Program

#include <stdio.h>

#include <stdlib.h>

struct Node{

    struct node \*left;

    int data;

    struct node \*right;

} \*root=NULL;

struct Node \*insertion(struct Node \*p,int key)

{

    struct Node \*t=NULL;

    if(p==NULL)

    {

        t=(struct Node \*)malloc(sizeof(struct Node));

        t->data=key;

        t->left=t->right=NULL;

        return t;

    }

    if(key < p->data)

    p->left=insertion(p->left,key);

    else if(key > p->data)

    p->right=insertion(p->right,key);

    return p;

}

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->left;

        else

            t=t->right;

    }

    return NULL;

}

void traversal(struct Node \*root)

{

    if (root != NULL) {

        traversal(root->left);

        printf("%d \n", root->data);

        traversal(root->right);

    }

}

int Height(struct Node \*p)

{

    int x,y;

    if(p==NULL)return 0;

        x=Height(p->left);

        y=Height(p->right);

        return x>y?x+1:y+1;

}

struct Node \*InPre(struct Node \*p)

{

    while(p && p->right!=NULL)

        p=p->right;

    return p;

}

struct Node \*InSucc(struct Node \*p)

{

    while(p && p->left!=NULL)

        p=p->left;

    return p;

}

struct Node \*Delete(struct Node \*p,int key)

{

    struct Node \*q;

    if(p==NULL)

        return NULL;

    if(p->left==NULL && p->right==NULL)

    {

        if(p==root)

            root=NULL;

            free(p);

            return NULL;

    }

    if(key < p->data)

        p->left=Delete(p->left,key);

    else if(key > p->data)

        p->right=Delete(p->right,key);

    else

    {

        if(Height(p->left)>Height(p->right))

        {

            q=InPre(p->left);

            p->data=q->data;

            p->left=Delete(p->left,q->data);

        }

        else

        {

            q=InSucc(p->right);

            p->data=q->data;

            p->right=Delete(p->right,q->data);

        }

    }

    return p;

}

void main(){

    int key,a;

    struct Node \*temp;

    while(1){

        printf("-------------------Binary Tree Menu-------------------\n");

        printf("1. Insertion\n 2. Deletion\n 3. Traversal\n 4. Searching\n 0. Exit\n Choose the above any options\n");

        scanf("%d",&a);

        switch(a){

            case 1: printf("Enter the data\n");

                    scanf("%d",&key);

                    root=insertion(root,key);

                    break;

            case 2: printf("Enter the element you want to delete\n");

                    scanf("%d",&key);

                    Delete(root,key);

                    break;

            case 3: traversal(root);

                    break;

            case 4: printf("Enter the element to search\n");

                    scanf("%d",&key);

                    temp=Search(20);

                    if(temp!=NULL){

                        printf("element %d is found\n",temp->data);}

                    else{

                        printf("element is not found\n");}

                    break;

            case 0: exit(1);

            default: printf("Invalid option\n");

                    break;

        }

    }

}