**湖南工业大学**

**课程设计**

**资 料**

|  |  |  |
| --- | --- | --- |
| 学年学期 | ： | 2021-2022-2 |
| 课程名称 | ： | 数据结构课程设计 |
| 题目 | ： | 文章编辑 |
| 起止日期 | ： | 2022年06月20日 ～ 2022年06月26日 |
| 学院 | ： |  |
| 班级 | ： |  |
| 学生姓名 | ： |  |
| 学号 | ： |  |

目 录 清 单

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 材 料 名 称 | 资料数量 | 备 注 |
| 1 | 课程设计任务书 | 1 |  |
| 2 | 课程设计（实训）报告 | 1 |  |
| 3 | 设计文档和源程序（电子文档） | 1 |  |
| 4 | 系统演示 |  |  |

课程设计任务书

学 院： 学生姓名： 专业班级：

课程名称： 数据结构课程设计 设计题目： 文章编辑

完成期限：自 2022 年 06 月 20 日至 2022 年 06 月 26 日共 1 周

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 内  容  及  任  务 | 一、设计的内容及主要技术参数  输入一页文字，程序可以统计出文字、数字、空格的个数。要求分行输出用户输入的各行字符，分4行输出"全部字母数"、"数字个数"、"空格个数"、"文章总字数"，输出删除某一字符串后的文章。  二、设计任务  ①分别统计出其中英文字母数和空格数及整篇文章总字数；②统计某一字符串在文章中出现的次数，并输出该次数；③删除某一子串，并将后面的字符前移；④存储结构使用线性表，分别用几个子函数实现相应的功能；  三、设计工作量  1、 结点结构的设计  2、 算法设计与分析  3、 程序设计、实现、调试  4、 课程设计说明书 | | | | |
| 进  度  安  排 | 起止日期 | | 工作内容 | | |
| 2022-06-20 | | 查阅文献资料，选题。 | | |
| 2022-06-21~2022-06-22 | | 数据结构设计，编码，算法的实现，运行结果分析 | | |
| 2022-06-23~2022-06-26 | | 完成课程设计说明书，系统演示，确定分数，交文档材料。 | | |
| 主  要  参  考  资  料 | [1] 刘畅，姚学峰. 数据结构[M]. 上海交通大学出版社，2016.  [2] 陈越，何钦铭等. 数据结构[M]. 北京：高等教育出版社，2012.  [3] 耿国华. 数据结构——用C语言描述. 北京：高等教育出版社，2011.  [4] 孙凌，李丹. 数据结构[M]. 北京：人民邮电出版社，2010. | | | | |
| 同组信息 | 学号 | 姓名 | | 分工负责的内容 | 备注 |
|  |  | | 选题，数据结构设计，算法实现，结果分析 | 组长 |

指导教师（签字）： 年 月 日

系（教研室）主任（签字）： 年 月 日

课程设计报告

|  |  |  |
| --- | --- | --- |
| 学年学期 | ： | 2021-2022-2 |
| 课程名称 | ： | 数据结构课程设计 |
| 题目 | ： | 文章编辑 |
| 起止日期 | ： | 2022年06月20日 ～ 2022年06月26日 |
| 学院 | ： |  |
| 班级 | ： |  |
| 学生姓名 | ： |  |
| 学号 | ： |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 成绩评定 | ： |  |  |
| 课程目标 | | 成绩 | 评定人签名 |
| 目标1 | |  |  |
| 目标2 | |  |  |
| 目标3 | |  |  |

**2022 年 6 月**

目录

[第1章 绪论 1](#_Toc6132)

[1.1 文章编辑的介绍 1](#_Toc2752)

[1.1.1选题的意义 1](#_Toc23667)

[1.1.2设计内容与要求 1](#_Toc12133)

[1.2 文章编辑的技术准备 1](#_Toc18169)

[1.2.1相关技术介绍 1](#_Toc32324)

[1.2.2设计过程与进度安排 2](#_Toc3804)

[第2章 选题分析与设计 3](#_Toc21278)

[2.1选题分析 3](#_Toc6124)

[2.1.1 面临的问题 3](#_Toc31155)

[2.1.2 解决方案 3](#_Toc21750)

[2.2 流程图设计 3](#_Toc19925)

[2.2.1 功能模块图 3](#_Toc24052)

[2.2.2 程序执行流程图 4](#_Toc12329)

[第3章 选题实现与演示 6](#_Toc17468)

[3.1 存储文章模块 6](#_Toc9218)

[3.1.1 存储文章模块的实现 6](#_Toc7077)

[3.1.2 存储文章模块的演示效果 8](#_Toc28425)

[3.2 文章统计模块 8](#_Toc31348)

[3.2.1 文章统计模块的实现 8](#_Toc16345)

[文章统计模块的演示效果 9](#_Toc23309)

[3.3 文章查找模块 10](#_Toc23948)

[3.3.1 文章查找模块的实现 10](#_Toc10877)

[3.3.2 文章查找模块的演示效果 13](#_Toc29341)

[3.4 文章删除模块 15](#_Toc17875)

[3.4.1 文章删除模块的实现 15](#_Toc24609)

[3.4.2 文章删除模块的演示效果 16](#_Toc20827)

[第4章 总结 18](#_Toc2412)

[4.1 文章编辑问题之一 18](#_Toc6505)

[4.1.1 问题描述 18](#_Toc28308)

[4.1.2 解决过程或方法 18](#_Toc5791)

[4.2 文章编辑问题之二 18](#_Toc17160)

[4.2.1 问题描述 18](#_Toc16033)

[4.2.2 解决过程或方法 19](#_Toc15878)

[4.3 心得与体会 19](#_Toc19256)

# 

# 第1章 绪论

## 1.1 文章编辑的介绍

### 1.1.1选题的意义

### 本课程设计主要解决了基本的文章编辑问题。对于一页按要求输入到程序执行窗口的文章，静态存储结构使用线性表中，分别完成统计文章总字符数、文章中空格个数、文章中英文字母数、统计文章中英文字母数的操作和文章中某一字符串的出现的次数，并输出该次数：除以上的统计操作，还实现删除文章中指定字符串，并且将删除字符串后面的字符前移，再保存为编辑后的文章。

### 生活中或学术上有很多类似的问题，这个问题看似简单，却可能使很多重大问题的重要组成部分或者是核心。解决此问题的目的是以便在生活中解决根本性问题，使生活变得更加便利。

### 1.1.2设计内容与要求

## 静态存储一页文章，每行最多不超过80个字符，共N行；要求①分别统计出其中英文字母数和空格数及整篇文章总字数；②统计某一字符串在文章中出现的次数，并输出该次数；③删除某一子串，并将后面的字符前移；④存储结构使用线性表，分别用几个子函数实现相应的功能；

## 输入数据的形式和范围：可以输入大写、小写的英文字母、任何数字及标点符号。

## 输出形式：①分行输出用户输入的各行字符；②分4行输出"全部字母数"、"数字个数"、"空格个数"、"文章总字数"；③输出删除某一字符串后的文章。

## 1.2 文章编辑的技术准备

### 1.2.1相关技术介绍

文章编辑采用线性表作为存储结构。线性表是一种典型的数据结构, 线性结构的基本特点是线性表中的数据元素是有序且有限的, 在线性结构中, 有且仅有一个被称为"开始数据元素"和一个"最后数据元素", 除了开始数据元素没有直接前驱, 最后一个数据元素没有直接后继外, 其余的数据元素有且仅有唯一的一个直接前驱和直接后继。整理下来说, 线性表具有如下基本特征:（1）线性结构中必然存在唯一一个"开始数据元素" ;（2）线性结构中必然存在唯一一个"最后数据元素" ; （3）除第一个数据元素外, 其他元素均有唯一一个直接前驱;（4）除最后一个数据元素外, 其他元素均有唯一一个直接后继 。

### 1.2.2设计过程与进度安排

设计过程：

根据题目要求，可分将程序分为四大模块：存储文章模块、文章统计模块、文章查找模块、文章删除模块，以及六大功能：统计文章总字符数、删除文章中指定的字符串、统计文章指定字符串出现次数、构造线性表，将每一行字符都存储到线性表中。

对于文本的输入，采用头插法将文本信息存储到链表已经申请好的空间中。对于文本内容的设计，使用循环对已存储的文章进行匹配，大写字母数、小写字母数、空格数、数字数直接通过条件比囧即可得到。对于文本内容的处理，查找部分仍是使用循环对已存储的文章进行匹配，判断需要查找的字符或字符串是否与文章中某部分相同，如果存在相同的记录相同的个数，删除部分先使用程序的查找功能对文章中需要删除的字符或者字符串进行查找，然后对其进行删除。文章输入结束后，程序输出命令菜单，列出程序可以合法执行的操作命令，并选择输入合法的指令，以执行相应的指令操作。

进度安排：

|  |  |
| --- | --- |
| 2022-06-20 | 查阅文献资料，选题。 |
| 2022-06-21~2022-06-22 | 数据结构设计，编码，算法的实现，运行结果分析 |
| 2022-06-23~2022-06-26 | 完成课程设计说明书，系统演示，确定分数，交文档材料。 |

# 第2章 选题分析与设计

## 2.1选题分析

### 2.1.1 面临的问题

1.如何利用线性表实现多行存储文章

2.怎样判断结束输入文章

3.如何防止指针非法越界

### 2.1.2 解决方案

1.将文章的每一行用字符数组进行存储，判断每一行的字符数量是否超过80个字符，再将每一行字符串用线性表连接起来。

2.以特殊字符ctrl+E(^E)作为文章结束的标志，程序终止文章存储。

3.利用变量flag标记线性表中是否存储了数据项，防止未存储数据项就进行操作造成的程序异常关闭。

## 2.2 流程图设计

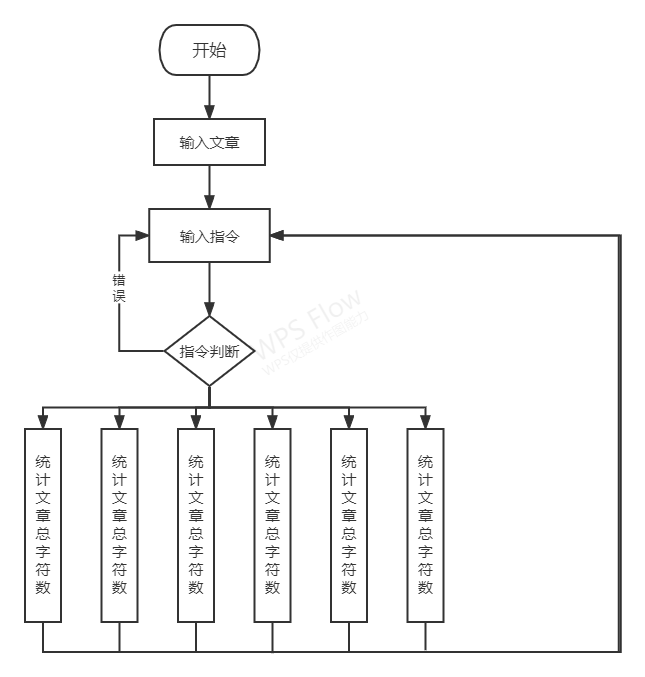
### 2.2.1 功能模块图

### 1

图2.1功能模块图

功能说明：  
输出原文：此功能是将原先录入的信息读出，并在屏幕上显示出来  
统计文章字数：此功能是统计出文章中的字母数，空格数，数字数和文章总字数  
统计某一字符串在文章中出现的次数：通过用户输入要查询项来统计要查询的  
字符在文章中的出现次数。  
删除某一字符串并输出文章：删除文章中用户希望删除的字符，并将修改后的  
文章在屏幕上显示出来。  
结束：退出程序。

### 2.2.2 程序执行流程图



# 第3章 选题实现与演示

## 3.1 存储文章模块

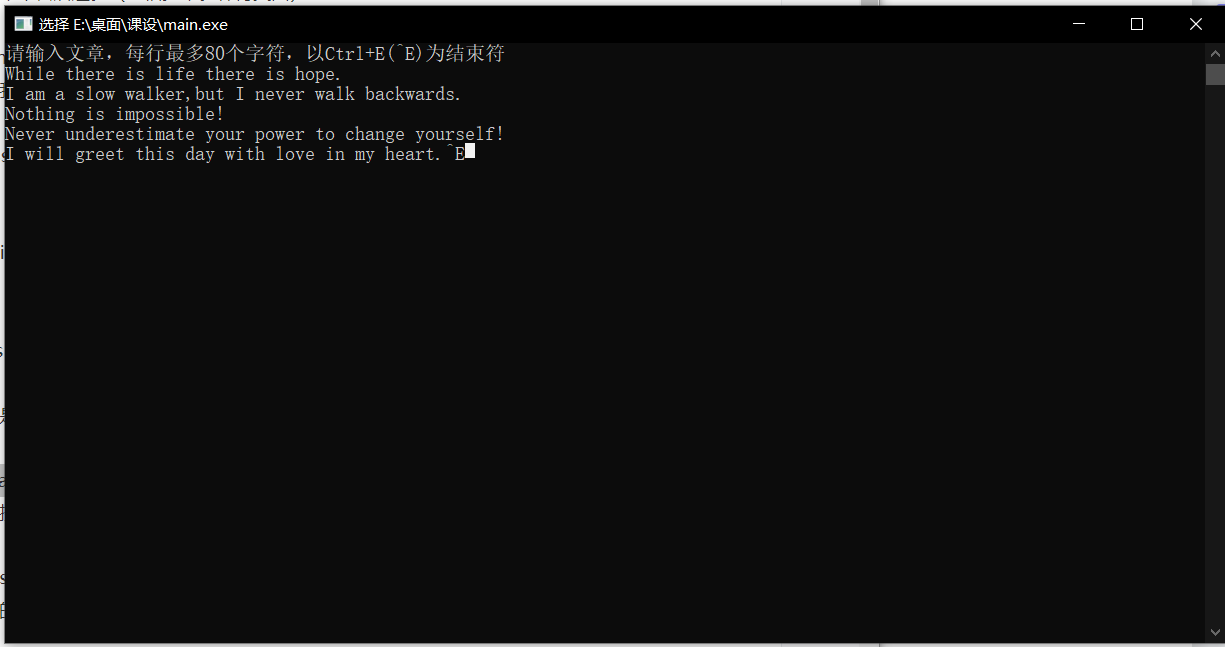
### 3.1.1 存储文章模块的实现

void Add(article \*&p) {  
    printf ("请输入文章，每行最多80个字符，以Ctrl+E(^E)为结束符\n");  
    article \*q = new article;//首先为链表建立一个附加表头结点  
  
    p = q;//表头指针  
  
    char tmp[100];  
  
    while (1)  
    {  
        gets(tmp);//输入字符串  
        if (strlen(tmp) > 80)  
        {  
            printf("每行最多输入80个字符!");  
            continue;  
        }  
  if (tmp[0] == 5)break;//如果发现输入^E，则退出输入  
  
        q = q->next = new article;  
  
        q->word = new char[strlen(tmp) + 1];//为结点分配空间  
  
        strcpy(q->word, tmp);  
  
        if (tmp[strlen(tmp) - 1] == 5)//除去最后一个控制符^E  
  
        {  
            q->word[strlen(tmp) - 1] = '\0';  
            break;  
        }  
    }  
    q->next = NULL; //最后一个指针为空  
    p = p->next;  
    printf("\n");  
    flag=1;  
}

* 表题行：中文黑体五号，居中，段前段后各0.5行，与下段同页，表序按章内连续编号。
* 三线表：上下边线1磅，中间细线1/4磅（打印后或增大显示比例可分辨），其余用无框线，无斜线；
* 复杂表格可拆分成多个简单表格。尽量避免表格跨页，如不得已，则应在下一的上方标明“续前表：”（该行按正文格式，段后0.5行）。补充表头和边线。
* 几句话能表达清楚的事项，就不必用表格。

### 3.1.2 存储文章模块的演示效果

## 

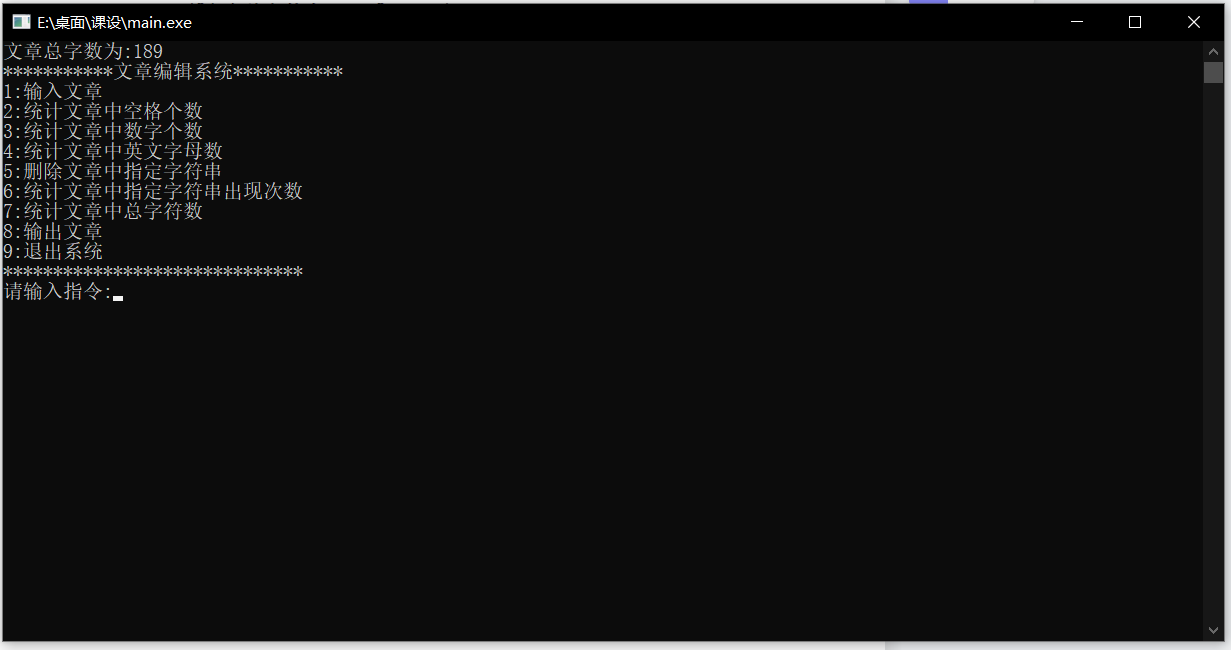


## 3.2 文章统计模块

### 3.2.1 文章统计模块的实现

### **void** Length**(**article **\*&**p**)** **{**     **if(!**flag**){**         cout**<<"请输入文章！"<<**endl**;**         **return** **;**     **}**     **int** sum **=** 0**;**     article **\***q **=** p**;**     **while** **(**q **!=** NULL**)** **{**         sum **+=** strlen**(**q**->**word**);**         q **=** q**->**next**;**     **}**     printf**("文章总字数为:%d**\n**",** sum**); }** 3.2.2 文章统计模块的演示效果

## 

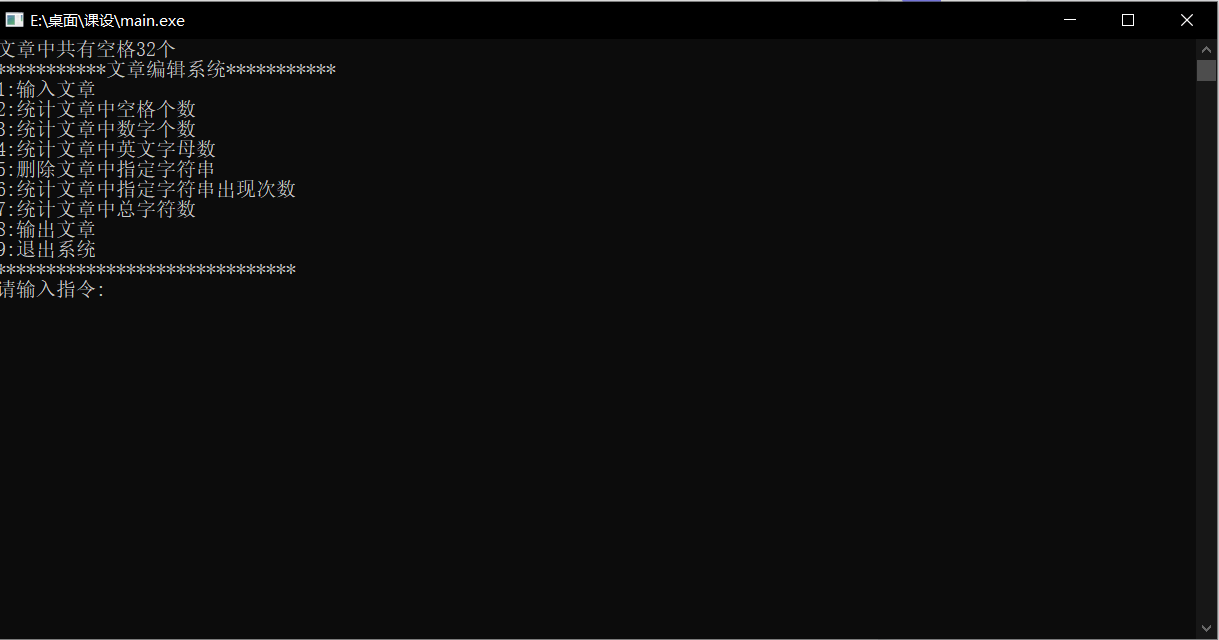
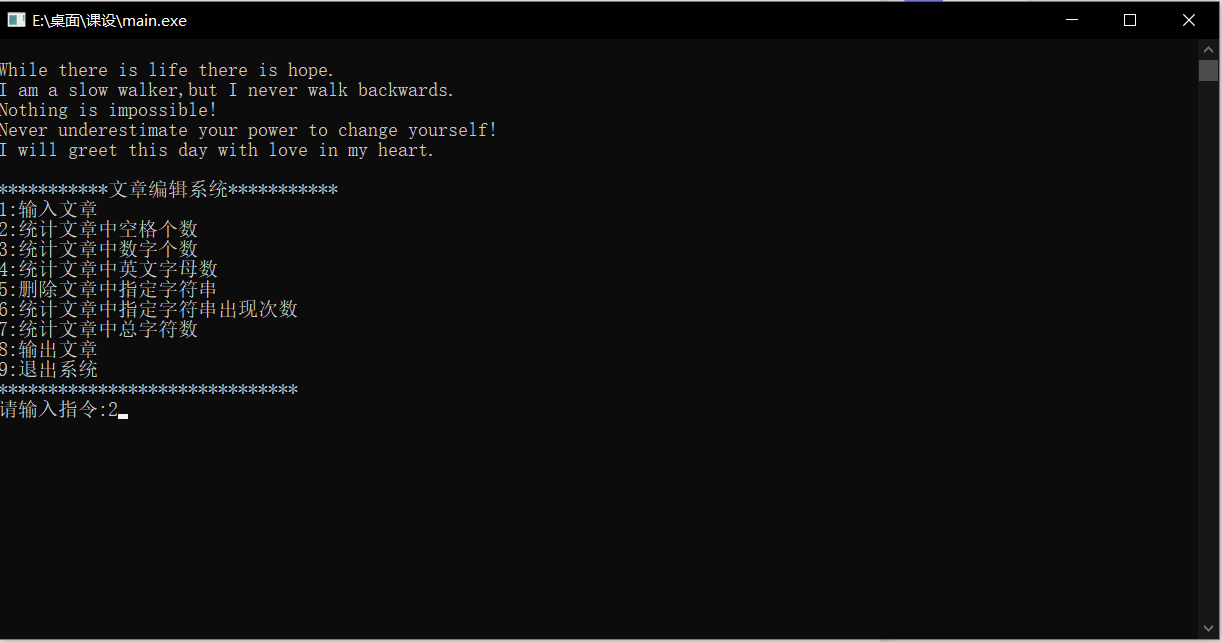


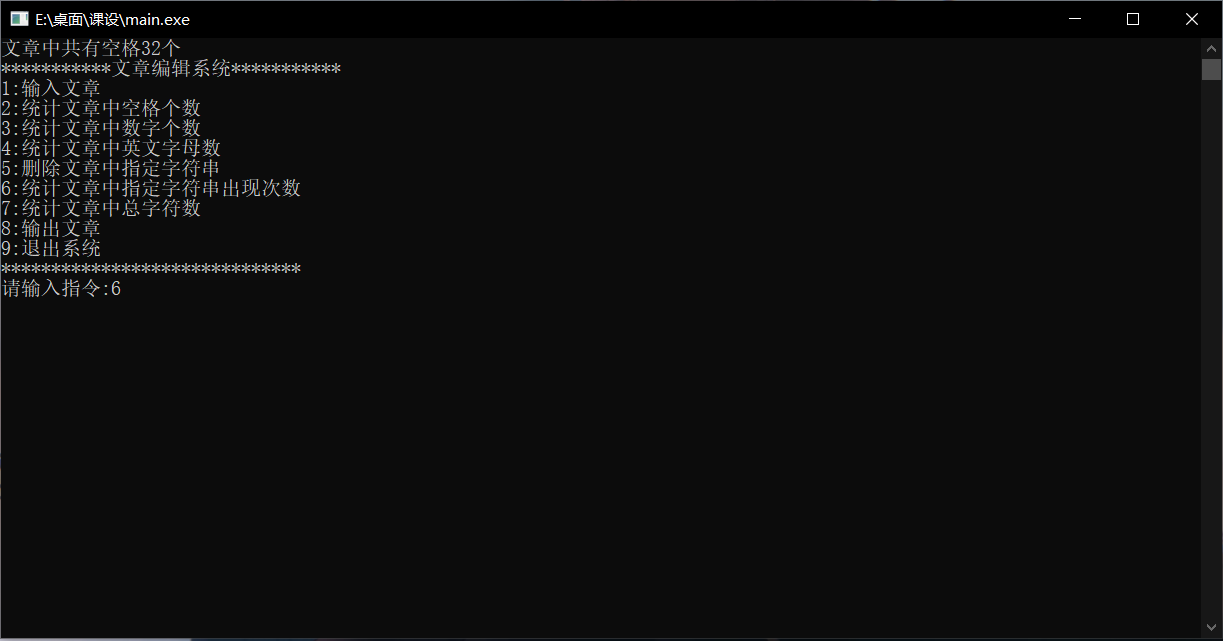
## 3.3 文章查找模块

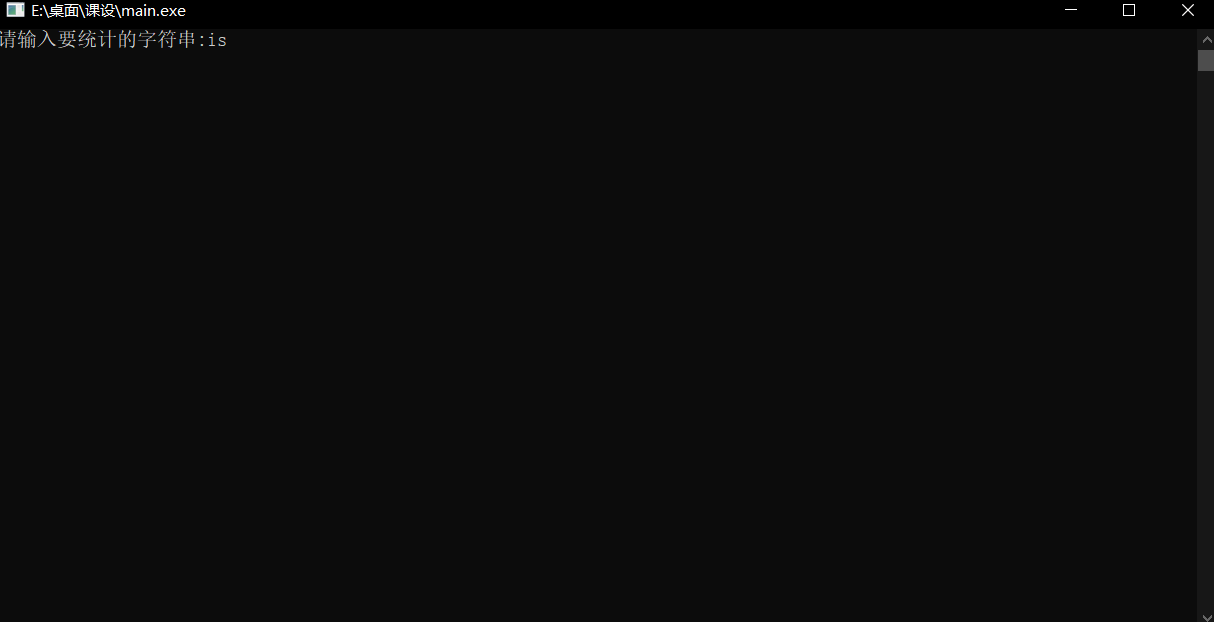
### 3.3.1 文章查找模块的实现

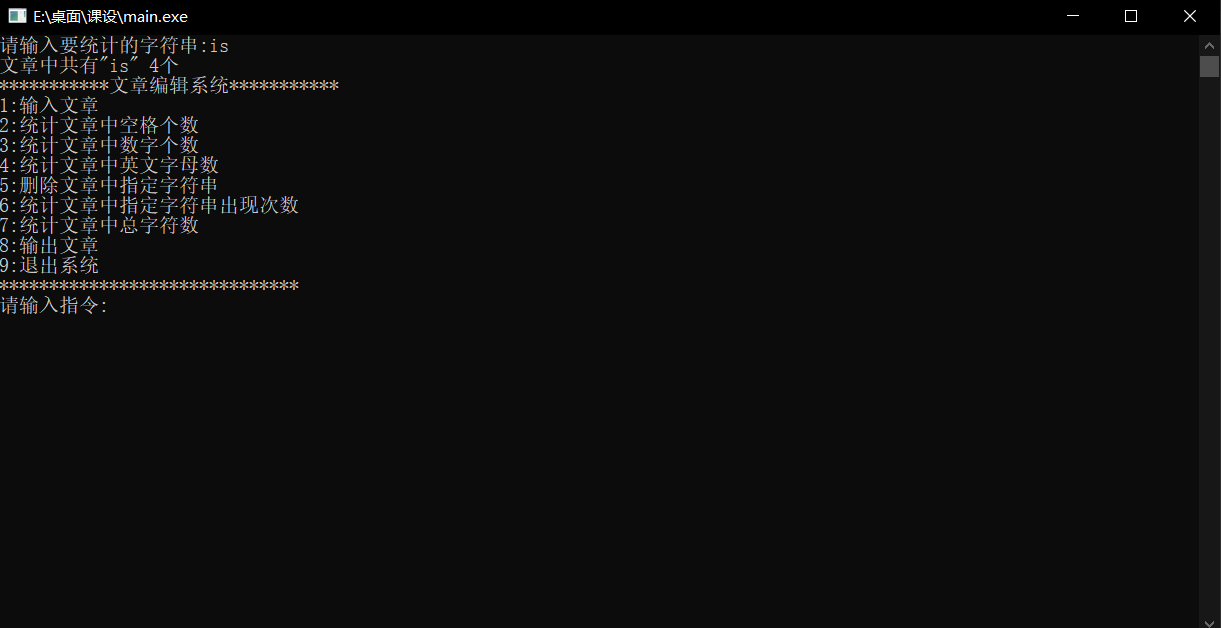
**int** count1**(**article **\*&**p**){**    article **\***q**=**p**;**    **int** count**=**0**;**    **do{**        **int** Len**=**strlen**(**q**->**word**);**        **for(int** i**=**0**;**i**<**Len**;**i**++)**            **if(**q**->**word**[**i**]==**' '**)**count**++;**    **}**    **while((**q**=**q**->**next**)!=**NULL**);**    **return** count**;  
}  
int** count2**(**article **\*&**p**){**    article **\***q**=**p**;**    **int** count**=**0**;**    **do{**        **int** Len**=**strlen**(**q**->**word**);**        **for(int** i**=**0**;**i**<**Len**;**i**++)**            **if(**q**->**word**[**i**]<=**'9'**&&**q**->**word**[**i**]>=**'0'**)**count**++;**    **}**    **while((**q**=**q**->**next**)!=**NULL**);**    **return** count**;  
}  
int** count3**(**article **\*&**p**){**    article **\***q**=**p**;**    **int** count**=**0**;**    **do{**        **int** Len**=**strlen**(**q**->**word**);**        **for(int** i**=**0**;**i**<**Len**;**i**++)**            **if((**q**->**word**[**i**]<=**'z'**&&**q**->**word**[**i**]>=**'a'**)||(**q**->**word**[**i**]<=**'Z'**&&**q**->**word**[**i**]>=**'A'**))**count**++;**    **}**    **while((**q**=**q**->**next**)!=**NULL**);**    **return** count**;  
}  
int** count4**(**article **\*&**p**,char** **\***s**){**    article **\***q**=**p**;**    **int** count**=**0**;**    **int** len2**=**strlen**(**s**);**    **do** **{**        **int** len1**=**strlen**(**q**->**word**);**        **for(int** i**=**0**;**i**<**len1**;**i**++)**        **{**            **if(**q**->**word**[**i**]==**s**[**0**])**            **{**                **int** k**=**0**;**                **for(int** j**=**0**;**j**<**len2**;**j**++)**                    **if(**q**->**word**[**i**+**j**]==**s**[**j**])** k**++;**                **if(**k**==**len2**)**    **{**count**++;**i**=**i**+**k**-**1**;}**            **}**        **}**    **}**    **while((**q**=**q**->**next**)!=**NULL**);***//遍历链表*    **return** count**;  
}  
void** Count**(**article **\*&**p**,** **int** num**)** **{**    **if(!**flag**){**        cout**<<"请输入文章！"<<**endl**;**        **return** **;**    **}**    **int** sum **=** 0**;**    **switch** **(**num**)** **{**    **case** 1**:**        sum **=** count1**(**p**);**        printf**("文章中共有空格%d个**\n**",**sum**);**        **break;**    **case** 2**:**        sum **=** count2**(**p**);**        printf**("文章中共有数字%d个**\n**",**sum**);**        **break;**    **case** 3**:**        sum **=** count3**(**p**);**        printf**("文章中共有字母%d个**\n**",**sum**);**        **break;**        **default:**            **char** str**[**80**];**            printf**("请输入要统计的字符串:");**            cin**>>**str**;**            sum **=** count4**(**p**,**str**);**            printf**("文章中共有**\"**%s**\"**%d个**\n**",**str**,**sum**);**            **break;**    **}  
  
}**

### 3.3.2 文章查找模块的演示效果







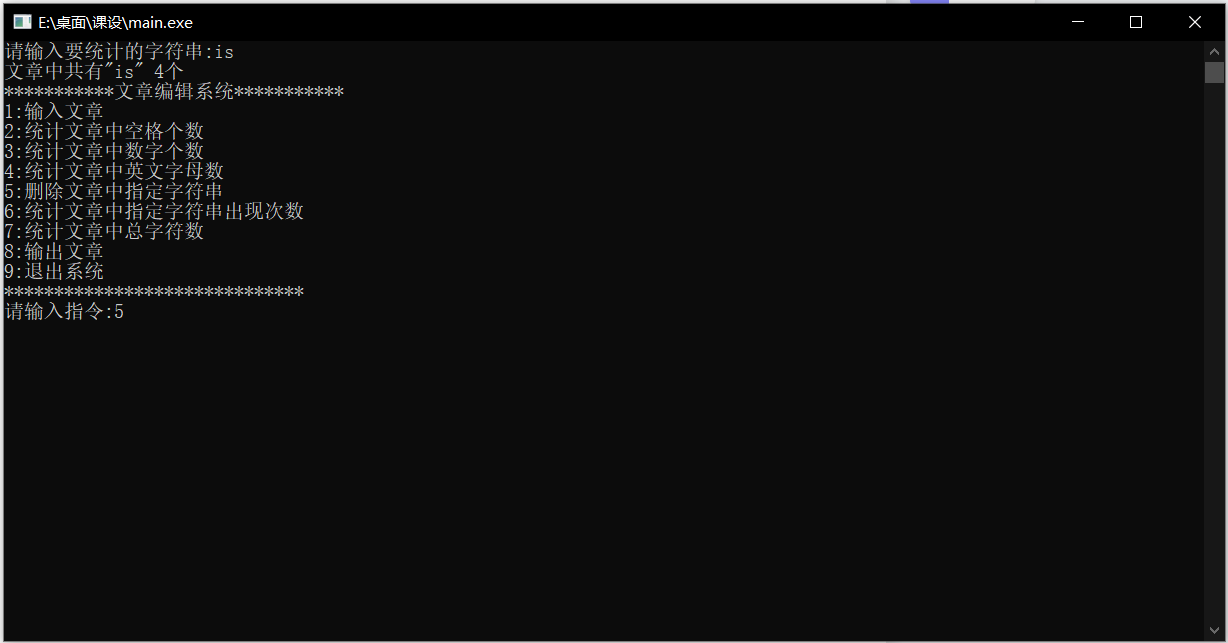


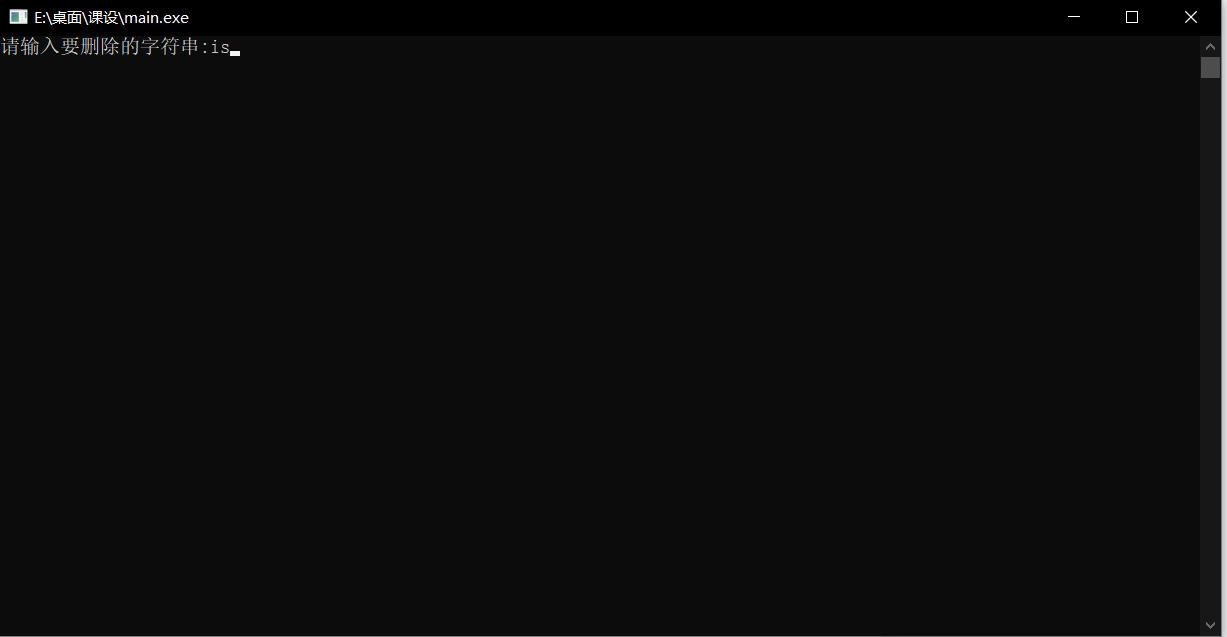
## 3.4 文章删除模块

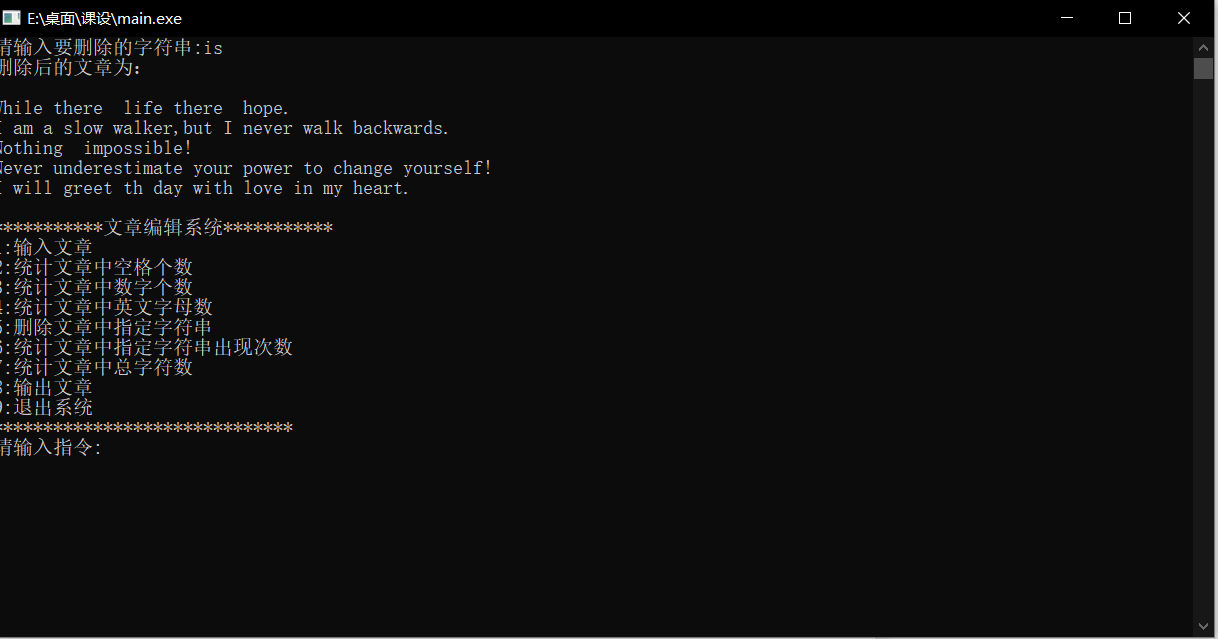
### 3.4.1 文章删除模块的实现

**void** delstringword**(char** **\***s**,char** **\***str**)***//删除指定的字符串***{***//\*s为输入的字符串，\*str为将要删除的字符*    **char** **\***p**=**strstr**(**s**,**str**);** *//从字符串中寻找str第一次出现的位置*    **char** tmp**[**80**];**    **int** len**=**strlen**(**s**);**    **int** i**=**len**-**strlen**(**p**);**    **int** j**=**i**+**strlen**(**str**);**    **int** count**=**0**;**    **for(int** m**=**0**;**m**<**i**;**m**++)**tmp**[**count**++]=**s**[**m**];**    **for(int** n**=**j**;**n**<**len**;**n**++)**tmp**[**count**++]=**s**[**n**];**    tmp**[**count**]=**'\0'**;**    strcpy**(**s**,**tmp**);  
  
}  
void** Delete**(**article **\*&**p**){**    **if(!**flag**){**        cout**<<"请输入文章！"<<**endl**;**        **return** **;**    **}**    article **\***q**=**p**;**    **char** str**[**80**];**    cout**<<"请输入要删除的字符串:";**    cin**>>**str**;**    **do**    **{**        **if(**strstr**(**q**->**word**,**str**)!=**NULL**)**delstringword**(**q**->**word**,**str**);**    **}**    **while((**q**=**q**->**next**)!=**NULL**);**    cout**<<"删除后的文章为：**\n**";**    Print**(**p**);  
}**

### 3.4.2 文章删除模块的演示效果







# 第4章 总结

## 4.1 文章编辑问题之一

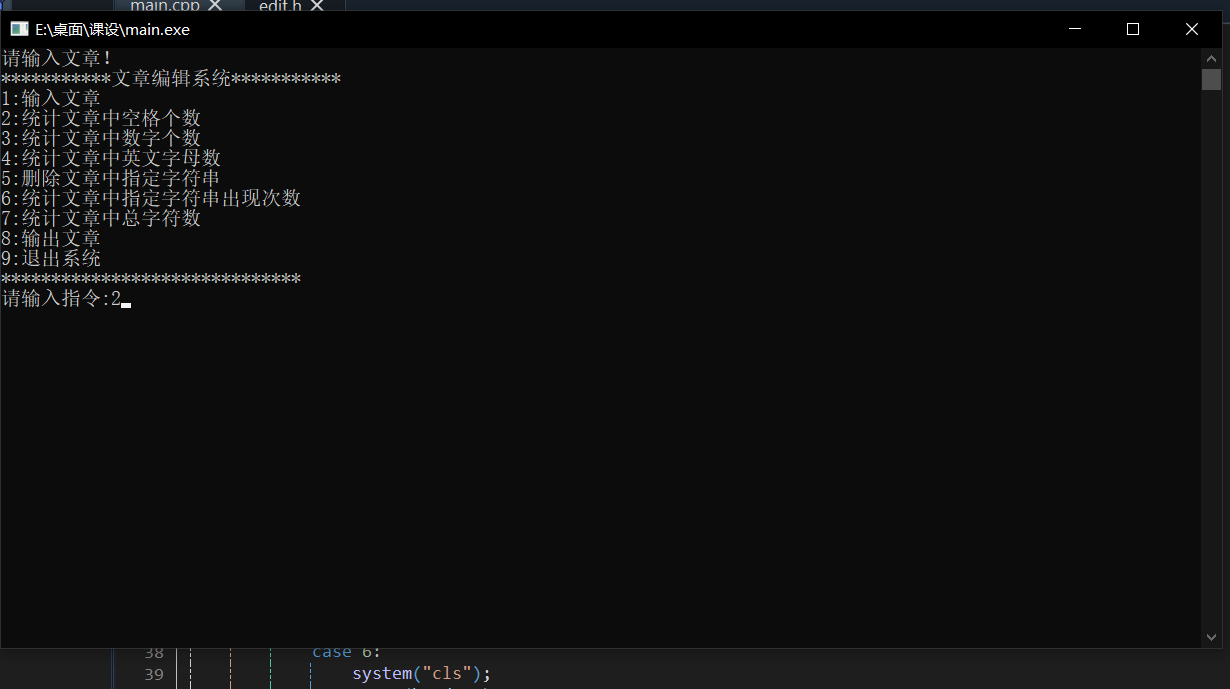
### 4.1.1 问题描述

* 表题行：中文黑体五号，居中，段前段后各0.5行，与下段同页，表序按章内连续编号。
* 三线表：上下边线1磅，中间细线1/4磅（打印后或增大显示比例可分辨），其余用无框线，无斜线；
* 复杂表格可拆分成多个简单表格。尽量避免表格跨页，如不得已，则应在下一页，表的上方标明“续前表：”（该行按正文格式，段后0.5行）。补充表头和边线。
* 几句话能表达清楚的事项，就不必用表格。

当未输入文章时，线性表中的内容为空，这时候如果进行统计、查找、删除等操作时，可能会造成指针非法访问导致程序异常终止。

### 4.1.2 解决过程或方法

用一个flag标志判断是否有执行文章输入，如果为flag值为1说明线性表中存在数据，可以进行操作，若为0，说明还未存储文章，给出提示并返回主界面。



## 4.2 文章编辑问题之二

### 4.2.1 问题描述

* 表题行：中文黑体五号，居中，段前段后各0.5行，与下段同页，表序按章内连续编号。
* 三线表：上下边线1磅，中间细线1/4磅（打印后或增大显示比例可分辨），其余用无框线，无斜线；
* 复杂表格可拆分成多个简单表格。尽量避免表格跨页，如不得已，则应在下一页，表的上方标明“续前表：”（该行按正文格式，段后0.5行）。补充表头和边线。
* 几句话能表达清楚的事项，就不必用表格。

文章内容含有多种字符，要既能保证文章中所有字符都能存储又要保证能够读取到某一特殊字符就结束文章存储。

### 4.2.2 解决过程或方法

以ctrl+e(^E)打出来的特殊字符作为结束标志，通过判断是否读到这个字符来决定是否终止输入。

## 4.3 心得与体会

经过这几天的数据结构课程设计，我收获很大，在实现系统功能的过程中，也遇到很多问题，及时的向周围的同学请教，寻求帮助，设计修改，编写调试，每一步都渗透着沟通和分析，每一次沟通之后，都会对系统功能有更深层次的理解和阐述，这也促使我一步步完成了整个系统，并使之可以运行。时间太短暂了，我没有办法把全部学到的知识运用到具体的系统实现中，深深感受到数据结构这门课程的重要性，以及它在实际应用中的广泛性。这个文章编辑系统由于种种原因仍然存在需要改进，需要完善的地方，我对于这个系统的认识和对数据结构这门课程的理解已经更加深入了。在实现系统的过程中，我感受到自己的知识的薄弱零散，经过这次做系统，我觉得自己的知识得到巩固和整合，知识体系更加系统化。我也深深体会到实践检验真理，只有实践中，才能出真知。我懂得了学习不只是纸上谈兵，更重要的是实践，就比如平常我们在学习中经常会遇到一些问题，看似无关紧要跳过去，不求甚解，在得知答案的时候，总是感觉心领神会，久而久之，对于的知识点就模棱两可，形成知识漏洞。但这次课程设计所要编写的程序很长，涉及的知识点也很多，选择此题目的时候我心里面还底气十足，自认为没什么问题，对于程序的步骤仿佛了然于胸，但在实际操作中才发现并非如此，在编写程序的过程中，我不断地上网查阅相关资料进行学习，积极地思考该怎样实现系统的每一个功能。

通过这次数据结构的课程设计，我巩固和加深对数据结构基本知识的理解，锻炼了综合运用课本知识的能力，自己也真正地完整而系统地设计并编写了一个程序，使得自己的编程能力大为提高。通过对我所选择文章编辑系统的设计和编写，我初步掌握简单软件的分析方法和设计方法，第一次体会到了数据结构编程知识在实际生活中的重要性和应用的广泛性，最后，我要感谢老师和同学们，正是因为他们的热心帮助，我才能正确的完成这次课程设计。在今后的学习和工作中，会更加深入地把知识和思想体会更广泛地应用到其他学科的学习中。