

```
#include <stdio.h>
#include <alloc.h>
#include <stdlib.h>
#include <conio.h>

int stack_count;
int stack[100];

struct tree_node
{
int data;
struct tree_node *left;
struct tree_node *right;
};

struct headnode
{
int count;
struct tree_node *head;
}*tree;

void createTree()
{
tree=(struct headnode*)malloc(sizeof(struct headnode));
tree->count=0;
tree->head=NULL;
}
```

```
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;}
            }
        }
        tree->count+=1;
    }/*end of count !=0*/
}
```

```
void check(int x)
{
int i;
if (stack_count!=0)
{
    for(i=0;i<stack_count;i++)
    if (x==stack[i]) break;

    if (i< stack_count)
    printf("Duplicate=%d\n",x);
    else stack[stack_count++]=x;
}
else stack[stack_count++]=x;
}

void Duplicates(struct tree_node *p)
{
if (p!=NULL)
{
    check(p->data);

    Duplicates(p->left);
    Duplicates(p->right);
}
}
```

```
void main( )
{
int choice,delKey,i;
createTree();
do
{
    printf("1.Add Node\n");
    printf("2. Find Duplicates\n");
    printf("3.Exit\n");
    printf("Feed in your choice: ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:    insertNode();
                  break;
        case 2:    if (tree->count!=0)
                  {
                      stack_count=0;
                      Duplicates(tree->head);
                  }
                  break;
        case 3:    break;
    }
}while(choice !=3 );
}
```