#include"stdafx.h"

#include <iostream>

using namespace std;

typedef char Type;

typedef struct BiThrNode

{

Type data;

struct BiThrNode\* lchild, \*rchild;

} BiThrNode, \*BiThrTree;

BiThrTree T;

BiThrNode \*f;

BiThrNode \*p;

BiThrNode\*Search(char key)

{

p = T;

f = NULL;

while (p!=NULL&&p->data!=key)

{

if (key < p->data)

{

f = p;

p = p->lchild;

}

else

{

f = p;

p = p->rchild;

}

}

return p;

}

bool Insert(BiThrTree &r, const char &e)

{

BiThrNode \*f=NULL;

if (Search(e) == NULL)

{

BiThrNode\*p = new BiThrNode;

p->data = e;

if (e < f->data)

{

f->lchild = p;

}

else

f->rchild = p;

return true;

}

return false;

}

bool Delete(BiThrTree&r, char e)

{

BiThrNode\*p = Search(e);

BiThrNode\* q, \*s;

if (p->rchild == NULL)

{

q = p;

p = p->lchild;

f->rchild = p;

free(q);

}

else if (p->lchild == NULL)

{

q = p;

p = p->rchild;

f->rchild = p;

free(q);

}

else

{

q = p;

s = p->lchild;

while (s->rchild)

{

q = s;

s = s->rchild;

}

p->data = s->data;

if (q != p)

q->rchild = s->lchild;

else

q->lchild = s->lchild;

free(s);

}

return true;

}

//创建一棵二叉树，约定用户遵照前序遍历方式

void CreateBiThrTree(BiThrTree\* T)

{

char c;

cin >> c;

if ('^' == c)

{

\*T = NULL;

}

else

{

\*T = new BiThrNode;

(\*T)->data = c;

CreateBiThrTree(&(\*T)->lchild);

CreateBiThrTree(&(\*T)->rchild);

}

}

int main()

{

CreateBiThrTree(&T);

Delete(T, 'H');

return 0;

}

使用的输入数据：

CBA^^^HED^^GF^^^LJI^^K^^^^^^^^^^^^^^^^^^^^