

```
struct node  
{ int ele;  
  struct node *llink, *rlink;  
}; typedef struct node *Node;
```

```
Node getnode(int item)
```

```
{ Node temp = (Node) malloc(sizeof(struct node));  
  temp->ele = item;  
  temp->llink = temp->rlink = NULL;  
  return temp;  
}
```

```
Node insert(Node node, int item)
```

```
{ if (node == NULL)  
  { return getnode(item); }  
  if (item < node->ele)  
    node->llink = insert(node->llink, item);  
  else  
    node->rlink = insert(node->rlink, item);  
  return node;  
}
```

```
void preorder(Node root)
```

```
{ if (root == NULL)  
  return;  
  printf("%d -> ", root->ele);  
  preorder(root->llink);  
  preorder(root->rlink);  
}
```

```
void inorder (Node root)
{
    if (root == NULL)
        return;

    inorder (root -> llink);
    printf (" %d -> ", root -> ele);
    inorder (root -> rlink);
}
```

```
void postorder (Node root)
{
    if (root == NULL)
        return;

    postorder (root -> llink);
    postorder (root -> rlink);
    printf (" %d -> ", root -> ele);
}
```

```
void display (Node root, int i)
{
    int j;
    if (root != NULL)
    {
        display (root -> rlink, i+1);
        for (j=0; j<i; j++)
            printf (" ");
        printf (" %d \n", root -> ele);
        display (root -> llink, i+1);
    }
}
```

```
int main()
{
    Node root = NULL;
    int choice, item;
    for (;;)
    {
        printf (" 1. Insert 2. Preorder 3. Inorder 4. Postorder\n5. Display 6. Exit \n");
    }
}
```

```
printf("Enter choice : ");  
scanf("%d", &choice);  
switch(choice)
```

```
{ case 1: printf("Enter item to be inserted : ");  
        scanf("%d", &item);  
        root = insert(root, item);  
        break;
```

```
case 2: printf("Preorder traversal : ");  
        preorder(root);  
        break;
```

```
case 3: printf("Inorder traversal : ");  
        inorder(root);  
        break;
```

```
case 4: printf("Postorder traversal : ");  
        postorder(root);  
        break;
```

```
case 5: display(root, 0);  
        break;
```

```
default: printf("Enter proper instructions (1\n");  
        break;
```

```
}
```

```
}
```

```
return 0;
```

```
}
```