

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  struct node
5  {
6      int data;
7      struct node *right;
8      struct node *left;
9  };
10
11 struct node* create_node(int data)
12 {
13     struct node *node=(struct node *)malloc(sizeof(struct node));
14     node->data=data;
15     node->right=NULL;
16     node->left=NULL;
17     return node;
18 }
19
20 void pre_order(struct node *root)
21 {
22     if(root!=NULL)
23     {
24         printf("\t%d",root->data);
25         pre_order(root->left);
26         pre_order(root->right);
27     }
28 }
29 void post_order(struct node *root)
30 {
31     if(root!=NULL)
32     {
33         post_order(root->left);
34         post_order(root->right);
35         printf("\t%d",root->data);
36     }
37 }
38
39 void in_order(struct node *root)
40 {
41     if(root!=NULL)
42     {
43         in_order(root->left);
44         printf("\t%d",root->data);
45         in_order(root->right);
46     }
47 }
48
49
50
51 void main()
52 {
53     struct node *root;
54     root=create_node(20);
55     root->left=create_node(30);
56     root->right=create_node(40);
57     root->left->left=create_node(10);
58     root->left->right=create_node(12);
59     root->right->left=create_node(45);
60     printf("PRE ORDER \n");
61     pre_order(root);
62     printf("\n\nPOST ORDER \n");
63     post_order(root);
64     printf("\n\nIN ORDER \n");
65     in_order(root);
66 }

```