

### 【问题描述】

从标准输入中读入一个英文单词及查找方式，在一个给定的英文常用单词字典文件 dictionary3000.txt 中查找该单词，返回查找结果（查找到返回 1，否则返回 0）和查找过程中单词的比较次数。查找前，先将所有字典中单词读入至一个单词表（数组）中，然后按要求进行查找。字典中单词总数不超过 3500，单词中的字符都是英文小写字母，并已按字典序排好序（可从课件下载区下载该字典文件）。字典中的单词和待查找单词的字符个数不超过 20。

查找方式说明：查找方式以 1~4 数字表示，每个数字含义如下：

- 1：在单词表中以顺序查找方式查找，因为单词表已排好序，遇到相同的或第一个比待查找的单词大的单词，就要终止查找；
- 2：在单词表中以折半查找方式查找；
- 3：在单词表中通过索引表来获取单词查找范围，并在该查找范围中以折半方式查找。索引表构建方式为：以 26 个英文字母为头字母的单词在字典中的起始位置和单词个数来构建索引表，如：

字母	起始位置	单词个数
a	0	248
b	248	167
...	...	...

该索引表表明以字母 a 开头的单词在单词表中的开始下标位置为 0，单词个数为 248。

- 4：按下面给定的 hash 函数为字典中单词构造一个 hash 表，hash 冲突时按字典序依次存放单词。hash 查找遇到冲突时，采用链地址法处理，在冲突链表中找到或未找到(遇到第一个比待查找的单词大的单词或链表结束)便结束查找。

```
/* compute hash value for string */
#define NHASH 3001
#define MULT 37
unsigned int hash(char *str)
{
    unsigned int h=0;
    char *p;

    for(p=str; *p!='\0'; p++)
        h = MULT*h + *p;
    return h % NHASH;
}
```

提示：hash 表可以构建成指针数组，hash 冲突的单词形成一有序链表。

### 【输入形式】

单词字典文件 dictionary3000.txt 存放在当前目录下，待查找单词和查找方式从标准输入读取。待查找单词只包含英文小写字母，与表示查找方式的整数之间以一个空格分隔。

### 【输出形式】

将查找结果和单词比较次数输出到标准输出上，两整数之间以一个空格分隔。

### 【样例输入与输出】

单词字典文件 dictionary3000.txt 与课件下载中提供的相同，下面两列中，左侧为

待查找单词与查找方式，右侧为对应的输出结果：

wins 1	0 3314
wins 2	0 12
wins 3	0 7
wins 4	0 2
yes 1	1 3357
yes 2	1 10
yes 3	1 4
yes 4	1 1

**【样例说明】**

wins 在单词字典中不存在，4 种查找方式都输出结果 0，顺序查找、折半查找、索引查找和 hash 查找的单词比较次数分别为：3314、12、7 和 2 次（wins 的 hash 位置与字典中 physics 和 suggest 相同）。

yes 在单词字典中存在，4 种查找方式都输出结果 1，顺序查找、折半查找、索引查找和 hash 查找的单词比较次数分别为：3357、10、4 和 1。

**【评分标准】**

该题要求输出查找结果和查找过程中的单词比较次数，提交程序名为 find.c。