

DSA EXP NO. 6

CODE:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <malloc.h>
4 struct node {
5     int data;
6     struct node *left;
7     struct node *right;
8 };
9
10 struct node *tree;
11 void create(struct node *);
12 struct node *insert(struct node*, int);
13 void inorder(struct node *);
14 void preorder(struct node *);
15 void postorder(struct node *);
16
17 void main() {
18     printf("\n Binary Tree Traversals \n");
19     int choice, x;
20     struct node *ptr;
21     create(tree);
22     do {
23         printf("\n **Operations Available**");
24         printf("\n 1. Insert a Node");
25         printf("\n 2. Display Inorder Traversal"); printf("\n 3. Display Preorder Traversal");
26         printf("\n 4. Display Postorder Traversal");
27         printf("\n 5. Exit \n");
28         printf(" Please enter your choice: ");
29         scanf("%d", &choice);
30
31         switch (choice) {
32             case 1:
33                 printf("\n Enter no of data to be inserted : ");
34                 int b;
35                 scanf("%d", &b);
36                 for (int a = 0; a < b ;a++){
37                     printf("\n Enter data no. %d : ", a+1);
38                     scanf("%d", &x);
39                     tree = insert(tree, x);
40                 }
41                 break;
```

```

42
43     case 2:
44         printf("\n Elements in the inorder traversals are : ");
45         inorder(tree);
46         printf("\n");
47         break;
48
49     case 3:
50         printf("\n Elements in the preorder traversals are : ");
51         preorder(tree);
52         printf("\n");
53         break;
54
55     case 4:
56         printf("\n Elements in the postorder traversals are : ");
57         postorder(tree);
58         printf("\n");
59         break;
60
61     case 5:
62         printf("Exit: Program Finished !!");
63         break;
64
65     default:
66         printf("\n Please enter a valid option 1, 2, 3, 4, 5.");
67         break;
68     }
69 } while(choice!=5);
70 }
71
72
73 void create(struct node *tree) {
74     tree = NULL;
75 }
76 // inserting a new node
77 struct node *insert(struct node *tree, int x) {
78
79     struct node *p, *temp, *root;
80     p = (struct node *)malloc(sizeof(struct node));
81     p->data = x;
82     p->left = NULL;

```

```

83     p->right = NULL;
84
85     if (tree == NULL) {
86         tree = p;
87         tree->left = NULL;
88         tree->right = NULL;
89     }
90     else {
91         root = NULL;
92         temp = tree;
93
94         while (temp != NULL) {
95             root = temp;
96             if (x < temp->data) {
97                 temp = temp->left;
98             }
99             else{
100                 temp = temp->right;
101             }
102         }
103
104         if(x<root->data) {
105             root->left = p;
106         }
107         else{
108             root->right = p;
109         }
110     }
111     return tree;
112 }
113
114
115 // Inorder Traversals
116 void inorder (struct node *tree) {
117
118     if (tree != NULL) {
119         inorder (tree->left);
120         printf("%d \t", tree->data);
121         inorder (tree->right);
122     }
123 }

```

```

124 // Preorder Traversals
125 void preorder(struct node *tree) {
126
127     if (tree != NULL) {
128         printf("%d \t", tree->data);
129         preorder(tree->left);
130         preorder(tree->right);
131     }
132 }
133 // Postorder Traversals
134 void postorder (struct node *tree) {
135
136     if (tree != NULL) {
137         postorder(tree->left);
138         postorder(tree->right);
139         printf("%d \t", tree->data);
140     }
141 }

```

OUTPUT :

```
Output
Binary Tree Traversals

**Opertaions Available**
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Exit
Please enter your choice: 1
Enter no of data to be inserted : 4
Enter data no. 1 : 33
Enter data no. 2 : 25
Enter data no. 3 : 89
Enter data no. 4 : 43
**Opertaions Available**
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Exit
Please enter your choice: 2
Elements in the inorder traversals are : 25 33 43 89

**Opertaions Available**
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Exit
Please enter your choice: 3
Elements in the preorder traversals are : 33 25 89 43

**Opertaions Available**
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Exit
Please enter your choice: 4
Elements in the postorder traversals are : 25 43 89 33
```