

P1308 [NOIP2011 普及组] 统计单词数

题目描述

一般的文本编辑器都有查找单词的功能，该功能可以快速定位特定单词在文章中的位置，有的还能统计出特定单词在文章中出现的次数。

现在，请你编程实现这一功能，具体要求是：给定一个单词，请你输出它在给定的文章中出现的次数和第一次出现的位置。注意：匹配单词时，不区分大小写，但要求完全匹配，即给定单词必须与文章中的某一独立单词在不区分大小写的情况下完全相同（参见样例1），如果给定单词仅是文章中某一单词的一部分则不算匹配（参见样例2）。

输入格式

共22行。

第11行为一个字符串，其中只含字母，表示给定单词；

第22行为一个字符串，其中只可能包含字母和空格，表示给定的文章。

输出格式

一行，如果在文章中找到给定单词则输出两个整数，两个整数之间用一个空格隔开，分别是单词在文章中出现的次数和第一次出现的位置（即在文章中第一次出现时，单词首字母在文章中的位置，位置从00 开始）；如果单词在文章中没有出现，则直接输出一个整数-1-1。

输入样例

```
To
to be or not to be is a question
```

输出样例

```
2 0
```

解析

关键点：

- 1、搜索的目标单词是独立的单词，被包含在中间的不算
- 2、多次搜索

编码

```
#include <bits/stdc++.h>

//命名空间
using namespace std;

int main() {
    //定义两个字符串
    string a;
    string b;
    //用string库，调用getline，直接读入一整行
    getline(cin, a);
    getline(cin, b);
    //转换大小写，可以都转换为大写，或者小写
    for (int i = 0; i < a.length(); ++i) {
        a[i] = tolower(a[i]);
    }
    for (int i = 0; i < b.length(); ++i) {
        b[i] = tolower(b[i]);
    }
    //待查找的必须是独立的单词，因此我们在其前后位置加上空格，避免错误选择；
    a = ' ' + a + ' ';
    //防止由于增加空格而引起的位置错乱
    b = ' ' + b + ' ';
    //先看看会不会找不到，用a.find()和string::npos
    if (b.find(a) == string::npos) {
        cout << -1 << endl;
    }
    //如果找得到
    else {
        int first = b.find(a);
        int position = first;
        int s = 0; //计数器初始化为0
        while (position != string::npos) {
            ++s; //计数器
            position = b.find(a, position + 1);
        }
        //输出第一个和总共有几个
        cout << s << " " << first << endl;
    }
}
```

```
//函数返回值为0，结束整个程序  
return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

