

Aim:

Write a recursive C program for traversing a binary tree in preorder, inorder and postorder.

Source Code:**binaryTree.c**

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *left;
    struct node *right;
};
struct node *root=NULL;
void inorder(struct node *temp)
{
    if(temp)
    {
        inorder(temp->left);
        printf("%d->",temp->data);
        inorder(temp->right);
    }
}
void preorder(struct node *temp)
{
    if(temp)
    {
        printf("%d->",temp->data);
        preorder(temp->left);
        preorder(temp->right);
    }
}
void postorder(struct node *temp)
{
    if(temp)
    {
        postorder(temp->left);
        postorder(temp->right);
        printf("%d->",temp->data);
    }
}
void create()
{
    root=NULL;
    insert();
}
struct node *createnode()
{
    struct node *r;
    r=(struct node*)malloc(sizeof(struct node));
```

```

    return r;
}
void insert()
{
    struct node *temp,*r;
    r=createnode();
    printf("Enter the data: ");
    scanf("%d",&r->data);
    r->left=NULL;
    r->right=NULL;
    if(root==NULL)
    {
        root=r;
    }
    else
    {
        temp=root;
        while(temp!=NULL)
        {
            if(temp->data>r->data)
            {
                if(temp->left==NULL)
                {
                    temp->left=r;
                    temp=temp->left;
                }
                temp=temp->left;
            }
            else
            {
                if(temp->right==NULL)
                {
                    temp->right=r;
                    temp=temp->right;
                }
                temp=temp->right;
            }
        }
    }
}
int main()
{
    root=NULL;
    int x,choice;
    do{
        printf("0.create\n1.insert\n2.preorder\n3.postorder\n4.inorder\n5.exit\n");
        printf("Enter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 0:
            {
                create();
                break;
            }
            case 1:

```

```

    {
        insert();
        break;
    }
    case 2:
    {
        printf("Display tree in Preorder ");
        preorder(root);
        printf("\n");
        break;
    }
    case 3:
    {
        printf("Display tree in Postorder ");
        postorder(root);
        printf("\n");
        break;
    }
    case 4:
    {
        printf("Display tree in Inorder ");
        inorder(root);
        printf("\n");
        break;
    }
    case 5:
    {
        exit(0);
    }
    default:printf("Enter valid input\n");
}
}while(choice!=5);
return 0;
}

```

Execution Results - All test cases have succeeded!

| Test Case - 1 |
|----------------------|
| User Output |
| 0.create 0 |
| 1.insert 0 |
| 2.preorder 0 |
| 3.postorder 0 |
| 4.inorder 0 |
| 5.exit 0 |
| Enter your choice: 0 |
| Enter the data: 25 |
| 0.create 1 |
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |

| |
|--|
| Enter your choice: 1 |
| Enter the data: 245 |
| 0.create 0 |
| 1.insert 0 |
| 2.preorder 0 |
| 3.postorder 0 |
| 4.inorder 0 |
| 5.exit 0 |
| Enter your choice: 0 |
| Enter the data: 345 |
| 0.create 1 |
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |
| Enter your choice: 1 |
| Enter the data: 36 |
| 0.create 1 |
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |
| Enter your choice: 1 |
| Enter the data: 589 |
| 0.create 2 |
| 1.insert 2 |
| 2.preorder 2 |
| 3.postorder 2 |
| 4.inorder 2 |
| 5.exit 2 |
| Enter your choice: 2 |
| Display tree in Preorder 345->36->589-> 3 |
| 0.create 3 |
| 1.insert 3 |
| 2.preorder 3 |
| 3.postorder 3 |
| 4.inorder 3 |
| 5.exit 3 |
| Enter your choice: 3 |
| Display tree in Postorder 36->589->345-> 4 |
| 0.create 4 |
| 1.insert 4 |
| 2.preorder 4 |
| 3.postorder 4 |
| 4.inorder 4 |
| 5.exit 4 |
| Enter your choice: 4 |
| Display tree in Inorder 36->345->589-> 5 |
| 0.create 5 |

| |
|----------------------|
| 1.insert 5 |
| 2.preorder 5 |
| 3.postorder 5 |
| 4.inorder 5 |
| 5.exit 5 |
| Enter your choice: 5 |

| Test Case - 2 |
|----------------------|
| User Output |
| 0.create 0 |
| 1.insert 0 |
| 2.preorder 0 |
| 3.postorder 0 |
| 4.inorder 0 |
| 5.exit 0 |
| Enter your choice: 0 |
| Enter the data: 21 |
| 0.create 0 |
| 1.insert 0 |
| 2.preorder 0 |
| 3.postorder 0 |
| 4.inorder 0 |
| 5.exit 0 |
| Enter your choice: 0 |
| Enter the data: 325 |
| 0.create 1 |
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |
| Enter your choice: 1 |
| Enter the data: 586 |
| 0.create 0 |
| 1.insert 0 |
| 2.preorder 0 |
| 3.postorder 0 |
| 4.inorder 0 |
| 5.exit 0 |
| Enter your choice: 0 |
| Enter the data: 26 |
| 0.create 1 |
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |
| Enter your choice: 1 |
| Enter the data: 478 |
| 0.create 1 |

| |
|--|
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |
| Enter your choice: 1 |
| Enter the data: 213 |
| 0.create 1 |
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |
| Enter your choice: 1 |
| Enter the data: 36 |
| 0.create 1 |
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |
| Enter your choice: 1 |
| Enter the data: 21 |
| 0.create 1 |
| 1.insert 1 |
| 2.preorder 1 |
| 3.postorder 1 |
| 4.inorder 1 |
| 5.exit 1 |
| Enter your choice: 1 |
| Enter the data: 2245 |
| 0.create 2 |
| 1.insert 2 |
| 2.preorder 2 |
| 3.postorder 2 |
| 4.inorder 2 |
| 5.exit 2 |
| Enter your choice: 2 |
| Display tree in Preorder 26->21->478->213->36->2245-> 3 |
| 0.create 3 |
| 1.insert 3 |
| 2.preorder 3 |
| 3.postorder 3 |
| 4.inorder 3 |
| 5.exit 3 |
| Enter your choice: 3 |
| Display tree in Postorder 21->36->213->2245->478->26-> 4 |
| 0.create 4 |
| 1.insert 4 |
| 2.preorder 4 |
| 3.postorder 4 |

| |
|--|
| 4.inorder 4 |
| 5.exit 4 |
| Enter your choice: 4 |
| Display tree in Inorder 21->26->36->213->478->2245-> 5 |
| 0.create 5 |
| 1.insert 5 |
| 2.preorder 5 |
| 3.postorder 5 |
| 4.inorder 5 |
| 5.exit 5 |
| Enter your choice: 5 |