



```
if (node == NULL )
return new_node(value);

else
{
    if (value < node->data)
        node->left = insert(node->left, value);
    else if (value > node->data)
        node->right = insert(node->right, value);
}
return node;
```

### **Inorder traversal**

First, visit all the nodes in the left subtree

Then the root node

Visit all the nodes in the right subtree

### **Preorder traversal**

Visit root node

Visit all the nodes in the left subtree

Visit all the nodes in the right subtree

### **Postorder traversal**

Visit all the nodes in the left subtree

Visit all the nodes in the right subtree

Visit the root node

## Source Code

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *left, *right;
};
struct node *new_node(int n)
{
    struct node *temp;
    temp = (struct node *)malloc(sizeof(struct node));
    temp->data = n;
    temp->left = temp->right = NULL;
    return temp;
}
struct node *insert(struct node *node, int value)
{
    if (node == NULL )
        return new_node(value);

    else
    {
        if (value < node->data)
            node->left = insert(node->left, value);
        else if (value > node->data)
            node->right = insert(node->right, value);
    }
    return node;
}

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

```

    }
}
void preorder(struct node *root)
{
    if (root != NULL)
    {
        printf("%d -> ", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}
void postorder(struct node *root)
{
    if (root != NULL)
    {
        postorder(root->left);
        postorder(root->right);
        printf("%d -> ", root->data);
    }
}

int main()
{
    struct node *root = NULL;
    int n,i;
    printf("\n Enter the Egede Number : ");
    scanf("%d",&n);
    int a[n];
    for (i = 0; i < n; i++)
    {
        printf("User Input Number %d : ",i +1);
        scanf("%d", &a[i]);
    }
    for (i = 0; i < n; i++)
        root = insert(root, a[i]);
    while (1)
    {
        int n;

        printf("\n 1. Preorder ");
        printf("\n 2. Inorder ");
        printf("\n 3. Postorder ");
        printf("\n 4. Exit ");

```

```

printf("\nEnter You Want To Choice: ");
scanf("%d", &n);
    switch (n)
    {
        case 1:
            printf("\n\n");
            printf("\nPreorder Traversal : ");
            preorder(root);
            break;
        case 2:
            printf("\n\n");
            printf("\nInorder Traversal : ");
            inorder(root);
            break;
        case 3:
            printf("\n\n");
            printf("\nPostorder Traversal :");
            postorder(root);
            break;
        case 0:
            exit(0);
            break;
        default:
            printf("\nWrrong Choice Try Again ");
            break;
    }
}
return 0;
}

```

Output

```

13     return temp;
14 }
15 struct node *insert(struct node *node, int value)
16 {
17     if (node == NULL )
18         return new_node(value);
19
20     else
21     {
22         if (value < node->data)
23             node->left = insert(node->left, value);
24         else if (value > node->data)
25             node->right = insert(node->right, value);
26     }
27     return node;
28 }
29
30 void inorder(struct node *root)
31 {
32     if(root != NULL)
33     {
34         inorder(root->left);
35         printf("%d ->", root->data);
36         inorder(root->right);
37     }
38 }
39 void preorder(struct node *root)
40 {
41     if (root != NULL)
42     {
43         printf("%d -> ", root->data);
44         preorder(root->left);
45         preorder(root->right);
46     }
47 }
48 void postorder(struct node *root)
49 {
50
51     if (root != NULL)
52     {
53         postorder(root->left);
54         postorder(root->right);
55         printf("%d ->", root->data);
56     }
57 }
58
59 int main()

```

D:\Coding in C\CSE 106\Lab report 6.exe

Enter the Egede Number : 4  
 User Input Number 1 : 12  
 User Input Number 2 : 3  
 User Input Number 3 : 4  
 User Input Number 4 : 2

1. Preorder
2. Inorder
3. Postorder
4. Exit

Enter You Want To Choice: 1

Preorder Traversal : 12 -> 3 -> 2 ->

1. Preorder
2. Inorder
3. Postorder
4. Exit

Enter You Want To Choice: 2

Inorder Traversal : 2 ->3 ->4 ->12 -

1. Preorder
2. Inorder
3. Postorder
4. Exit

Enter You Want To Choice: 3

Postorder Traversal :2 ->4 ->3 ->12

1. Preorder
2. Inorder
3. Postorder
4. Exit

Enter You Want To Choice: 4