BST Insert and Search

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

struct Node
{
    struct Node *lchild;
    int data;
    struct Node *rchild;
}*root=NULL;

void Insert(int key)
{
    struct Node *t=root;
    struct Node *r=NULL,*p;

    if(root==NULL)
    {
        p=(struct Node *)malloc(sizeof(struct Node));
        p->data=key;
        p->lchild=p->rchild=NULL;
        root=p;
        return;
    }
    while(t!=NULL)
    {
        r=t;
        if(key<t->data)
            t=t->lchild;
        else if(key>t->data)
            t=t->rchild;
        else
            return;
    }
    p=(struct Node *)malloc(sizeof(struct Node));
    p->data=key;
    p->lchild=p->rchild=NULL;
```

```
    if(key<r->data) r->lchild=p;
    else r->rchild=p;
}

void Inorder(struct Node *p)
{
    if(p)
    {
        Inorder(p->lchild);
        printf("%d ",p->data);
        Inorder(p->rchild);
    }
}

struct Node * Search(int key)
{
    struct Node *t=root;

    while(t!=NULL)
    {
        if(key==t->data)
            return t;
        else if(key<t->data)
            t=t->lchild;
        else
            t=t->rchild;
    }
    return NULL;
}




int main()
{
    struct Node *temp;

    root=RInsert(root,50);
    RInsert(root,10);
    RInsert(root,40);
```

```
    RInsert(root,20);
    RInsert(root,30);

    Delete(root,30);

    Inorder(root);
    printf("\n");

    temp=Search(20);
    if(temp!=NULL)
        printf("element %d is found\n",temp->data);
    else
        printf("element is not found\n");

    return 0;
}
```