#include "def.h"

typedef int TElemType;

typedef struct BiTNode {

TElemType data;

struct BiTNode \*lchild, \*rchild;

BiTNode():lchild(0),rchild(0){}

}BiTNode,\*BiTree;

void CreateBiTree(BiTree &T);

void InsertBiTree(BiTree &T, TElemType e);

void Delete(BiTree &P);

void print(BiTree T, int level);

#include"BiTree.h"

#include<iostream>

using namespace std;

void InsertBiTree(BiTree &T, int e)

{

if (T == 0)

{

T = new BiTNode;

T->data = e;

T->lchild = T->rchild = 0;

}

else if (T->data < e)

InsertBiTree(T->rchild, e);

}

void print(BiTree T, int level)

{

int i;

for (i = 0; i < level; i++)

cout << "\t" ;

if(T == 0)

{

cout << "\_" << endl;

return;

}

cout << T->data << endl;

print(T->lchild, level + 1);

print(T->rchild, level + 1);

}

void Delete(BiTree &T)

{

BiTree P1 = T, P2 = T->lchild, P3 = T->rchild;

cout << T->data << endl;

delete P1;

if (P2)

{

T = P2;

P1 = P2;

while (P1->rchild!=0) P1 = P1->rchild;

P1->rchild = P3;

}

else

T = P3;

}

void Delete(BiTree &T, int e)

{

if (T == 0)

return ;

else if (T->data > e)

Delete(T->lchild, e);

else if (T->data <e)

Delete(T->rchild,e);

else

Delete(T);

}

#include"def.h"

#include"BiTree.h"

#include<iostream>

using namespace std;

int main()

{

BiTree T=0;

int a[] = { 2,8,5,10,7,6 };

int i;

for (i = 0; i < 6; i++)

InsertBiTree(T, a[i]);

print(T, 0);

Delete(T,2);

print(T, 0);

system("pause");

return 0;

}