```
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node *right;
    struct node *left;
};
struct node *getnode(int val)
{
    struct node *temp=(struct node*)malloc(sizeof(struct node));
    temp->right=NULL;
    temp->left=NULL;
    temp->data=val;
    return temp;
}
struct node *insert(struct node *root,int val)
{
    struct node *temp=getnode(val);
    if(root==NULL)
    return temp;
    struct node *cur=root;
    struct node *prev=NULL;
    while(cur!=NULL)
    {
        prev=cur;
        if(val<cur->data)
        cur=cur->left;
        else
        cur=cur->right;
    }
    if(val<prev->data)
    prev->left=temp;
    else
    prev->right=temp;
    return root;
}
void inorder(struct node *root)
{
    if(root==NULL)
    return;
    inorder(root->left);
    printf("%d ",root->data);
    inorder(root->right);
}
void postorder(struct node *root)
{
    if(root==NULL)
    return;
    postorder(root->left);
```

```
    postorder(root->right);
    printf("%d ",root->data);
}
void preorder(struct node *root)
{
    if(root==NULL)
    return;
    printf("%d ",root->data);
    preorder(root->left);
    preorder(root->right);
}
struct node *deletekey(struct node *root,int key)
{
    struct node *cur=NULL;
    struct node *q=NULL;
    struct node *suc=NULL;
    struct node *parent=NULL;
    if(root==NULL)
    {
        printf("list is empty\n");
        return root;
    }
    cur=root;
    while(cur!=NULL)
        {
            if(key==cur->data)
            break;
            parent=cur;
            if(key<cur->data)
            cur=cur->left;
            else
            cur=cur->right;
        }
        if(cur==NULL)
        {
            printf("key not found\n");
            return root;
        }
        if(cur->right==NULL)
        q=cur->left;
        else if(cur->left==NULL)
        q=cur->right;
        else{
            q=suc=cur->right;
            while(suc->left!=NULL)
            suc=suc->left;
            suc->left=cur->left;
        }
        if(parent==NULL)
```

```
        return q;
        if(parent->left==cur)
        parent->left=q;
        else
        parent->right=q;
        free(cur);
        return root;

}
void main()
{
    struct node *root=NULL;
    int choice,key,val;
    printf("main
menu\n1.insert\n2.inorder\n3.preorder\n4.postorder\n5.deletebykey\n6.exit
\n");
    for(;;)
    {
        printf("enter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("enter the value: ");
                   scanf("%d",&val);
                   root=insert(root,val);
                   break;
            case 2:inorder(root);
                   break;
            case 3:preorder(root);
                   break;
            case 4:postorder(root);
                   break;
            case 5:printf("enter the key to delete: ");
                   scanf("%d",&key);
                   root=deletekey(root,key);
                   break;
            case 6:printf("exiting program\n");
                   exit(0);
            default:printf("invalid choice\n");
        }
    }
}
```