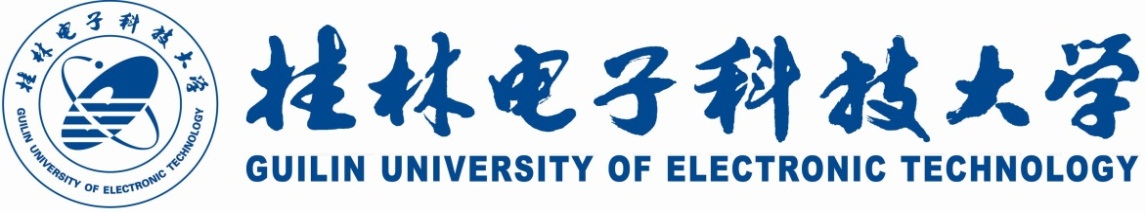
组号： 25



数据结构与算法课程设计

说明书

**文学研究助手与模式匹配算法KMP**

院 系： 计算机工程学院

专 业： 网络工程

指导教师： 谢邵敏

2021年 3 月 11 日

分组与分工说明

1. 组员名单：刘健成
2. 分工说明（给出每个组员所完成的任务）：

代码编写与修改，文档编写与修改。

摘 要

文学研究人员需要统计某篇英文小说中某些形容词的出现次数和位置,然而一篇英语小说里的形容词多达上百种，甚至上千种，对于人类而言，这是个繁琐且复杂的工程，不但会消耗大量的时间和精力，并且正确率无法得到保障。

随着科技的发展，计算机技术的日渐成熟和广泛运用，当今计算机系统的运算速度已达到每秒万亿次，微机也可达每秒亿次以上，使大量复杂的科学计算问题得以解决。因此，我们可以借助计算机来帮助我们解决那些复杂的问题，节约大量的时间和精力，大幅度提高工作效率。

本程序基于Code::Blocks开发环境，以C++为开发语言，并借助KMP匹配算法来减少重复匹配的繁琐过程，以此来提高程序运行效率。相对于简单匹配算法而言，KMP算法具有较高的查询效率，简单匹配算法的时间复杂度为O(m\*n)，而KMP匹配算法的时间复杂度为O(m+n)，使用KMP算法更有利于减少检索的复杂过程。

本课题着重于文本关键字的查询，具有较大的实际意义。通过使用本程序，用户可以迅速查询到所需文本里的关键字及其对应位置,对于相关的学术研究者而言，本程序能大幅度的减少工作时间，提高工作效率。

关键词：文学研究人员 C++ KMP匹配算法 查询

**Abstract**

Literature researchers need to count the frequency and location of some adjectives in an English novel. However, there are hundreds or even thousands of adjectives in an English novel. For human beings, this is a tedious and complex project, which not only consumes a lot of time and energy, but also cannot guarantee the accuracy.

With the development of science and technology, computer technology is becoming more and more mature and widely used. The computing speed of today's computer system has reached trillions of times per second, and the computer can also reach more than billion times per second, so that a large number of complex scientific computing problems can be solved. Therefore, we can use computers to help us solve those complex problems, save a lot of time and energy, and greatly improve work efficiency.

This program is based on Code::blocks development environment, with C++ as the development language, and with KMP matching algorithm to reduce the tedious process of repeated matching, so as to improve the efficiency of the program. Compared with simple matching algorithm, KMP algorithm has higher query efficiency. The time complexity of simple matching algorithm is O (m\*n), while that of KMP matching algorithm is O (m+n). Using KMP algorithm is more conducive to reducing the complexity of retrieval process.

This topic focuses on the query of text keywords, which has great practical significance. By using this program, users can quickly find the keywords and their corresponding positions in the required text. For relevant academic researchers, this program can greatly reduce working hours and improve work efficiency.

**Keyword: literature researcher C + + KMP matching algorithm query**

目录

[引 言 1](#_Toc67865478)

[第1章 系统分析 2](#_Toc67865479)

[1.1 需求分析 2](#_Toc67865480)

[1.2 系统可行性 2](#_Toc67865481)

[1.2.1 C++编程语言介绍 3](#_Toc67865482)

[1.2.2 Code::Blocks介绍 3](#_Toc67865483)

[1.2.3 模式匹配算法KMP介绍 3](#_Toc67865484)

[第2章 系统框架与功能 6](#_Toc67865485)

[2.1系统框架的构建 6](#_Toc67865486)

[2.1.1 系统流程设计 6](#_Toc67865487)

[2.1.2 所需函数及其功能 7](#_Toc67865488)

[2.2系统模块的设计 7](#_Toc67865489)

[2.2.1 菜单模块 7](#_Toc67865490)

[2.2.2 字体颜色模块 10](#_Toc67865491)

[2.2.3 查询模块 11](#_Toc67865492)

[第3章 系统运行与测试 15](#_Toc67865493)

[3.1系统测试与修改 15](#_Toc67865494)

[3.2系统运行 17](#_Toc67865495)

[3.2.1 菜单模块运行界面 17](#_Toc67865496)

[3.2.2退出系统 17](#_Toc67865497)

[3.2.3查询文本关键字 17](#_Toc67865498)

[3.2.4 输出文本内容 18](#_Toc67865499)

[结论 20](#_Toc67865500)

[参考文献 21](#_Toc67865501)

[致谢 22](#_Toc67865502)

[附录 23](#_Toc67865503)

引 言

计算机是20世纪最先进的科学技术发明之一，对人类的生产活动和社会活动产生了极其重要的影响，并以强大的生命力飞速发展。它的应用领域从最初的军事科研应用扩展到社会的各个领域，已形成了规模巨大的计算机产业，带动了全球范围的技术进步，由此引发了深刻的社会变革，计算机已遍及一般学校、企事业单位，进入寻常百姓家，成为信息社会中必不可少的工具。

在信息化的今天，人类已经无法只用纸张来记录数据，因为这样无法记录海量的数据，并且数据不易保存，容易丢失难以找回；而使用计算机来记录数据，不仅可以记录海量的内容，还可以将其共享给其他用户，实现数据的随时可取，随时可用。人们喜欢用计算机来记录数据最重要的一点是，数据可以随身携带，且可以备份复原。

以人力来统计英文小说中某些形容词的出现次数和位置，浪费了大量人力和物力，已经无法适应当今的世界。因此，依靠计算机的算力来统计所需词语，开发一个程序也就显得理所当然。

该文学研究系统基于Code::Blocks开发环境，以C++为开发语言，利用模式匹配算法KMP的核心——利用匹配失败后的信息，尽量减少模式串与主串间的匹配次数以达到快速匹配的目的，以此来节约工作时间，提高工作效率。

同时它成功实现了对于英文小说中某些形容词的出现次数和位置的查询和统计，并进行了更加全面的开发，让用户来输入需要统计的词语，并一次性的显示出所需查询的词语出现的次数和位置，大大增加了程序的可用性，提高程序的活力。

# 系统分析

1.1 需求分析

正式开发管理系统之前进行调研是非常必要的，其必要性主要表现在以下几个方面：

（1）明确用户对通讯录管理系统的要求，以根据调查结果进行可行性分析，确认系统的开发是否可行。

（2）提出新系统的人员并不都是系统研究人员，有些人对于系统的功能和处理数据的方法没有明确的认识，他们只是根据自己业务工作的需求提出了要求，系统开发人员要对此进行详细的调查和分析，确认用户的要求可以通过现有的计算机技术实现，保证开发的管理系统的功能和用户提出的要求相吻合。

（3）对于现在大家已经接触过的通讯录管理系统，种类繁多，无论是何种形式的，都要详细的调查现有的系统中信息处理情况，系统内部功能结构，以便设计出一个合理的、更好的系统逻辑模型，为自己的系统的设计工作打好基础，保证整个系统开发的质量。

总之，必要的对现行系统进行详细的调查，明确用户需求，保证开发的系统功能与用户的要求相吻合，避免耗费不必要的人力、物理、财力。

根据对当前市面上的调查，还未出现与此项目相关的产品，再结合对相关职业的抽样问卷调查，有63%的人对此项目有一定的需求，特别是文学研究人员需要统计某篇英文小说中某些形容词的出现次数和位置，单纯的人力劳动，无法保障工作的准确率和工作效率。

综上所述，本项目具有一定用户需求和市场。

1.2 系统可行性

**在进行系统可行性的分析之前，必须得对系统的相关功能做一些预设，结合自身的实际能力，再对其加以改进或删除。现在我们先对系统的主要功能做一些预设。**

本系统的主要功能是统计英文小说中某些形容词的出现次数和位置，英文小说存于一个文本文件中。需要统计的词汇集合要一次输入完毕，即统计工作必须在程序的一次运行之后就全部完成。程序的输出结果是每个词的出现次数和出现位置所在的行的行号，格式自行设计。需要统计的“单词”在文本串中不跨行出现，它或者从行首开始，或者前置以一个空格符。

**接下来我们根据所设功能对其进行技术分析，依据开发语言和开发环境以及大致的算法，来判断该系统的可行性。**

### 1.2.1 C++编程语言介绍

C++是C语言的继承，它既可以进行[C语言](https://baike.baidu.com/item/C%E8%AF%AD%E8%A8%80/105958" \t "_blank)的过程化[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1/223952)，又可以进行以抽象数据类型为特点的基于对象的程序设计，还可以进行以继承和多态为特点的面向对象的程序设计。C++擅长面向对象程序设计的同时，还可以进行基于过程的程序设计，因而C++就适应的问题规模而论，大小由之。

C++不仅拥有计算机高效运行的实用性特征，同时还致力于提高大规模程序的编程质量与程序设计语言的问题描述能力。C++常用于系统开发，引擎开发等应用领域，支持类、封装、继承、多态等特性。C++语言灵活，运算符的数据结构丰富、具有结构化控制语句、程序执行效率高，而且同时具有高级语言与汇编语言的优点。

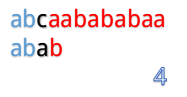
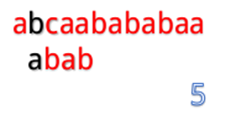
### 1.2.2 Code::Blocks介绍

Code::Blocks 是一个开放源码的全功能的跨平台C/C++集成开发环境。 Code::Blocks是开放源码软件。Code::Blocks由纯粹的C++语言开发完成，它使用了著名的图形界面库[wxWidgets(3.x)](https://baike.baidu.com/item/wxWidgets" \t "_blank)版。对于追求完美的C++程序员，再也不必忍受Eclipse的缓慢，再也不必忍受VS.NET的庞大和高昂的价格。

Code::Blocks支持语法彩色醒目显示，支持代码完成，支持工程管理、项目构建、调试。

### 1.2.3 模式匹配算法KMP介绍

KMP算法是一种改进的字符串匹配算法，由D.E.Knuth，J.H.Morris和V.R.Pratt提出的，因此人们称它为克努特—莫里斯—普拉特操作（简称KMP算法）。KMP算法的核心是利用匹配失败后的信息，尽量减少模式串与主串的匹配次数以达到快速匹配的目的。具体实现就是通过一个next()函数实现，函数本身包含了模式串的局部匹配信息。KMP算法的时间复杂度为O(m+n)。

**为了对模式匹配算法KMP有更深刻的理解，下面我们来看一个例子，现在有两个字符串A和B，问你在A中是否有B，有几个？为了方便叙述，我们先给定两个字符串的值  
A="abcaabababaa"  
B="abab"**  
那么普通的匹配是怎么操作的呢？  
当然就是一位一位地比啦。（**下面用蓝色表示已经匹配，黑色表示匹配失败**）但是我们发现这样匹配很浪费**!**  
为什么这么说呢，我们看到第4步：  
  
在第4步的时候，我们发现第3位上c与a不匹配，然后第五步的时候我们把B串向后移一位，再从第一个开始匹配。  
  
这里就有一个对已知信息很大的浪费，因为根据前面的匹配结果，我们知道B串的前两位是ab，所以不管怎么移，都是不能和b匹配的,所以应该直接跳过对A串第二位的匹配，对于A串的第三位也是同理。

**或许这这个例子还不够经典，我们再举一个。**  
A="abbaabbbabaa"  
B="abbaaba"

在这个例子中，我们依然从第1位开始匹配，直到匹配失败：  
abbaabbbabba  
abbaaba  
我们发现第7位不匹配  
那么我们若按照原来的方式继续匹配，则是把B串向后移一位，重新从第一个字符开始匹配  
abbaabbbabba  
\_abbaaba  
依然不匹配，那我们就要继续往后移。  
等会！  
既然我们已经匹配了前面的6位，那么我们也就知道了A串这6位和B串的前6位是匹配的，我们能否利用这个信息来优化我们的匹配呢？  
也就是说，我们能不能在上面匹配失败后直接跳到  
abbaabbbabba  
\_\_\_\_abbaaba  
这样就可以省去很多不必要的匹配。

KMP算法就是解决上面的问题的，在讲述之前，我们先摆出两个概念：

前缀：指的是字符串的子串中从原串最前面开始的子串，如abcdef的前缀有：a,ab,abc,abcd,abcde

后缀：指的是字符串的子串中在原串结尾处结尾的子串，如abcdef的后缀有：f,ef,def,cdef,bcdef

KMP算法相对于其他算法的优势在于，每当一趟匹配过程中出现字符比较不等时，不需回溯i指针，而是利用已经得到的“部分匹配”的结果将模式向右“滑动”尽可能远的一段距离后，继续进行比较。

KMP算法是在已知模式串的next函数值的基础上执行的，那么，如何求得模式串的next函数值呢？next[i]表示的是前i的字符组成的这个子串最长的相同前缀后缀的长度。怎么理解呢？

例如字符串aababaaba的相同前缀后缀有a和aaba，那么其中最长的就是aaba。next数组的值是代表着字符串的前缀与后缀相同的最大长度,(不能包括自身)，而KMP算法的精髓就在于next数组，从而达到跳跃式匹配的高效模式。

这里举个例子：

模式串t A B A B A A

下标 0 1 2 3 4 5

next 0 0 1 2 3 1

其中next[0]代表t[0]~t[0]即"A"的最大前后缀，显然为0.

next[1]代表t[0]~t[1]即"AB"的最大前后缀，为0.

next[2]代表t[0]~t[2]即"ABA"的最大前后缀，即"A",长度为1.

next[3]代表t[0]~t[3]即"ABAB"的最大前后缀，即"AB",长度为2.

next[4]代表t[0]~t[4]即"ABABA"的最大前后缀，即"ABA",长度为3.

next[5]代表t[0]~t[5]即"ABABAA"的最大前后缀，即"A",长度为1.

有了next数组我们就可以利用它来加快匹配效率，更快的为用户找出所需内容，提高工作效率。

# 系统框架与功能

2.1系统框架的构建

### 2.1.1 系统流程设计

该流程图是根据用户需求设计的系统大体流程，通过该流程图，可以大致了解系统大体框架和大致功能，具体如下[图 2‑1](#_Toc67821701)

**用户输入**

菜单

**输出文本内容**

**查询文本关键字**

**退出系统**

**判断文本**

**判断文本**

**输出结果**

**输出结果**

**用户输入**

**用户输入**

**退出系统**

**返回菜单**

**返回菜单**

**退出系统**

***图 2‑1***

### 2.1.2 所需函数及其功能

1. int length(SString str) //求串长

2.exit(0) //退出程序

3.system("cls") //清屏函数

4.fopen(name,"r") //以只读方式打开文件

5.fclose(fp) //关闭文件

6.feof(fp) //若返回值非零，说明文件结束

7.fscanf(fp,"%c",&ch) //读取一个字符,遇到空格和换行结束

8.putchar(ch) //输出一个字符

9. void menu() //系统菜单函数，用于菜单输出及用户输入

10. void get\_next(SString T,int next[ ] //求模式串 T的 next函数值并存入数组 next

11. int Index(SString S,SString T,int pos) //模式匹配算法KMP

12. void color(short x) //局部字体改变颜色,用于提示错误的红色字体。

13.void find(char name[],SString keys) //判断文本是否存在以及查询用户输入的关键字

14. void main() //主函数,输出系统菜单函数

2.2系统模块的设计

### 2.2.1 菜单模块

（1）主要思想：

菜单函数menu()用于简洁明了、方便快捷地供用户选择系统功能，更重要的是用于连接其他的模块。此函数主要运用一个switch()函数来完成用户输入及其系统输出功能。若用户输入的数字不在0,1,2之中，则提示用户输入错误，重新输入。

（2）具体代码如下：

void menu()

{

char name[100]; //存储输入的小说路径字符串

SString words[100]; //定义字符串数组， 用于存储输入的关键字

int n,i,m;

FILE \*fp;

char ch;

printf("\t\t\t \*\*\*\*\*\*\*\*\*\*\* 欢迎使用文学研究系统 \*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t┌────────┌───┐────────┐\n");

printf("\t\t\t│ │ 菜单 │ │\n");

printf("\t\t\t│ └───┘ │\n");

printf("\t\t\t│\*\*\*\*\*\*\* 0.退出系统 \*\*\*\*\*\*\*│\n");

printf("\t\t\t│\*\*\*\*\*\*\* 1.查询文本关键字 \*\*\*\*\*\*\*│\n");

printf("\t\t\t│\*\*\*\*\*\*\* 2.输出文本内容 \*\*\*\*\*\*\*│\n");

printf("\t\t\t└─────────────────────┘\n");

printf("\t\t\t\*\*\*\*\*\*您的选择是？\*\*\*\*\*\*\n");

printf("\t\t\t");

scanf("%d",&m);

if(m<0||m>2)

{

color(12);

printf("\t\t\t输入错误!请重新输入！！！\n");

color(0);

Sleep(2000);//暂停3秒

system("cls");

menu();

}

else

{

if(m==0)

{

printf("\t\t\t\*\*\*\*\*\*\*\*感谢您的使用，再见！！！\*\*\*\*\*\*\*\*\*\*\*\*");

exit(0);

}

else

switch(m)

{

case(1):

printf("\t\t\t请输入已创建的文本文件的路径 (如 D:\\novel.txt):\n");

printf("\t\t\t");

scanf("%s",name); //用户输入包含路径的文本文件名

printf("\t\t\t请输入要查找的单词数： \n");

printf("\t\t\t");

scanf("%d",&n); //用户输入要查找的关键字个数

printf("\t\t\t");

printf("请输入要查找的单词， 词与词之间用空格隔开 (区分大小写):\n");

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

{

printf("\t\t\t");

scanf("%s",&words[i][1]); //用 户 一 次 性 输 入 要 查 找 的 关 键 字 ，words[i][0]用于存放字符串的长度

}

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

find(name,words[i]);

printf("\n\t\t\t\*\*\*\*\*输入1返回菜单,或按下其他键退出系统\*\*\*\*\*\n");

printf("\t\t\t");

scanf("%d",&m);

if(m==1)

{

system("cls");

menu();

}

else

{

printf("\t\t\t\*\*\*\*\*\*感谢您的使用，再见！！！\*\*\*\*\*\*");

exit(0);

}

break;

case(2):

printf("\t\t\t请输入已创建的文本文件的路径 (如 D:\\novel.txt):\n");

printf("\t\t\t");

scanf("%s",name); //用户输入包含路径的文本文件名

fp=fopen(name,"r");

if (!(fp=(fopen(name,"r")))) //打开小说文件

{

printf("\t\t\t");

color(12);

printf("\n\t\t\t未找到该文件！！！\n");

color(0);

printf("\t\t\t");

printf("\n\t\t\t\*\*\*\*\*输入1返回菜单,或按下其他键退出系统\*\*\*\*\*\n");

printf("\t\t\t");

scanf("%d",&m);

if(m==1)

{

system("cls");

menu();

}

else

{

printf("\t\t\t\*\*\*\*\*\*感谢您的使用，再见！！！\*\*\*\*\*\*");

exit(0);

}

}

else

{

system("cls");

printf("\*\*\*\*\*\*\*\*\*\*该文本内容如下：\*\*\*\*\*\*\*\*\*\*\*\*\n");

while(!feof(fp))

{

putchar(ch);

fscanf(fp,"%c",&ch);

}

fclose(fp);

printf("\n\*\*\*\*\*输入1返回菜单,或按下其他键退出系统\*\*\*\*\*\n");

scanf("%d",&m);

if(m==1)

{

system("cls");

menu();

}

else

{

printf("\t\t\t\*\*\*\*\*\*感谢您的使用，再见！！！\*\*\*\*\*\*");

exit(0);

}

}

break;

}

}

}

### 2.2.2 字体颜色模块

（1）主要思想：

SetConsoleTextAttribute是Windows系统中一个可以设置控制台窗口字体颜色和背景色的计算机函数，该函数被广泛应用于软件开发中。使用此函数前，必须包含Windows.h头文件，即#include 。

该模块主要用于提示用户未找到指定文本或用户输入错误（将提示字体变红），用来提示用户选择正确菜单功能和输入正确的文本路径。

（2）具体代码如下：

void color(short x)//局部字体改变颜色

{

if (x==12)

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 12); //改变字体颜色为红色

Else //默认的颜色白色

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 7);

}

### 2.2.3 查询模块

（1）主要思想

本模块主要用于查询用户输入的关键词，输出其关键字在文本的位置和次数；若为找到指定文件，则输出“未找到该文件！！！”。本模块分为三个部分，一部分是求模式串 T的 next函数值并存入数组 next，一部分是利用模式串 T 的 next 函数求 T 在主串 S 中第 pos 个字符之后的位置的 KMP 算法，最后一部分将前两部分结合起来，利用模式串 T 的 next 函数求 T 在主串 S 中第 pos 个字符之后的位置的 KMP 算法，来输出关键字在指定文本中的位置和次数。

（2）具体代码如下：

void get\_next(SString T,int next[ ])//求模式串 T的 next函数值并存入数组 next

{

int j=1,k=0;

next[1]=0;

while (j<T[0])

{

if (k==0 || T[k]==T[j])

{

++j; ++k;

if (T[j]!=T[k]) next[j]=k;

else next[j]=next[k];

}

else k=next[k];

}

}

int Index(SString S,SString T,int pos)// 利用模式串 T 的 next 函数求 T 在主串 S 中第 pos 个字符之后的位置的 KMP 算法

{

int i=pos,j=1;

while (i<=S[0]&&j<=T[0])

{

if (j==0||S[i]==T[j]) {++i;++j;} // 继续比较后继字符

else j=next[j]; // 模式串向右移动

}

if (j>T[0]) return (i-T[0]); //匹配成功

else return 0; // 匹配失败

}

int length(SString str) //求串长

{

int i=1;

while(str[i]) i++;

return(i-1);

}

void find(char name[],SString keys) //对于输入的每一个要查找的关键字， 从小说文件中逐行读取字符串查找

{

int m;

char ch;

SString text; //存放从小说文件读取的一行字符串

int i=1,j=0,k; //i 用于存放行号， j 用于存放列号,k 用于输出格式的控制

int n=0; //n 记录出现次数

FILE \*fp;

if (!(fp=(fopen(name,"r+")))) //打开小说文件

{

printf("\t\t\t");

color(12);

printf("\n\t\t\t未找到该文件！！！\n");

color(0);

printf("\t\t\t");

printf("\n\t\t\t\*\*\*\*\*输入1返回菜单,或按下其他键退出系统\*\*\*\*\*\n");

printf("\t\t\t");

scanf("%d",&m);

if(m==1)

{

system("cls");

menu();

}

else

{

printf("\t\t\t");

printf("\*\*\*\*\*\*感谢您的使用，再见！！！\*\*\*\*\*\*");

exit(0);

}

}

keys[0]=length(keys); //调用 length 函数求关键字的长度

get\_next(keys,next); //调用 get\_next 函数求模式串（关键字）每一个字符对应的 next 值

printf("\t\t\t");

printf("\n\t\t\t%s出现于:\n",&keys[1]); //打印关键字

while(!feof(fp)) //如果还没到小说文件末尾

{

k=0;

fgets(&text[1],maxsize,fp);//从小说文件中读取一行字符串， 存入 text 串中

text[0]=length(text); //求读入的串的长度

j=Index(text,keys,j+1); //调用 KMP 算法， 统计关键字在该行出现的位置， 若匹配不成功则返回 0

if (j!=0)

{ printf("\t\t\t行=%d,列=%d",i,j); k++; n++;} //若匹配成功则打印行号和列号

while(j!=0) //若该行找到了关键字， 则继续寻找看是否还能匹配成功

{

j=Index(text,keys,j+1); //调用 KMP 算法从刚找到的列号后一字符起匹配

if (j!=0)

{n++;printf(",%d",j);} //若匹配成功， 则打印列号

}

i++; //行号加 1， 在下一行中寻找

if(k) printf("\n"); //输出格式控

}

printf("\t\t\t");

printf("%s 共出现%d 次\n",&keys[1],n);

}

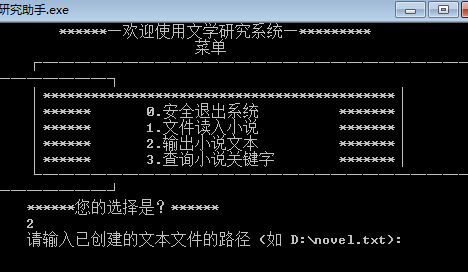
# 系统运行与测试

3.1系统测试与修改

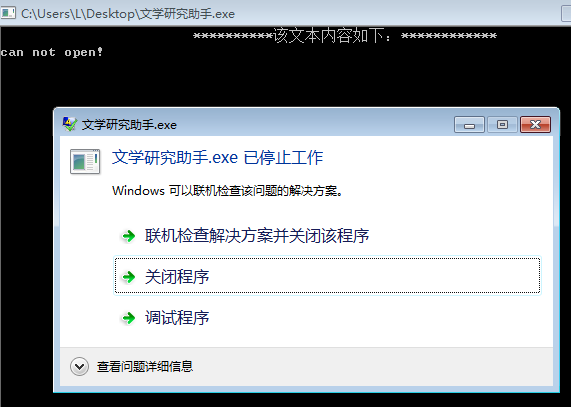
一个完整的，可靠的，安全的系统，往往要经过成千上万次测试，系统测试本身的目的是尽可能多地暴露程序中的错误，但是找出错误的最终目的还是为了改正错误，使系统更加可靠。开发系统根本目标是以较低的成本开发出高质量且完全符合用户要求的系统软件，因此在设计完整个系统之后，还需要进一步诊断和改正程序中的错误，这就是系统测试的任务。系统测试工作是在整个系统设计过程中十分重要的一个环节，关系到本设计最终的成败。

本系统经过多次的修改，下面只列出关键性的错误，用于提高读者的编程思想和编程能力。

该模块设计之初，具有一个很严重的问题：在用户选择输出小说文本时，输入了根本不存在的文本路径，从而导致系统中断停止工作，出现此问题的原因是在用户输入文本路径时，程序并没有检查该文本是否存在，导致系统中断。具体如图***图 3‑1***，***图 3‑2***所示。



***图 3‑1***



***图 3‑2***

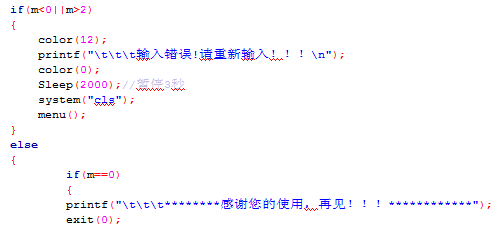
此情况虽然合理，但是不利于系统的稳定性，使用户的使用效果大大降低。

解决该问题的方式是在用户输入文本路径后，加入读取文本函数即可解决该问题。具体代码如***图 3‑3***所示。



***图 3‑3***

该问题解决后，有利于系统的稳定性，且加强的用户的可操作性，使本系统具有更加完善的功能。另外，在系统设计之初，当用户输入0,1,2以外的数字时，系统并不会响应，原因是未在用户输入后系统并未判断输入是否合法，修改后代码如***图 3-4***所示。

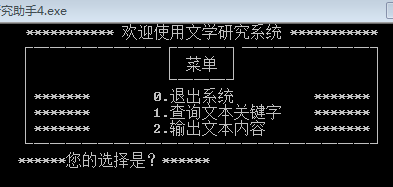
****

***图 3‑4***

3.2系统运行

### 3.2.1 菜单模块运行界面

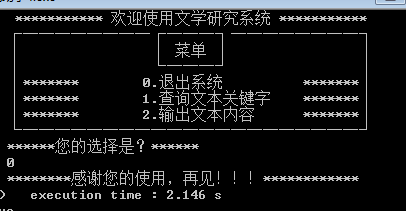
为了实现文学研究系统各功能的管理，需要设计一个含有多个菜单项的主控菜单子程序用以链接系统的各项子功能，方便用户的使用。本系统主控菜单运行界面下图***图 3‑5***所示。



***图 3‑5***

### 3.2.2退出系统

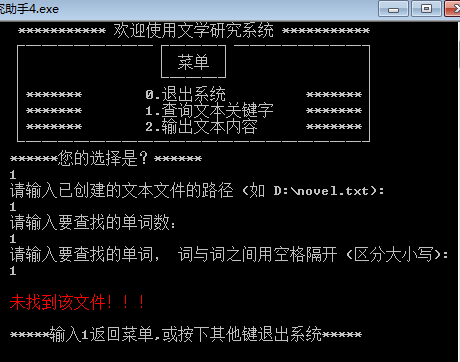
用户输入则退出该文学研究系统。运行结果如下图***图 3‑6。***



***图 3‑6***

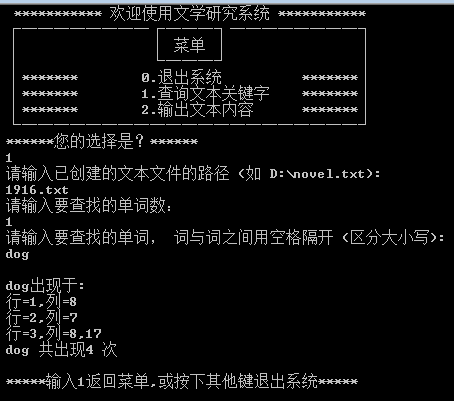
### 3.2.3查询文本关键字

用户输入需要查询的文本路径，若该路径不存在，则提示用户未找到该文件（如***图 3‑7***），同时让用户做出选择（返回菜单or退出系统）;



***图 3‑7***

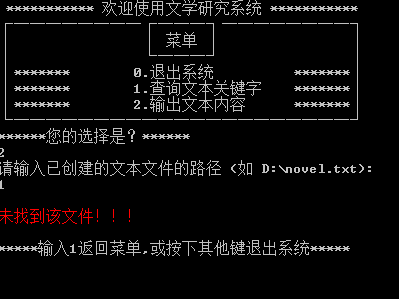
若文件存在，则根据用户输入的关键词查询其位置和次数，并将其输出，最后让用户做出选择（返回菜单or退出系统）（如***图 3‑8***）。



***图 3‑8***

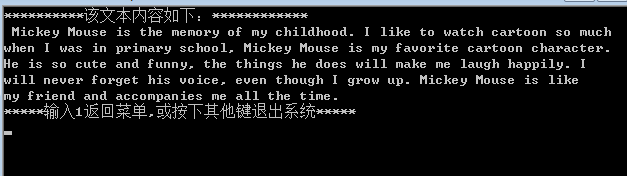
### 3.2.4 输出文本内容

当用户输入需要输出文本内容的文本路径，若该路径不存在，则提示用户未找到该文件（如***图3‑9***），同时让用户做出选择（返回菜单or退出系统）;

******

***图 3‑9***

若文件存在，则输出其内容，最后让用户做出选择（返回菜单or退出系统）（如***图 3‑10***）。



***图 3‑10***

结论

文学研究助手与模式匹配算法KMP完成了在用户指定的文本最后查询用户输入的关键词所在的位置及其总次数，通过运用模式匹配算法KMP，利用匹配失败后的信息，尽量减少模式串与主串的匹配次数以达到快速匹配的目的，从而快匹配效率，更快的为用户找出所需内容，提高工作效率。

经过多次的检验，最终，较好的实现了用户所需功能，同时增加了局部字体变色，程序延时，用户循环使用，输出文本内容等功能。

该文学研究系统采用了递归思想，较好的实现了一个系统应该具备的功能，循环使用以及随时退出，以便于用户的使用效果和操作。

该系统的界面设计也是很简明美观，提示语到位，易于用户理解与使用。

然而，系统也存在一些不足之处：

1. 系统的安全性无法保障，每个拥有该程序的人，都可以使用和修改程序，后期可增加登陆系统及其系统的封装和加密，用以反编译，提高系统的安全性。
2. 该系统目前只能读取txt文本文件，使用范围过小，不利于系统的推广。
3. 注释应尽量详细，详细的注释不仅能够让看的人清晰明了，对编程者本人的思路整理也有好处，使得在编程的过程中不易产生混乱。思路是编程的核心，在编程之前一定要理清思路，这样编程时才有理有据，事半功倍。

参考文献

【1】严蔚民，吴伟明：《数据结构（C语言版）》【M】，清华大学出版社2007年版，第80-84页。

【2】严蔚敏，吴伟民，米宁：数据结构题集（C语言版），清华大学出版社，1999.2

【3】汪沁.奚李峰.数据结构与算法[M].北京:清华大学出版社,2012.9

【4】唐国文.数据结构课程设计[M]. 北京:清华大学出版社,2010.9

致谢

这次课程设计在一定程度上锻炼了我的学习能力、提高了我的编程能力和资料的查找能力。每一个模块流程图的绘画，都引领着我去思考这个模块代码的优缺点及可行性；每一次调试的错误，都促使着我去查找原因的热情与耐心；每一次界面的美观修改，都是我追求进步的信心；每一次运行的成功，都荡漾着我内心的喜悦……这一切的一切，都源于课程设计的开始。我感谢这次课程设计，感谢学院开设这门实践课，因为它的开设增加了一份我对代码更深层的热爱。

也同样非常感谢老师建设性的指导与耐心解答，协助我顺利地完成了此次课程设计。

最后再次感谢学院，提供了一个干净整洁舒适的机房让我们进行实践课程。

感谢所有在我需要帮助的时候帮助我的人，谢谢！

附录

/\*\*\*\*\*\*\*\*文学研究助手与模式匹配算法KMP\*\*\*\*\*\*\*\*/

在此处贴上整个系统的所有代码(代码可贴可不贴)