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

struct bin_tree {
int data;
struct bin_tree * right, * left;
};
typedef struct bin_tree node;

void insert(node ** tree, int val)
{
   node *temp = NULL;
   if(!(*tree))
   {
      temp = (node *)malloc(sizeof(node));
      temp->left = temp->right = NULL;
      temp->data = val;
      *tree = temp;
      return;
   }

   if(val < (*tree)->data)
   {
      insert(&(*tree)->left, val);
   }
   else if(val > (*tree)->data)
   {
      insert(&(*tree)->right, val);
   }

}
```

```
void deltree(node * tree)
{
   if (tree)
   {
      deltree(tree->left);
      deltree(tree->right);
      free(tree);
   }
}

node* search(node ** tree, int val)
{
   if(!(*tree))
   {
      return NULL;
   }

   if(val < (*tree)->data)
   {
      search(&((*tree)->left), val);
   }
   else if(val > (*tree)->data)
   {
      search(&((*tree)->right), val);
   }
   else if(val == (*tree)->data)
   {
      return *tree;
```

```
    }
}

void main()
{
    node *root;
    node *tmp;
     int i;

    root = NULL;
    insert(&root, 2);
    insert(&root, 41);
    insert(&root, 9);
    insert(&root, 18);
    insert(&root, 6);
    insert(&root, 7);
    insert(&root, 14);


    tmp = search(&root, 4);
    if (tmp)
    {
        printf("Searched node=%d\n", tmp->data);
    }
    else
    {
        printf("Data Not found in tree.\n");
    }

    deltree(root);
}
```