

EXPERIMENT NO.06

CODE:

```
Activities Text Editor Sep 20 12:38 tree.c ~/Desktop Save
1 #include<stdio.h>
2 #include<malloc.h>
3 #include<stdlib.h>
4 struct node
5 {
6     int data;
7     struct node *left;
8     struct node *right;
9 };
10
11 struct node *tree;
12 void create(struct node *);
13 struct node *insert(struct node *, int);
14 void preorder(struct node *);
15 void inorder(struct node *);
16 void postorder(struct node *);
17 void main()
18 {
19     printf("IMPLEMENTATION OF BINARY TREE TRAVERSAL");
20     int ch,x;
21     struct node *ptr;
22     create(tree);
23
24     do
25     {
26         printf("\n MENU \n");
27         printf("\n 1. Insert Element");
28         printf("\n 2. Preorder Traversal");
29         printf("\n 3. Inorder Traversal");
30         printf("\n 4. Postorder Traversal");
31         printf("\n 5.EXIT \n");
32         printf("\n Enter your option : ");
33         scanf("%d", &ch);
34         switch(ch)
35         {
36
37             case 1:
38                 printf("Enter the data to be inserted:");
39                 scanf("%d",&x);
40                 tree=insert(tree,x);
41                 break;
42
43             case 2:
44                 printf("Elements in inorder traversal are:");
45                 inorder(tree);
46                 break;
47
48             case 3:
49                 printf("Elements in preorder traversal are:");
50                 preorder(tree);
51                 printf("\n");
52                 break;
53
54             case 4:
55                 printf("Elements in postorder traversal are:");
56                 postorder(tree);
57                 printf("\n");
58                 break;
59
60             case 5:
61                 printf("EXIT:Program finished");
62                 break;
63
64             default:printf("ENTER VALID CHOICE(1,2,3,4,5):");
65                 break;
66         }
67     } while(ch!=5);
68 }
69 void create(struct node*tree)
70 {
71     tree=NULL;
72 }
73
```

```
Activities Text Editor Sep 20 12:38 tree.c ~/Desktop Save
30 printf("\n 4. Postorder Traversal");
31 printf("\n 5.EXIT \n");
32 printf("\n Enter your option : ");
33 scanf("%d", &ch);
34 switch(ch)
35 {
36
37     case 1:
38         printf("Enter the data to be inserted:");
39         scanf("%d",&x);
40         tree=insert(tree,x);
41         break;
42
43     case 2:
44         printf("Elements in inorder traversal are:");
45         inorder(tree);
46         printf("\n");
47         break;
48
49     case 3:
50         printf("Elements in preorder traversal are:");
51         preorder(tree);
52         printf("\n");
53         break;
54
55     case 4:
56         printf("Elements in postorder traversal are:");
57         postorder(tree);
58         printf("\n");
59         break;
60
61     case 5:
62         printf("EXIT:Program finished");
63         break;
64
65     default:printf("ENTER VALID CHOICE(1,2,3,4,5):");
66         break;
67 }
68 } while(ch!=5);
69 }
70 void create(struct node*tree)
71 {
72     tree=NULL;
73 }
```

```
Activities Text Editor Sep 20 12:38 tree.c ~/Desktop Save
63         break;
64     }
65     } while(ch!=5);
66 }
67
68 void create(struct node*tree)
69 {
70     tree=NULL;
71 }
72
73
74 //FUNCTION FOR INSERTION OF NODE
75
76 struct node*insert(struct node*tree,int x)
77 {
78     struct node *p,*temp,*root;
79     p = (struct node*)malloc(sizeof(struct node));
80     p->data = x;
81     p->left = NULL;
82     p->right = NULL;
83
84     if(tree==NULL)
85     {
86         tree=p;
87         tree->left=NULL;
88         tree->right=NULL;
89     }
90     else
91     {
92         root=NULL;
93         temp=tree;
94         while(temp!=NULL)
95         {
96             {
97                 root=temp;
98                 if(x<temp->data)
99                     temp=temp->left;
100                 else
101                     temp=temp->right;
102             }
103         }
104         if(x<root->data)
```

```
Activities Text Editor Sep 20 12:38 tree.c ~/Desktop Save
111     }
112     return tree;
113 }
114 }
115
116
117 void preorderTraversal(struct node *tree)
118 {
119     if(tree != NULL)
120     {
121         printf("%d\t", tree->data);
122         preorderTraversal(tree->left);
123         preorderTraversal(tree->right);
124     }
125 }
126
127
128 void inorder(struct node *tree)
129 {
130     if(tree != NULL)
131     {
132         inorder(tree->left);
133         printf("%d\t", tree->data);
134         inorder(tree->right);
135     }
136 }
137 void postorder(struct node *tree)
138 {
139     if(tree != NULL)
140     {
141         postorder(tree->left);
142         postorder(tree->right);
143         printf("%d\t", tree->data);
144     }
145 }
146
147
148
149
150
151
152
153
154
155
```

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