#include <stdio.h>

#include <malloc.h>

#include <string.h>

typedef struct BiTNode

{

char s[20];

struct BiTNode \*lchild,\*rchild,\*order;

}BiTNode,\*BiTree;

void print(BiTree T)

{

if(T)

{

print(T->lchild);

print(T->rchild);

printf("%s ",T->s);

}

}

struct BiTNode\* judge(char a[20],struct BiTNode \*T)

{

BiTNode \*q=T;

int flag=0;

while(q)

{

if(strcmp(a,q->s)<0) q=q->lchild;

else if(strcmp(a,q->s)>0) q=q->rchild;

else return q;

}

return NULL;

}

int main()

{

BiTNode \*p,\*q,\*t,\*der;

BiTree T;

char a[20],b[20];

char \*ttt[10000];

int i,flag,j,k,num,c,d;

T=(struct BiTNode\*)malloc(sizeof(BiTNode));

scanf("%s",T->s);

T->lchild=NULL;

T->rchild=NULL;

der=T;

scanf("%s",a);

while(a[0]!='#')

{

flag=0;

p=(struct BiTNode\*)malloc(sizeof(BiTNode));

for(i=0;a[i]!='\0';i++)

p->s[i]=a[i];

p->s[i]='\0';

p->lchild=NULL;

p->rchild=NULL;

der->order=p;

der=p;

q=T;

while(q)

{

t=q;

if(strcmp(q->s,p->s)>=0) {q=q->lchild;flag=1;}

else {q=q->rchild;flag=2;}

}

if(flag==1) t->lchild=p;

else if(flag==2) t->rchild=p;

scanf("%s",a);

}

print(T);

printf("\n");

scanf("%s",a);

while(a[0]!='#')

{

der=T;

num=0;

while(der!=NULL)

{

flag=0;

for(i=0;a[i]!='\0';i++);

for(j=0;der->s[j]!='\0';j++);

c=i-j;

if(c!=1&&c!=-1&&c!=0) flag=0;

else if(c==0)

{

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

{

if(a[c]!=der->s[c])

{

flag++;

if(flag>1) break;

}

}

if(c==i&&flag==0) flag=3;

if(flag==1) {ttt[num]=der;num++;}

}

else if(c==1)

{

for(c=0,d=0;c<i;)

{

if(a[c]!=der->s[d])

{

c++;

flag++;

if(flag>1) break;

}

else {c++;d++;}

}

if(flag==1) {ttt[num]=der;num++;}

}

else if(c==-1)

{

for(c=0,d=0;d<j;)

{

if(a[c]!=der->s[d])

{

d++;

flag++;

if(flag>1) break;

}

else {c++;d++;}

}

if(flag==1) {ttt[num]=der;num++;}

}

if(flag==3)

{printf("%s is correct\n",a);break;}

der=der->order;

}

if(flag==3) {scanf("%s",a);continue;}

else

{

printf("%s:",a);

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

{

der=ttt[i];

printf(" %s",der->s);

}

}

printf("\n");

scanf("%s",a);

}

return 0;

}

**问题 J: 单词检查(Ⅱ)- 二叉排序树实现**

时间限制: 2 Sec 内存限制: 128 MB  
提交: 363 解决: 130  
[[提交](http://acm.hnust.cn/JudgeOnline/submitpage.php?cid=1569&pid=9&langmask=0)][[状态](http://acm.hnust.cn/JudgeOnline/problemstatus.php?id=1921)][[讨论版](http://acm.hnust.cn/JudgeOnline/bbs.php?pid=1921&cid=1569)]

**题目描述**

许多应用程序，如字处理软件，邮件客户端等，都包含了单词检查特性。单词检查是根据字典，找出输入文本中拼错的单词，我们认为凡是不出现在字典中的单词都是错误单词。不仅如此，一些检查程序还能给出类似拼错单词的修改建议单词。 例如字典由下面几个单词组成：  
bake cake main rain vase  
如果输入文件中有词vake ，检查程序就能发现其是一个错误的单词，并且给出 bake, cake或vase做为修改建议单词。  
修改建议单词可以采用如下生成技术：  
(1)在每一个可能位置插入‘a-'z'中的一者  
(2)删除单词中的一个字符  
(3)用‘a'-'z'中的一者取代单词中的任一字符

很明显拼写检查程序的核心操作是在字典中查找某个单词，如果字典很大，性能无疑是非常关键的。

你写的程序要求读入字典文件，然后对一个输入文件的单词进行检查，列出其中的错误单词并给出修改建议。

本题要求使用使用二叉排序树维护字典。**为了防止有些人取巧，本题要求输出相应的二叉排序树后序遍历。**

**输入**

输入分为两部分。  
第一部分是字典，每个单词占据一行，最后以仅包含'#'的一行表示结束。所有的单词都是不同的，字典中最多10000个单词。  
  
输入的第二部分包含了所有待检测的单词，单词数目不超过50。每个单词占据一行，最后以仅包含'#'的一行表示结束。  
  
字典中的单词和待检测的单词均由小写字母组成，并且单词最大长度为15。

**输出**

第一行输出二叉排序树字典的后序遍历，每一个单词后面跟一个空格。

然后按照检查次序每个单词输出一行，该行首先输出单词自身。如果单词在字典中出现，接着输出" is correct"。如果单词是错误的，那么接着输出':'，如果字典中有建议修改单词，则按照字典中出现的先后次序输出所有的建议修改单词（每个前面都添加一个空格），如果无建议修改单词，在':'后直接换行。

**样例输入**

i

is

has

have

be

my

more

contest

me

too

if

award

#

me

aware

m

contest

hav

oo

or

i

fi

mre

#

**样例输出**

award contest be have has if me more too my is i

me is correct

aware: award

m: i my me

contest is correct

hav: has have

oo: too

or:

i is correct

fi: i

mre: more me