

**exp\_10/a.c**

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node {
5     int data;
6     struct Node* left;
7     struct Node* right;
8 };
9
10 struct Node* createNode(int data) {
11     struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
12     newNode->data = data;
13     newNode->left = newNode->right = NULL;
14     return newNode;
15 }
16
17 void preorder(struct Node* root) {
18     if (root == NULL) return;
19     printf("%d ", root->data);
20     preorder(root->left);
21     preorder(root->right);
22 }
23
24 void inorder(struct Node* root) {
25     if (root == NULL) return;
26     inorder(root->left);
27     printf("%d ", root->data);
28     inorder(root->right);
29 }
30
31 void postorder(struct Node* root) {
32     if (root == NULL) return;
33     postorder(root->left);
34     postorder(root->right);
35     printf("%d ", root->data);
36 }
37
38 struct Node* insert(struct Node* root, int data) {
39     if (root == NULL) return createNode(data);
40     char dir;
41     printf("Insert %d to left or right of %d? (l/r): ", data, root->data);
42     scanf(" %c", &dir);
43     if (dir == 'l')
44         root->left = insert(root->left, data);
45     else if (dir == 'r')
46         root->right = insert(root->right, data);
47     else
48         printf("Invalid choice. Try again.\n");
```

```
49     return root;
50 }
51
52 int main() {
53     struct Node* root = NULL;
54     int n, val;
55     printf("Enter number of nodes: ");
56     scanf("%d", &n);
57     if (n > 0) {
58         printf("Enter root node value: ");
59         scanf("%d", &val);
60         root = insert(root, val);
61         for (int i = 1; i < n; i++) {
62             printf("Enter value of node %d: ", i + 1);
63             scanf("%d", &val);
64             root = insert(root, val);
65         }
66     }
67
68     printf("Preorder traversal: ");
69     preorder(root);
70     printf("\nInorder traversal: ");
71     inorder(root);
72     printf("\nPostorder traversal: ");
73     postorder(root);
74     printf("\n");
75     return 0;
76 }
77 }
```