//Implementation of binary tree with traversals

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

struct node
{

        int data;
        struct node *left,*right;
};
struct node *root=NULL;
int level=-1;

void create()
{
                if(root==NULL)
                {
                struct node *temp = (struct node*)malloc(sizeof(struct node));
                int value;
                printf("Enter a value : ");
                scanf("%d",&value);
                temp->data = value;
                temp->left = NULL;
                temp->right = NULL;
                root = temp;
                level = 0;
                }
                else
                {
                printf("Root already exists");
                }
}

void Insert()
{
        if(root==NULL){
                printf("Root is NULL");
                printf("Create the tree to insert elements.");
                create();
        }
        else{
                struct node *temp = (struct node*)malloc(sizeof(struct node));
                int value;
                printf("Enter any value : ");
                scanf("%d",&value);
                temp->data = value;
                temp->left = NULL;
                temp->right = NULL;

                if(root->left == NULL || root->right == NULL)
                {
                        if(root->left == NULL){
                                root->left = temp;
                        }
                        else if(root->right == NULL){
                                root->right = temp;
                        }

                        level = 1;
                }

                else if(level ==1 || level == 2)
                {
                        if((root->left)->left == NULL){
                                (root->left)->left = temp;
                        }

                        else if((root->left)->right == NULL){
                                (root->left)->right = temp;
                        }

                        else if((root->right)->left == NULL){
                                (root->right)->left = temp;
                        }

                        else if((root->right)->right == NULL){
                                (root->right)->right = temp;
                        }

                        level = 2;
                }
        }
}

void preorder(struct node *temp)
{
        if(temp!=NULL)
        {
                printf("%d ",temp->data);

                if(temp->left)
                        preorder(temp->left);

                if(temp->right)
                        preorder(temp->right);
        }
        else{
                printf("Cannot display");
                return;
        }
}

void inorder(struct node *temp)
{
        if(temp!=NULL)
        {
                if(temp->left)
                        inorder(temp->left);
        printf("%d ",temp->data);

        if(temp->right)
            inorder(temp->right);
    }
    else{
        printf("Cannot display");
        return;
    }
}

void postorder(struct node *temp)
{
    if(temp!=NULL)
    {
        if(temp->left)
            postorder(temp->left);

        if(temp->right)
            postorder(temp->right);

        printf("%d ",temp->data);
    }
    else{
        printf("Cannot display");
        return;
    }
}

int main()
{
    int ch,dis;
    while(1)
    {
        printf("\n1.Create\n2.Insert\n3.Display\n0.EXIT\n");
        printf("Enter your choice : ");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1: create(); break;
            case 2: Insert(); break;
            case 3: printf("1.Preorder\n2.Inorder\n3.Postorder\n");
                        printf("Enter your choice : ");
                        scanf("%d",&dis);
                        switch(dis)
                        {
                            case 1: preorder(root); break;
                            case 2: inorder(root); break;
                            case 3: postorder(root); break;
                            default : printf("Choose the correct option.");
break;
                        }
                        break;
            case 0: return 0;
            default : printf("Choose the correct option."); break;
```

```
                }
        }
}
```