

BINARY SEARCH TREE DELETION

```
#include <stdio.h>
#include <stdlib.h>
struct node{
    int data;
    struct node* left;
    struct node* right;
};
struct node* createNode(int data){
    struct node* newNode=(struct node*)malloc(sizeof(struct node));
    newNode->data=data;
    newNode->left=newNode->right=NULL;
    return newNode;
}
struct node* insert(struct node* root,int data){
    if (root==NULL)return createNode(data);
    if (data<root->data)
        root->left=insert(root->left,data);
    else if (data>root->data)
        root->right=insert(root->right,data);
    return root;
}
struct node* findMin(struct node* root) {
    while(root->left != NULL)
        root=root->left;
    return root;
}
struct node* deleteNode(struct node* root,int key){
    if (root==NULL)
        return root;
    if(key<root->data)
        root->left=deleteNode(root->left,key);
    else if(key > root->data)
        root->right=deleteNode(root->right,key);
    else {
        if (root->left==NULL){
            struct node* temp=root->right;
            free(root);
            return temp;
        }
        else if (root->right==NULL){
            struct node* temp=root->left;
            free(root);
            return temp;
        }
        struct node* temp=findMin(root->right);
        root->data=temp->data;
        root->right=deleteNode(root->right,temp->data);
    }
}
```

```

    }
    return root;
}
void inorder(struct node* root) {
    if (root!=NULL) {
        inorder(root->left);
        printf("%d ",root->data);
        inorder(root->right);
    }
}
int main(){
    struct node* root=NULL;
    root = insert(root, 50);
    insert(root, 30);
    insert(root, 70);
    insert(root, 20);
    insert(root, 40);
    insert(root, 60);
    insert(root, 80);
    printf("Inorder traversal before deletion: ");
    inorder(root);
    root = deleteNode(root, 50);
    printf("\nInorder traversal after deleting 50: ");
    inorder(root);
    return 0;
}

```

OUTPUT:

```

Inorder traversal before deletion: 20 30 40 50 60 70 80
Inorder traversal after deleting 50: 20 30 40 60 70 80
-----
Process exited after 0.09402 seconds with return value 0
Press any key to continue . . .

```