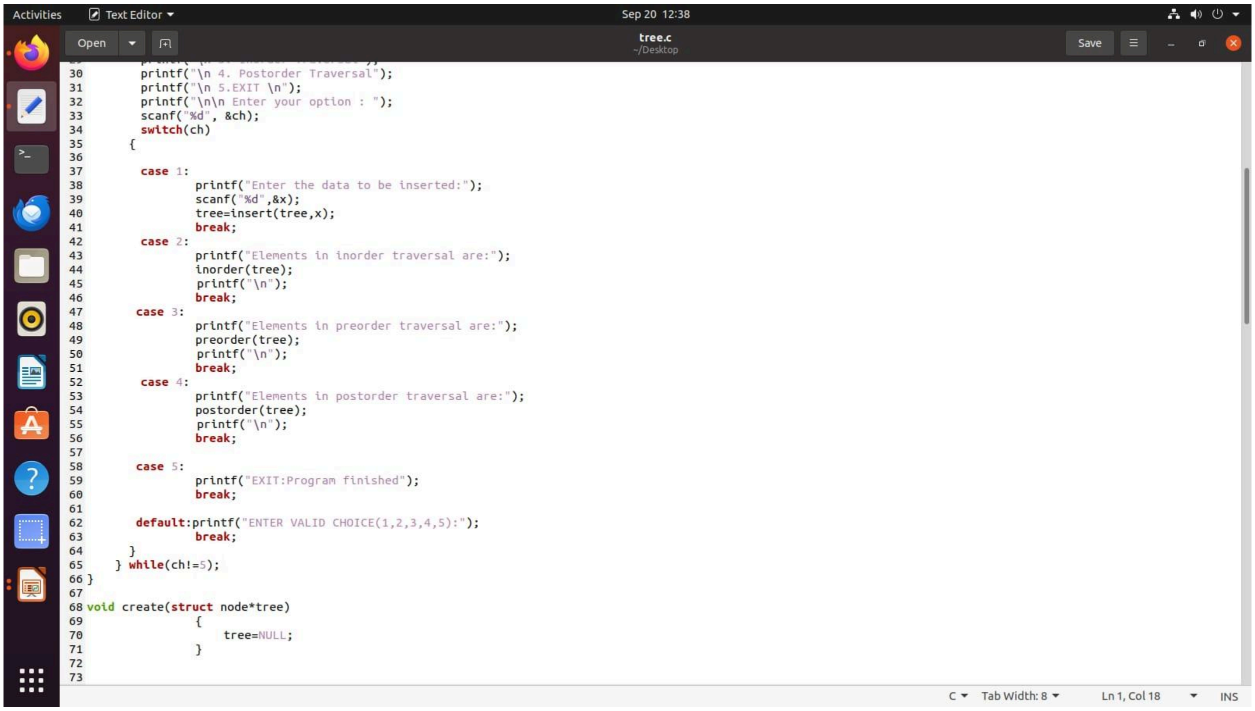
EXPERIMENT NO.06

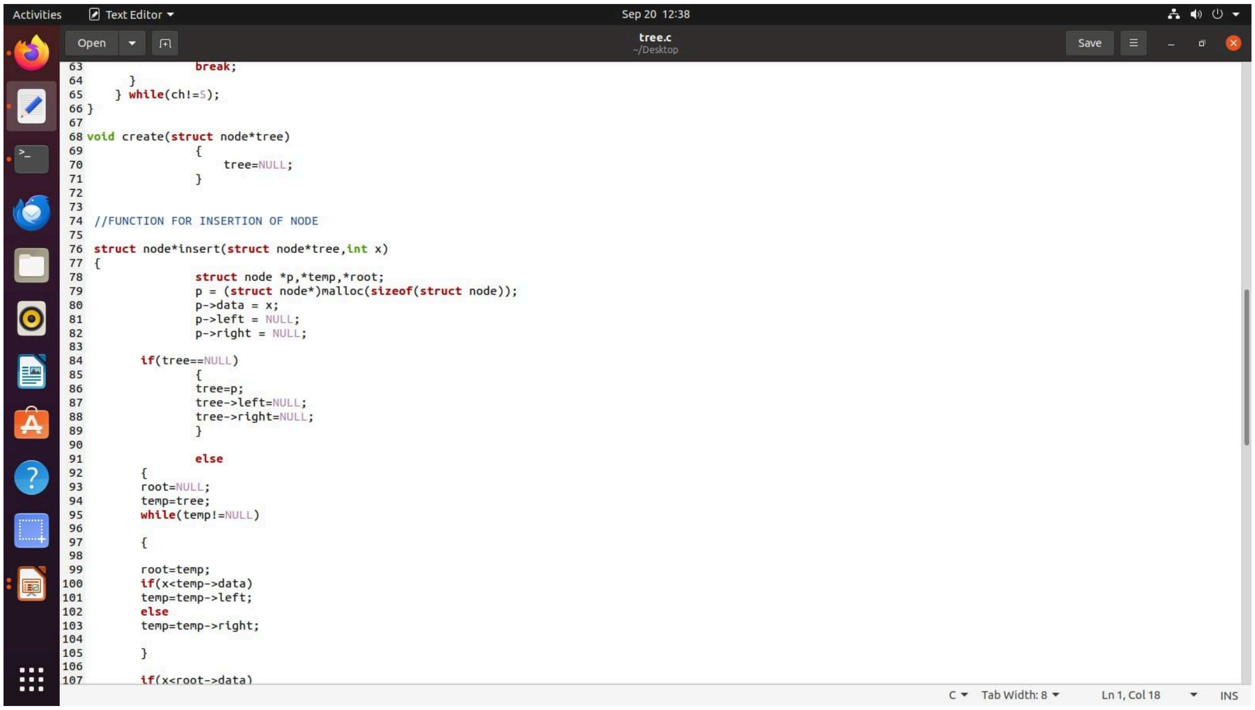
CODE:



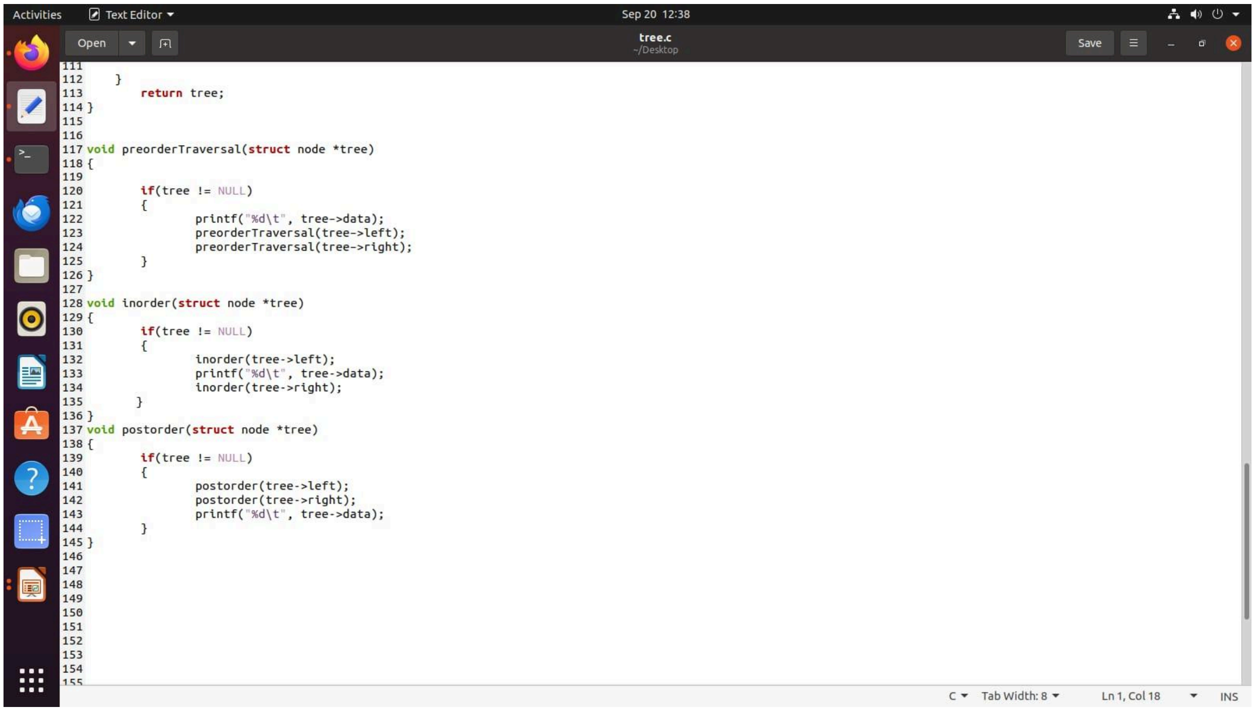
```
#include<stdio.h>
#include<malloc.h>
#include<stdlib.h>
struct node
{
        int data;
        struct node *left;
        struct node *right;
};

struct node *tree;
void create(struct node *);
struct node *insert(struct node *, int);
void preorder(struct node *);
void inorder(struct node *);
void postorder(struct node *);
void main()
{
        printf("IMPLEMENTATION OF BINARY TREE TRAVERSAL");
        int ch,x;
        struct node *ptr;
         create(tree);

         do
     {
        printf("\n MENU \n");
        printf("\n 1. Insert Element");
        printf("\n 2. Preorder Traversal");
        printf("\n 3. Inorder Traversal");
        printf("\n 4. Postorder Traversal");
        printf("\n 5.EXIT \n");
        printf("\n\n Enter your option : ");
        scanf("%d", &ch);
        switch(ch)
      {

        case 1:
                printf("Enter the data to be inserted:");
                scanf("%d",&x);
                tree=insert(tree,x);
                break;
        case 2:
                printf("Elements in inorder traversal are:");
                inorder(tree);
```



```
        printf("\n 4. Postorder Traversal");
        printf("\n 5.EXIT \n");
        printf("\n\n Enter your option : ");
        scanf("%d", &ch);
        switch(ch)
      {

        case 1:
                printf("Enter the data to be inserted:");
                scanf("%d",&x);
                tree=insert(tree,x);
                break;
        case 2:
                printf("Elements in inorder traversal are:");
                inorder(tree);
                 printf("\n");
                break;
       case 3:
                printf("Elements in preorder traversal are:");
                preorder(tree);
                 printf("\n");
                break;
        case 4:
                printf("Elements in postorder traversal are:");
                postorder(tree);
                 printf("\n");
                break;

       case 5:
                printf("EXIT:Program finished");
                break;

       default:printf("ENTER VALID CHOICE(1,2,3,4,5):");
                break;
      }
    } while(ch!=5);
}

void create(struct node*tree)
                {
                    tree=NULL;
                }


```

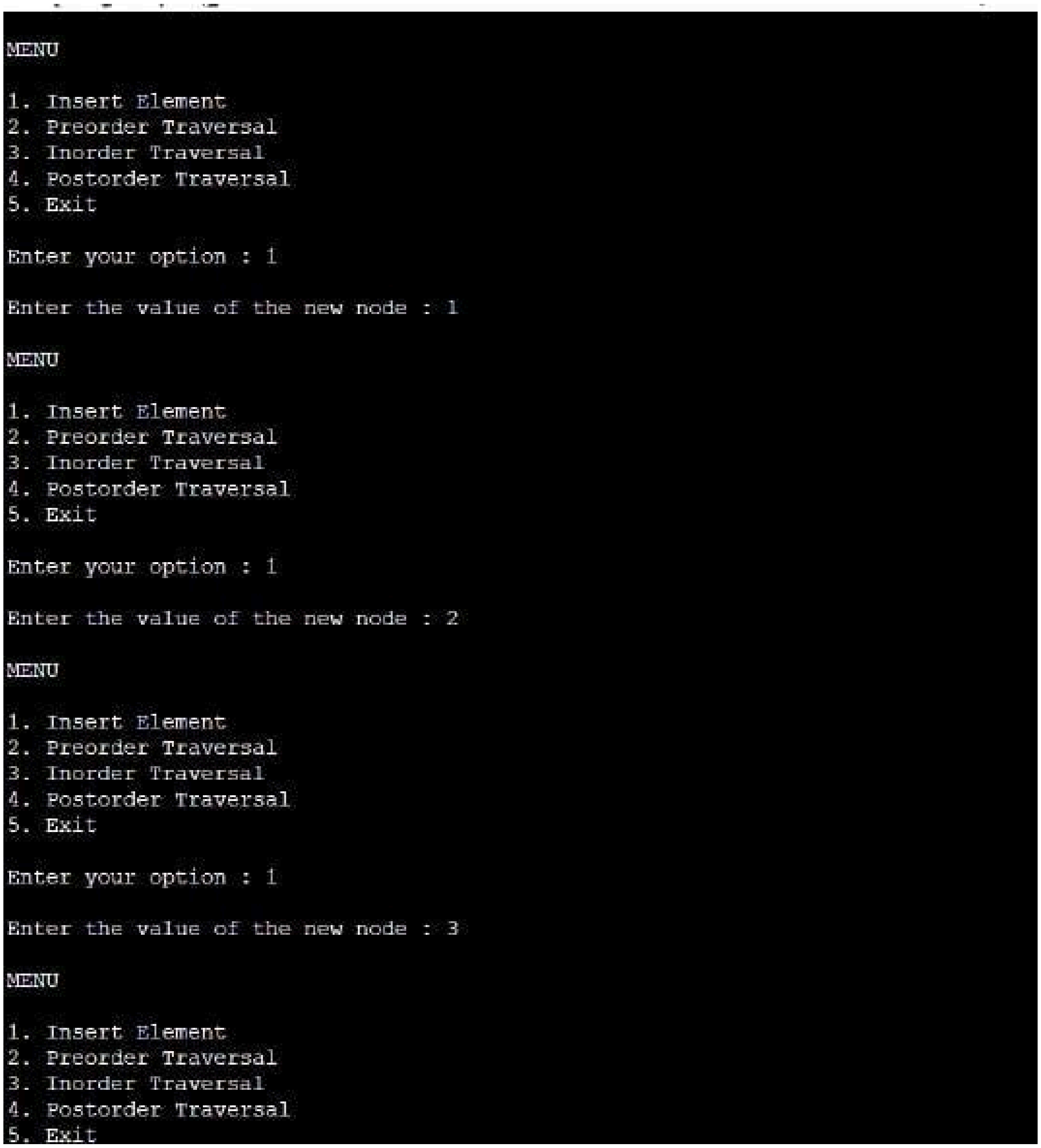


```
                break;
      }
    } while(ch!=5);
}

void create(struct node*tree)
                {
                    tree=NULL;
                }


 //FUNCTION FOR INSERTION OF NODE

 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;
                }

                else
        {
        root=NULL;
        temp=tree;
        while(temp!=NULL)

        {

        root=temp;
        if(x<temp->data)
        temp=temp->left;
        else
        temp=temp->right;

        }

        if(x<root->data)
```

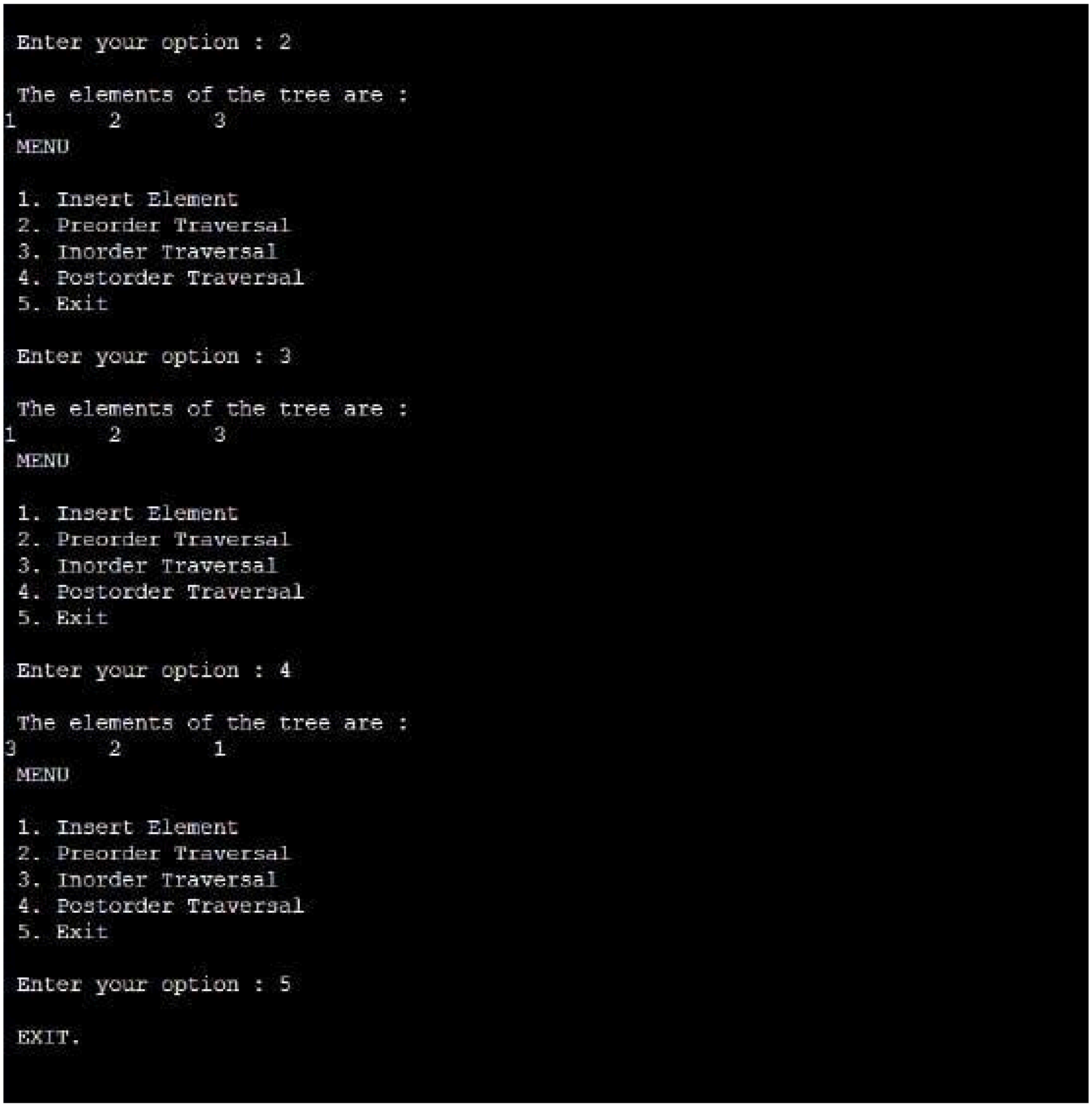


```
    }
        return tree;
}


void preorderTraversal(struct node *tree)
{

        if(tree != NULL)
        {
                printf("%d\t", tree->data);
                preorderTraversal(tree->left);
                preorderTraversal(tree->right);
        }
}

void inorder(struct node *tree)
{
        if(tree != NULL)
        {
                inorder(tree->left);
                printf("%d\t", tree->data);
                inorder(tree->right);
       }
}
void postorder(struct node *tree)
{
        if(tree != NULL)
        {
                postorder(tree->left);
                postorder(tree->right);
                printf("%d\t", tree->data);
        }
}
```

OUTPUT:



```
MENU

1. Insert Element
2. Preorder Traversal
3. Inorder Traversal
4. Postorder Traversal
5. Exit

Enter your option : 1

Enter the value of the new node : 1

MENU

1. Insert Element
2. Preorder Traversal
3. Inorder Traversal
4. Postorder Traversal
5. Exit

Enter your option : 1

Enter the value of the new node : 2

MENU

1. Insert Element
2. Preorder Traversal
3. Inorder Traversal
4. Postorder Traversal
5. Exit

Enter your option : 1

Enter the value of the new node : 3

MENU

1. Insert Element
2. Preorder Traversal
3. Inorder Traversal
4. Postorder Traversal
5. Exit
```



```
 Enter your option : 2

 The elements of the tree are :
1       2       3
 MENU

 1. Insert Element
 2. Preorder Traversal
 3. Inorder Traversal
 4. Postorder Traversal
 5. Exit

 Enter your option : 3

 The elements of the tree are :
1       2       3
 MENU

 1. Insert Element
 2. Preorder Traversal
 3. Inorder Traversal
 4. Postorder Traversal
 5. Exit

 Enter your option : 4

 The elements of the tree are :
3       2       1
 MENU

 1. Insert Element
 2. Preorder Traversal
 3. Inorder Traversal
 4. Postorder Traversal
 5. Exit

 Enter your option : 5

 EXIT.
```

SUBMITTED BY: SHAMAL BHANUDAS DEORE
CLASS/DIV:SY-IT-A
ROLL NO:18