

```
/******
```

```
Done by Abhishek Navlakhi
```

```
For
```

```
Kunal & Abhishek Private Tutions
```

```
www.navlakhi.com
```

```
*****/
```

```
#include <stdio.h>
```

```
#include <alloc.h>
```

```
#include <stdlib.h>
```

```
#include <conio.h>
```

```
int leafcnt=0; /*used to count the number of leaf nodes*/
```

```
struct tree_node
```

```
{
```

```
int data;
```

```
struct tree_node *left;
```

```
struct tree_node *right;
```

```
};
```

```
struct headnode
```

```
{
```

```
int count;
```

```
struct tree_node *head;
```

```
}*tree;
```

```
struct headnode* createTree()
```

```
{
```

```
struct headnode *pNew;
```

```
pNew=(struct headnode*)malloc(sizeof(struct headnode));
```

```
pNew->count=0;
```

```
pNew->head=NULL;
```

```
return pNew;
```

```
}
```

```

void insertNode()
{
struct tree_node *pNew,*pLoc;
if (tree->count==0)
{
tree->head=(struct tree_node*)malloc(sizeof(struct tree_node));
tree->head->left=NULL;
tree->head->right=NULL;
printf("Feed in the root data: ");
scanf("%d",&(tree->head->data));
tree->count=1;
}
else
{
pNew=(struct tree_node*)malloc(sizeof(struct tree_node));
pNew->left=NULL;
pNew->right=NULL;
printf("Feed in the node data: ");
scanf("%d",&pNew->data);
pLoc=tree->head;
while(1)
{
if (pNew->data>pLoc->data)
{
if (pLoc->right!=NULL) pLoc=pLoc->right;
else
{pLoc->right=pNew;break;}
}
if (pNew->data<pLoc->data)
{
if (pLoc->left!=NULL) pLoc=pLoc->left;
else
{pLoc->left=pNew;break;}
}
if (pNew->data==pLoc->data)
{
printf("Data already present...\n");
break;
}
}/*end of while*/
tree->count+=1;
}/*end of count !=0*/
}

```

```

void RecursivePreorderTraverse(struct tree_node *p)
{
if (p!=NULL)
{
    if (p->left==NULL && p->right==NULL) ++leafcnt;
    RecursivePreorderTraverse(p->left);
    RecursivePreorderTraverse(p->right);
}
}

```

```

void main()
{
int delKey;
char choice;

tree=createTree();
do
{
insertNode();
printf("Press 0 to stop any other key to continue adding\n");
choice=getch();
}while(choice!='0');

if (tree->count!=0) RecursivePreorderTraverse(tree->head);
printf("Number of leaf nodes=%d\n",leafcnt);
printf("Number of non-leaf nodes=%d\n",tree->count-leafcnt);
}

```