这题的做法太多了，但大部分其他题解都讲完了，下面的大概都是一些

**第一种做法 指针做法**

先来介绍一下我的思路吧~

* 输入
* 统一转换成小写的字母
* 使用系统函数找出单词首次出现的位置，然后改变指针
* 不断统计答案
* 输出

[关于指针的更多介绍](https://89396.blog.luogu.org/c-zhi-zhen-ge-zhong-xuan-xue-cao-zuo)

这里使用了很多系统函数，在代码中详细介绍：

#include <cstring>

#include <cctype>

#include <cstdio>

void strlower (char \*a) {//手写函数，将大写字母转换成小写字母

for(int i = 0; a[i]; i ++ ) {

if(isupper(a[i])) a[i] = tolower(a[i]);//isupper是判断是否是大写字母的系统函数，tolower是将其转换成小写字母的函数

}

}

int main () {

char destination[1000001], \*q, source[11], \*p;//destintion是要找的文章，source是要找的单词，p和q都是指针类，分别代表当前搜索到什么地方了和最后一次找到单词的指针

bool flag = false;//判断是否找到了

int ans = 0, ans1 = -1;//个数和首次出现的位置，ans1的初值是-1是因为在没找到的时候就直接输出就行了，省事

gets(source);

gets(destination);//输入

strlower(destination);//全部转换成小写字母

strlower(source);

int len = strlen(source);//长度，在后面防止越界和加快速度

p = destination;//先将指针设为全部

for(; q = strstr(p, source); ) {//循环，strstr是在一个字符串里面给定一个字符串，寻找有没有这个字符串，若有，返回首次出现的指针否则返回NULL（空指针）

if( q != NULL//找到了

&& ( q == destination || \*(q - 1) == ' ') //第一个条件是防止越界，第二个是判断前一个是不是空格

&& ( \*(q + len) == '\0' || \*(q + len) == ' ' ) ) {//如果后面也是空格

ans ++ ;//答案加一

if(flag == false) {//如果是首次找到

flag = true;

ans1 = q - destination;//第一个位置

}

}

p = q + len;//刷新指针

}

if(flag == true)//找到了

printf("%d %d" , ans, ans1);//输出

else

printf("%d", ans1);//输出-1

return 0;

}

**第二种做法 有穷自动机**

图灵机大概就是一个“自动机”，就是说代码分好几种状态，每种状态做不同的事。

**举个简单的例子吧**

输入一个字符串，输入的只有两种字符，一种是字母，一种是空格。现在求一共有几个单词。注意，有可能有多个空格连在一起，开头和结尾都有可能有空格。

那么这是一道简单的有穷自动机，运行时分两种情况：

①是空格

②是字母

（其实当前状态就是上一个字符的状态

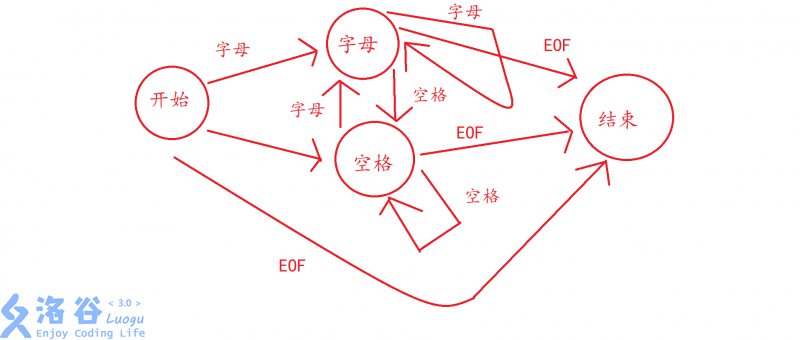
那么在遍历数组的时候拿一个变量记录下来当前是什么状态，可以用$0$代表当前是空格状态，$1$代表是字母状态

当如果当前状态是$1$，而现在却遇到空格，那么就计数器加一，**同时要将状态改为$0$**，如果当前状态是$0$，现在的字符却是字母，就只将状态改为$1$

**BUT!**

在跳出循环的时候如果状态是$1$，要将计数器加一，否则如果最后是字母就会少统计一个单词！（想想为什么）

状态图：



然后就是代码：

**注：不是本题的代码！！！是这个例子的代码！！！**

#include <cstdio>

int main () {

char a[1001];

int state, ans = 0;

gets(a);

if(a[0] == ' ') state = 0;//设置初始值

else state = 1;

for(int i = 1; a[i]; i ++ ) {//要从一开始遍历，因为零已经遍历过了

if(a[i] == ' ') {//是空格

if(state == 1) {//当前状态（前一个）是字母，说明找到一个单词了

ans ++ ;//答案加一

state = 0;//千万别忘了改状态

}

}

else {//是字母

if(state == 0) {//当前状态（前一个）是空格

state = 1;//将状态改为1

}

}

}

if(state == 1)//最后还要判断一下千万不要忘记

ans ++ ;

printf("%d", ans);

return 0;

}

那么，这就是简单的自动机代码，现在看看本题用自动机如何做

其实一样，就是注意字母状态分时要查找单词状态和不是要查找单词状态，而且单词第$n$个字母的状态就用$n$来表示

以下是code：

#include <cstdio>

#include <cctype>

#include <cstring>

const int SPACE = 0;//三种状态，这是空格状态

const int LETTER = -1;//字母状态，但这表示不是要查找的单词的字母的状态

const int WORD = 1;//而这种状态是要查找的单词的状态

//当然了，如果状态时大于1的数，说明是要查找的单词的中间部分的状态，上文讲过了

inline void strlower (char \*a) {//不解释，上面的代码有了

for(int i = 0; a[i]; i ++ ) {

if(isupper(a[i])) a[i] = tolower(a[i]);

}

}

int main () {

char a[1000001], word[20];

int ans = 0;

int ans2 = -1;

int state = 0;//表状态，假设是空格，因为空格上来就判断是不是三种状态

int i;

gets(word);

gets(a);

strlower(a);

strlower(word);

int len = strlen( word );

for(i = 0; a[i]; i ++ ) {//遍历数组

switch ( state ) {

case SPACE : //如果当前状态（上一个）是空格

if(a[i] == word[0]) state = WORD;//变成单词第一个字母状态

else if(a[i] == ' ') state = SPACE;//其实这句话可以省略，因为反正都是空格状态，改它是一样的

else state = LETTER;//剩下的肯定是其他字母状态了

break;

case LETTER : //是其他字母状态

if(a[i] == ' ') state = SPACE;//空格状态

break;

default: //是要查找的单词状态

if ( state < len ) {//还不是最后一个字母

if(a[i] == ' ') state = SPACE;

else if(a[i] == word[state]) state ++ ;//变成下一个字母状态

else state = LETTER;//其他字母状态

}

else if (state == len )//是最后一个字母

{

if(a[i] == ' ') {//如果下一个是空格，找到了！

state = SPACE;//状态不要忘记改变

if(ans2 == -1)//第一次找到，记录下来位置

ans2 = i - len;//因为i是单词的尾，所以要减长度

ans ++ ;//个数加一

}

else state = LETTER;//可惜，最后跟着其他字母，不是单词

}

}

}

if(state == len) {

ans ++ ;

if(ans2 == -1)

ans2 = i - 1 - len;

}

if(ans2 == -1) printf("-1");

else printf("%d %d", ans, ans2);

return 0;

}