



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

COURSE CODE: DJS22ITL302

DATE:

COURSE NAME: Data Structure Laboratory

CLASS: II-Batch1

NAME: Anish Sharma

SAP ID: 80003220045

ROLL NO.: 1011

Experiment No. 07

CO/LO: CO1

Aim: Perform Different tree traversals for binary tree.

Theory: In-order Traversal:

The typical pattern is left \rightarrow Root \rightarrow right.
Useful for BST to retrieve elements in sorted order.

Preorder Traversal:

The typical pattern is Root \rightarrow Left \rightarrow Right.
Useful in creating a prefix expression for expression trees.

Post-order Traversal:

The typical pattern is Left \rightarrow Right \rightarrow Root.
Useful in deleting nodes from a tree.

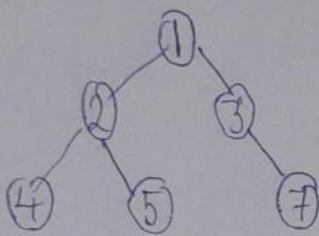


Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

Diagram:



Inorder : 4, 2, 5, 1, 3, 7

Preorder : 1, 2, 4, 5, 3, 7

Postorder : 4, 5, 2, 7, 3, 1

Conclusion :

Hence we studied and implemented different trees traversals for binary tree.

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  struct node
5  {
6      int data;
7      struct node *left;
8      struct node *right;
9  };
10
11 struct node *root = NULL;
12
13 struct node *create(int value)
14 {
15     struct node *newnode = (struct node *)malloc(sizeof(struct node *));
16     newnode->data = value;
17     newnode->left = newnode->right = NULL;
18     return newnode;
19 }
20
21 struct node *insert(struct node *root, int value)
22 {
23     if(root==NULL)
24         return create(value);
25
26     if(root->left==NULL)
27     {
28         root->left = insert(root->left, value);
29     }
30     else
31     {
32         root->right = insert(root->right, value);
33     }
34 }
35
36 return root;
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
49 void inorder(struct node *root)
50 {
51     if(root!=NULL)
52     {
53         inorder(root->left);
54         printf("%d ",root->data);
55         inorder(root->right);
56     }
57 }
58
59 void postorder(struct node *root)
60 {
61     if(root!=NULL)
62     {
63         postorder(root->left);
64         postorder(root->right);
65         printf("%d ",root->data);
66     }
67 }
68
69 int main()
70 {
71     printf("This is a Binary Tree\n");
72     int ch, val, rt;
73     printf("Enter the value of the root : ");
74     scanf("%d",&rt);
75     root = insert(root,rt);
76     do
77     {
78         printf("1.Insert\n2.Pre Order\n3.In Order\n4.Post Order\n");
79         printf("Enter your choice : ");
80         scanf("%d",&ch);
81
82         switch(ch)
83         {
84             case 1:
85                 printf("Enter the value to be inserted : ");
86                 scanf("%d",&val);
87                 insert(root,val);
88                 break;
89             case 2:
90                 printf("The elements in the tree are : ");
91                 preorder(root);
92                 break;
93             case 3:
94                 printf("The elements in the tree are : ");
95                 inorder(root);
96                 break;
97             case 4:
98                 printf("The elements in the tree are : ");
99                 postorder(root);
100                break;
101        }
102    }while(ch!=5);
103    return 0;
104 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\C\DSA> cd "e:\C\DSA\" ; if ($?) { gcc pracBT.c -o pracBT } ;  
○ This is a Binary Tree  
Enter the value of the root : 12  
1.Insert  
2.Pre Order  
3.In Order  
4.Post Order  
Enter your choice : 1  
Enter the value to be inserted : 45
```

```
1.Insert  
2.Pre Order  
3.In Order  
4.Post Order  
Enter your choice : 2  
The elements in the tree are : 45 12 54 37 28 63
```

```
1.Insert  
2.Pre Order  
3.In Order  
4.Post Order  
Enter your choice : 3  
The elements in the tree are : 12 45 37 54 63 28  
1.Insert  
2.Pre Order  
3.In Order  
4.Post Order  
Enter your choice : 4  
The elements in the tree are : 12 37 63 28 54 45
```