中 北 大 学

课程设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| **学 院、系：** | 软件学院 | | |
| **专 业：** | 软件工程 | | |
| **班 级：** |  | | |
| **学 生 姓 名：** |  | **学 号：** |  |
| **设 计 题 目：** | 文本编辑 | | |
|  |  | | |
| **起 迄 日 期:** | 2020年6月14日**~**2020年6月18日 | | |
| **指 导 教 师:** | 杨顺民 | | |

日期: 2020年6月28日

**1 设计目的**

《数据结构》课程主要介绍最常用的数据结构，阐明各种数据结构内在的逻辑关系，讨论其在计算机中的存储表示，以及在其上进行各种运算时的实现算法，并对算法的效率进行简单的分析和讨论。进行数据结构课程设计要达到以下目的：

* 了解并掌握数据结构与算法的设计方法，具备初步的独立分析和设计能力；
* 初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；
* 提高综合运用所学的理论知识和方法独立分析和解决问题的能力；

训练用系统的观点和软件开发一般规范进行软件开发，培养软件工作者所应具备的科学的工作方法和作风。

**2 任务概述**

设计内容：

1. 输入一页文字，每行最多不超过80个字符，共N行(N>=5)；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。
2. 分别统计出其中英文字母数和空格数及整篇文章总字数；
3. 统计某一字符串在文章中出现的次数，并输出该次数；
4. 删除某一子串，并将后面的字符前移, 输出删除某一字符串后的文章；

（5） 输出形式：

a）分行输出用户输入的各行字符；

b）分4行输出"全部字母数"、"数字个数"、"空格个数"、"文章总字数"等信息。

设计要求：

（1）符合课题要求，实现相应功能；

（2）要求界面友好美观，操作方便易行；

（3）注意程序的实用性、安全性；

**3本程序采用的数据结构及模块划分**

1）本程序采用的是线性表，运用链式存储，定义了一种结构体node

typedef struct node

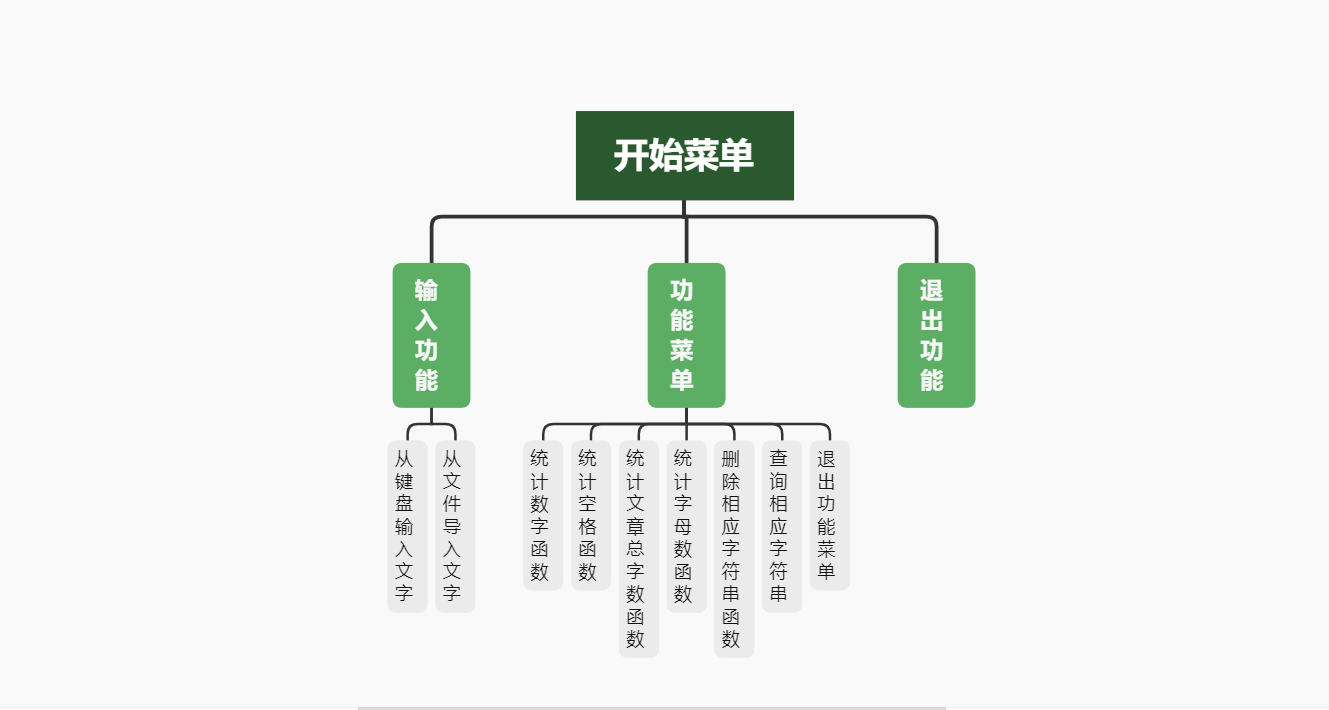
{

DataType data[MAXSIZE];

struct node \*next;

}LNode,\*LinkList;

2）模块图如图3.1所示：



**图3.1 模块划分图**

**4 主要函数说明及功能图**

1、本系统定义了十一个函数

1) 初始化函数：void InitLinkedList(LinkList &L)

2）键盘导入函数：LinkList InputKeyboard()

3）文件导入函数：LinkList InputText()

4）所有文本输出函数：LinkList OutPut(LinkList head)

5）所有字母个数输出函数：Elemtype Out\_Alpha(LinkList head)

6）所有数字个数输出函数：Elemtype Out\_NUm(LinkList head)

7）所有空格个数输出函数：Elemtype Out\_Space(LinkList head)

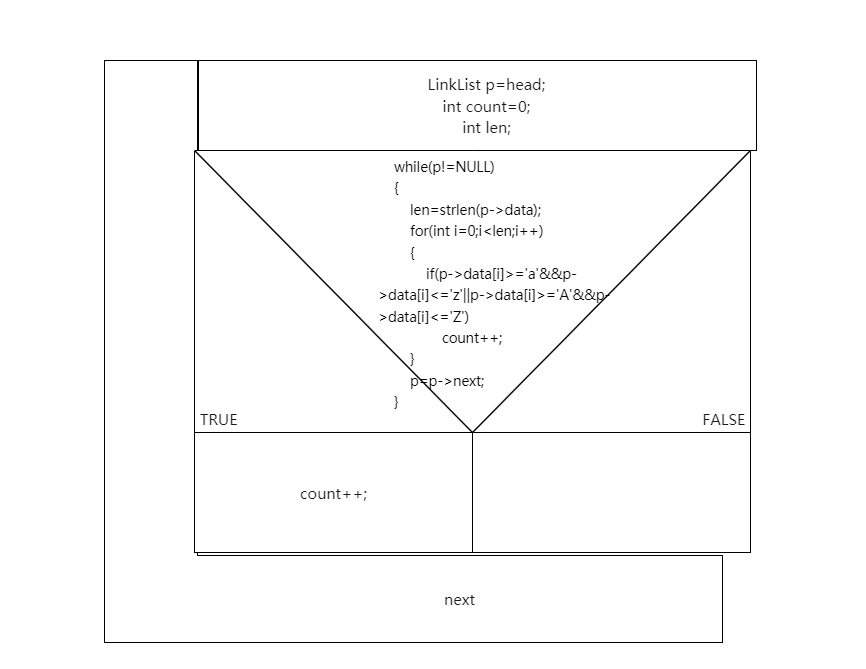
8）所有文本字数输出函数：Elemtype Out\_Text(LinkList head)

9）删除字符串的两个函数：LinkList Del\_Strings(LinkList head,char \*str)，Elemtype Del\_Words(char \*s,char \*str)

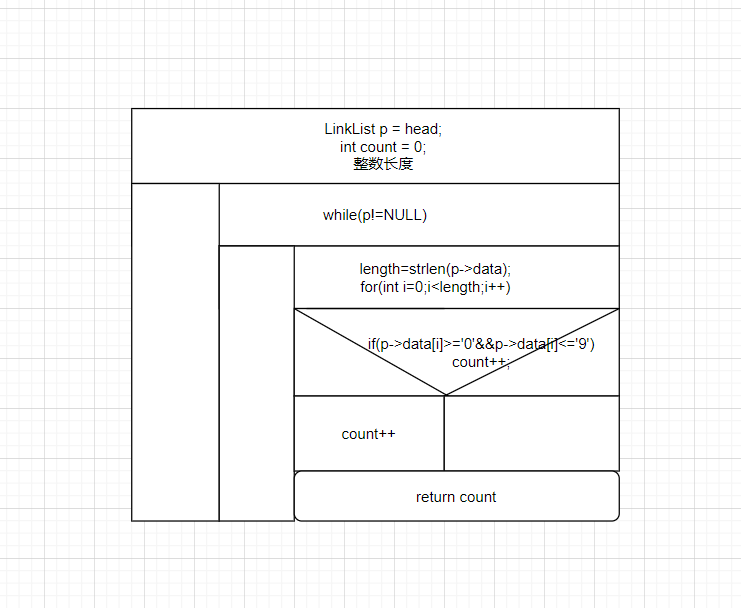
10）查询字符串的两个函数：Elemtype Search\_Strings(LinkList &head,char \*str)。，Elemtype Search\_Tools(char \*s,char \*str)

1. 主函数：main()

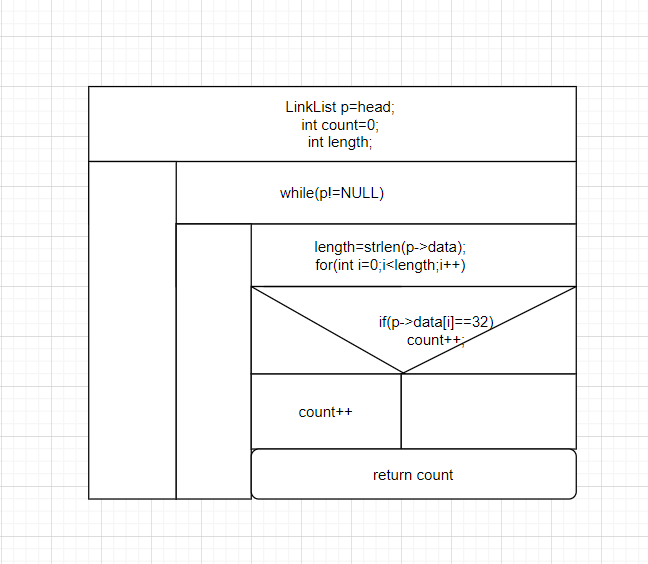
2、主要函数N-S图：



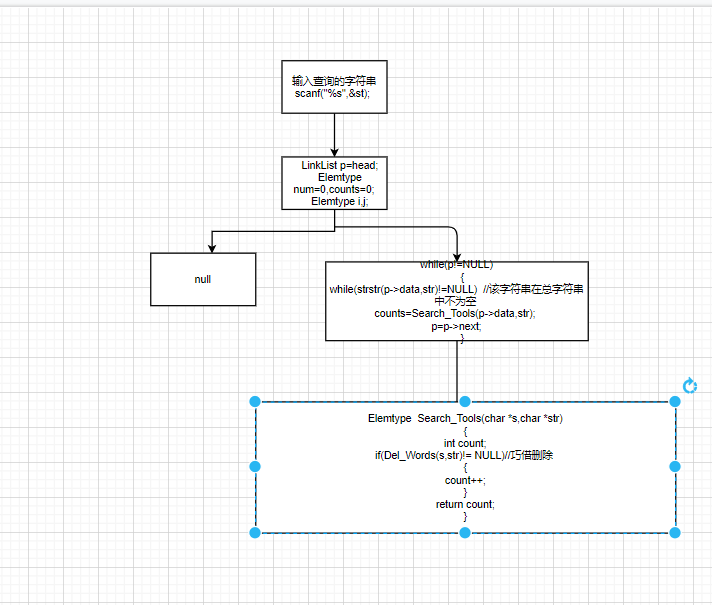
**图4.1 统计字母个数函数N-S图**



**图4.2统计数字个数函数N-S图**



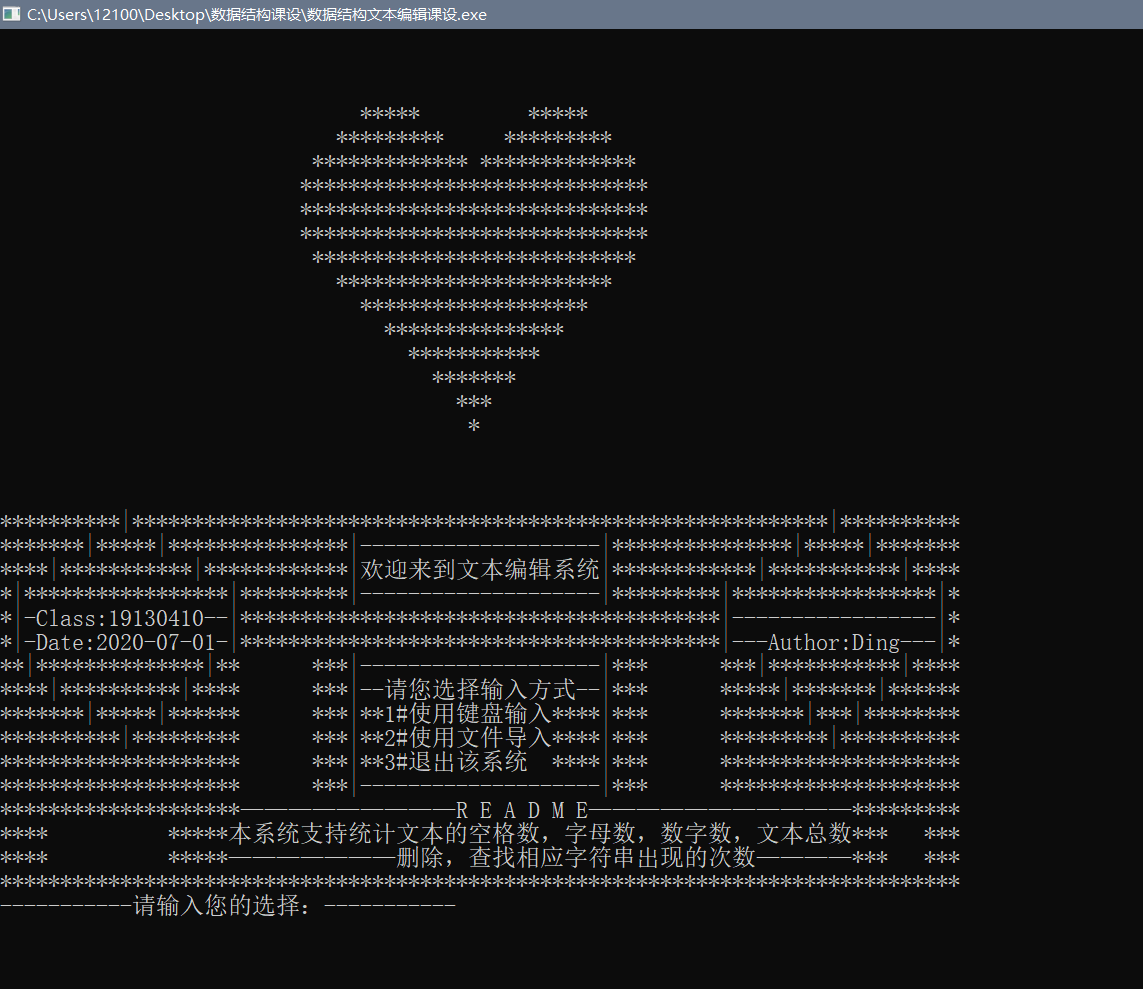
**图4.3统计空格个数函数N-S图**



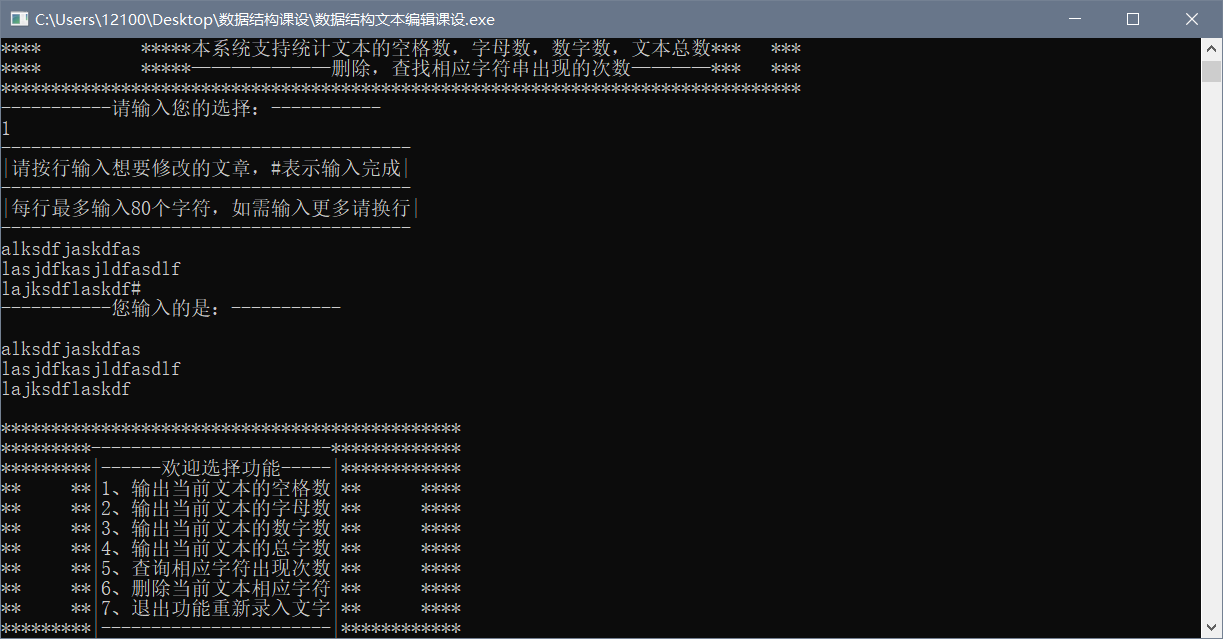
**图4.4查找字符串函数流程图**

**5运行程序及结果**

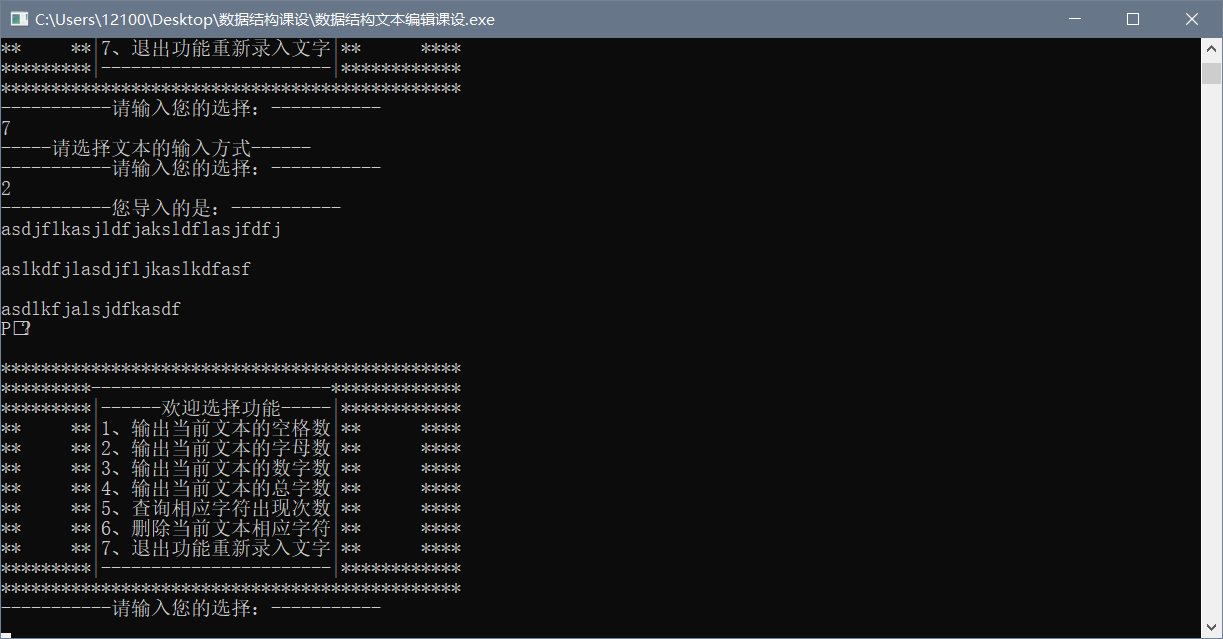
（1）开始菜单界面：



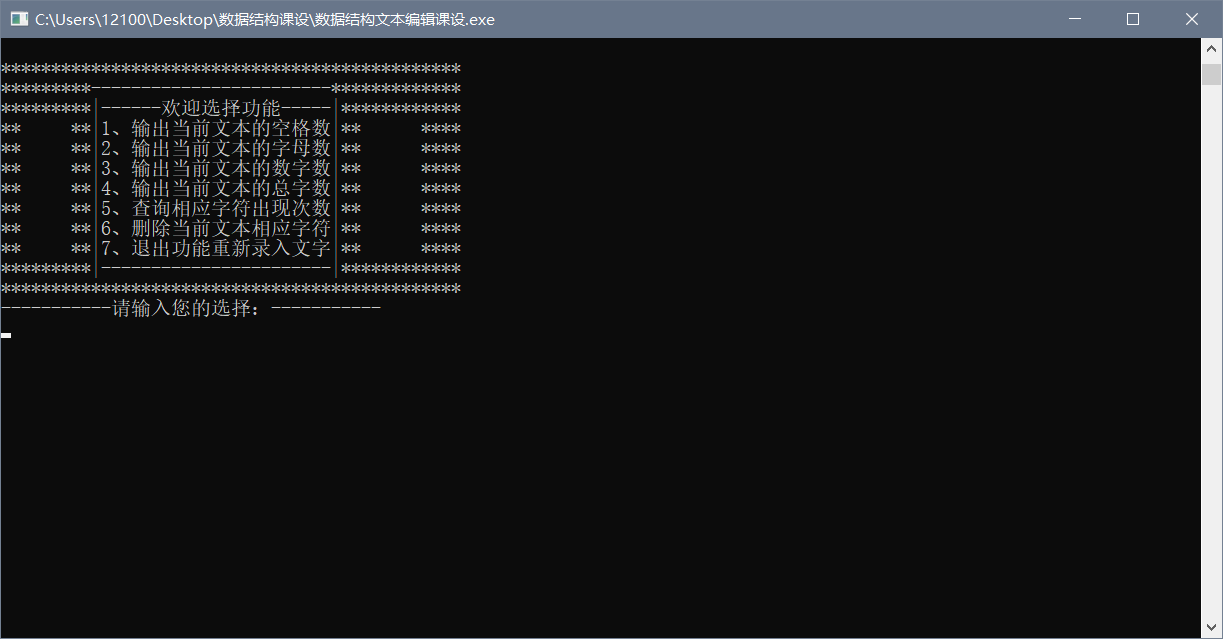
1. 键盘输入操作界面：



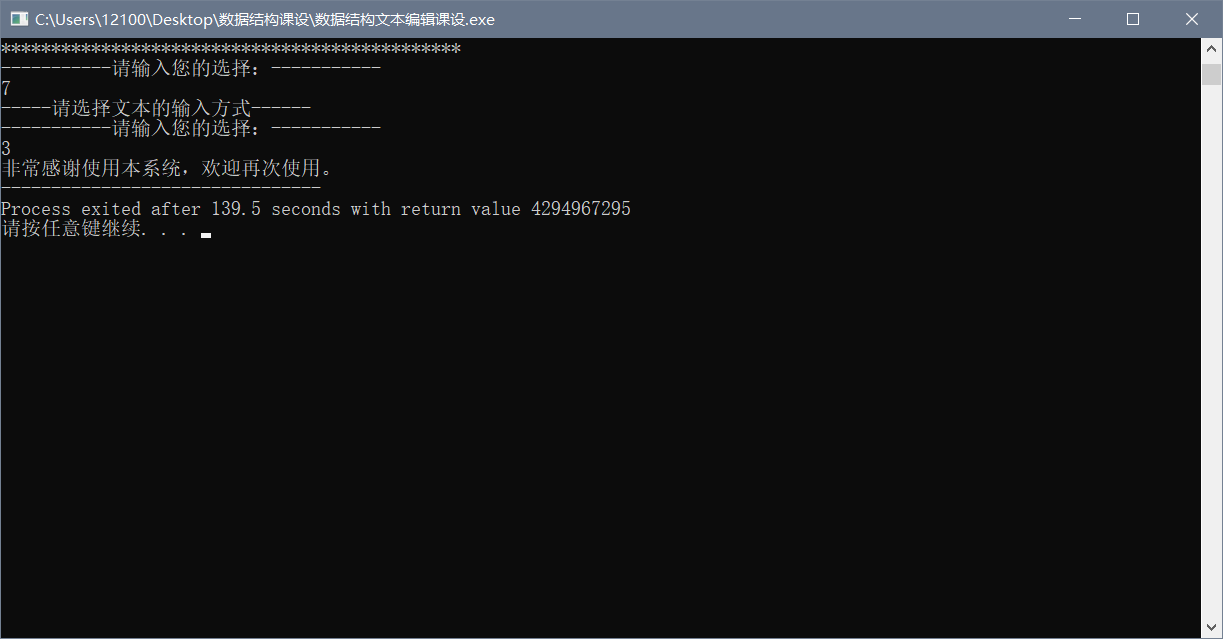
1. 文件输入操作界面



1. 功能界面



1. 结束界面



**6 课程设计心得：**

这次数据结构的课程设计不仅让我提前感受了个人开发项目的感觉，更是加强了我自己独立思，独立解决问题的能力，连续几天的挑灯夜战，让我感到了程序猿码农的辛苦，当第一遍敲完代码的时候，看着编译框中无数的bug，好像感觉自己就是来写bug的一样，baidu，google，必应，CSDN，github挨个挨个查，解决bug，问大佬，问老师，问同学，好像把他当成自己的孩子一样，毕竟第一个课设，用心程度肯定不一样，虽然我代码能力不强，图形化程序最终还是被否决了，也是一个遗憾吧，在系统界面藏了很多小惊喜，也算是自娱自乐吧，毕竟每一个好的工具可都是有一个奇奇怪怪的初始界面呢，不仅仅c语言有数据结构，其他语言也有，那么当学到其他语言的时候，那时候我的课设又会是什么样子的呢。

**附录：**

/\*

Name:文本编辑

Time：2020-6-28

Author:Ding

Function:

（1） 输入一页文字，每行最多不超过80个字符，共N行(N>=5)；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。

（2） 分别统计出其中英文字母数和空格数及整篇文章总字数；

（3） 统计某一字符串在文章中出现的次数，并输出该次数；

（4） 删除某一子串，并将后面的字符前移, 输出删除某一字符串后的文章；

（5）输出形式：

a）分行输出用户输入的各行字符；

b）分4行输出"全部字母数"、"数字个数"、"空格个数"、"文章总字数"等信息。

Else：

1-安全性

2-实用性

3-界面友好

\*/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<malloc.h>

#include<windows.h>

#define MAXSIZE 1000

#define SIZE 100

#define OVERFLOW -1

#define OK 1

#define ZERO 0

typedef char DataType;

typedef int Elemtype;

typedef struct node

{

DataType data[MAXSIZE];

struct node \*next;

}LNode,\*LinkList;

void InitLinkedList(LinkList &L)//初始化

{

//定义一个头结点

L = (LinkList)malloc(sizeof(LNode));//头结点申请地址

if (L == NULL)

printf("分配空间失败，请检查内存");

L->next = NULL;

}

LinkList InputKeyboard()//键盘导入编辑

{

LinkList p,head;

char str[SIZE];

int i=0;

printf("-----------------------------------------\n");

printf("|请按行输入想要修改的文章，#表示输入完成|\n");

printf("-----------------------------------------\n");

printf("|每行最多输入80个字符，如需输入更多请换行|\n");

printf("-----------------------------------------\n");

InitLinkedList(p);

head=p;

p->next=NULL;

while(1)

{

gets(str);

if(strlen(str)>80)

{

printf("\n");

printf("每行最多输入80个字符，请重新输入");break;

}

if(str[0]==35)

{

str[0]='\0';

p->next=NULL;

break;

}

p->next=(LinkList)malloc(sizeof(LNode));

strcpy(p->data,str);

if(str[strlen(str)-1]==35)

{

p->data[strlen(str)-1]='\0';

break;

}

p=p->next;

}

p->next=NULL;

return head;

}

LinkList InputText()//文本导入编辑

{

LinkList p,head;

FILE \*fp;

char str[SIZE][SIZE];

if(!(fp=fopen("test.txt","r")))//打开文件

{

printf("文件打开失败，请检查文件是否存在。");

exit(OVERFLOW);

}

InitLinkedList(p);//分配首地址初始化

head=p;

p->next=NULL;//尾插法

int i=0;

for(;fgets(str[i],SIZE,fp)!=NULL;i++)//读取每一行的内容

{

if(strlen(str[i])>80)

{

printf("\n");

printf("当前第%d行超过了80个字符",i);

break;

}

strcpy(p->data,str[i]);

p->next=(LinkList)malloc(sizeof(LNode));

p=p->next;

}

free(p);//释放p指针，保证数据安全

return head;

}

LinkList OutPut(LinkList head)//文本输出

{

LinkList p=head;//p指向开始节点

while(p!=NULL)

{

printf("%s\n",p->data);//输出p的值

p=p->next;

}

printf("\n");

return p;

}

Elemtype Out\_Alpha(LinkList head)//字母输出

{

LinkList p=head;

int count=0;

int len;

while(p!=NULL)

{

len=strlen(p->data);

for(int i=0;i<len;i++)

{

if(p->data[i]>='a'&&p->data[i]<='z'||p->data[i]>='A'&&p->data[i]<='Z')

count++;

}

p=p->next;

}

return count;

}

Elemtype Out\_NUm(LinkList head)//数字输出

{

LinkList p=head;

int count=0;

int length;

while(p!=NULL)

{

length=strlen(p->data);

for(int i=0;i<length;i++)

{

if(p->data[i]>='0'&&p->data[i]<='9')

count++;

}

p=p->next;

}

return count;

}

Elemtype Out\_Space(LinkList head)//空格输出

{

LinkList p=head;

int count=0;

int length;

while (p!=NULL)

{

length=strlen(p->data);

for(int i=0;i<length;i++)

{

if(p->data[i]==32)

count++;

}

p=p->next;

}

return count;

}

Elemtype Out\_Text(LinkList head)//所有文本输出

{

LinkList p=head;

int count=0;

while (p!=NULL)

{

count+=strlen(p->data);

p=p->next;

}

return count;

}

Elemtype Del\_Words(char \*s,char \*str)//删除的详细步骤

{

char \*p;

int count=0,length,i,j;

char a[SIZE];

if(s!=NULL)

{

p=strstr(s,str); // 找到相同字符串

length=strlen(s);

i=length-strlen(p);

j=i+strlen(str);

for(int n=j;n<length;n++)

{

a[count++]=s[n];//如果存在字符串则置为空

a[count]='\0';

strcpy(p,a); //将剩余字符串复制

}

return OK;

}

else

{

printf("文章中没有该字符串，请重新输入。");

return ZERO;

}

}

LinkList Del\_Strings(LinkList head,char \*str)//删除字符串

{

LinkList p=head;

while(p!=NULL)

{

while(strstr(p->data,str)!=NULL) //如果没有该字符串则循环到下一个

Del\_Words(p->data,str);

p=p->next;

}

return head;

}

Elemtype Search\_Tools(char \*s,char \*str)

{

int count;

if(Del\_Words(s,str)!= NULL)//巧借删除

{

count++;

}

return count;

}

Elemtype Search\_Strings(LinkList &head,char \*str)//查询字符串

{

LinkList p=head;

Elemtype num=0,counts=0;

Elemtype i,j;

if(p==NULL)

{

printf("文本有误。");

return ZERO;

}

else

{

while(p!=NULL)

{

while(strstr(p->data,str)!=NULL) //该字符串在总字符串中不为空

counts=Search\_Tools(p->data,str);

p=p->next;

}

return counts;

}

}

main()

{

int i, j, k, l, m;

char c = '\*';

for (i=1; i<=3; i++)

printf("\n");

for (i=1; i<=3; i++)

{

for (j=1; j<=32-2\*i; j++)

printf(" ");

for (k=1; k<=4\*i+1; k++)

printf("%c", c);

for (l=1; l<=13-4\*i; l++)

printf(" ");

for (m=1; m<=4\*i+1; m++)

printf("%c", c);

printf("\n");

}

Sleep(1000);

for (i=1; i<=3; i++)

{

for (j=1; j<=24+1; j++)

printf(" ");

for (k=1; k<=29; k++)

printf("%c", c);

printf("\n");

}

Sleep(1000);

for (i=7; i>=1; i--)

{

for (j=1; j<=40-2\*i; j++)

printf(" ");

for (k=1; k<=4\*i-1; k++)

printf("%c", c);

printf("\n");

}

Sleep(1000);

for (i=1; i<=39; i++)

printf(" ");

printf("%c\n", c);

Sleep(1000);

for (i=1; i<=3; i++)

printf("\n");

printf("\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*|\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|--------------------|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*|\*\*\*\*\*\*\*\n");

printf("\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*\*|欢迎来到文本编辑系统|\*\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\n");

printf("\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*|--------------------|\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*\n");

printf("\*|-Class:19130410--|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|-----------------|\*\n");

char st[SIZE],ss[SIZE];

int x,y;

LinkList head;

printf("\*|-Date:2020-07-01-|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|---Author:Ding---|\*\n");

printf("\*\*|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\*\* \*\*\*|--------------------|\*\*\* \*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\n");

printf("\*\*\*\*|\*\*\*\*\*\*\*\*\*\*|\*\*\*\* \*\*\*|--请您选择输入方式--|\*\*\* \*\*\*\*\*|\*\*\*\*\*\*\*|\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*|\*\*\*\*\*|\*\*\*\*\*\* \*\*\*|\*\*1#使用键盘输入\*\*\*\*|\*\*\* \*\*\*\*\*\*\*|\*\*\*|\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\* \*\*\*|\*\*2#使用文件导入\*\*\*\*|\*\*\* \*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*|\*\*3#退出该系统 \*\*\*\*|\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*|--------------------|\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*—————————R E A D M E———————————\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\* \*\*\*\*\*本系统支持统计文本的空格数，字母数，数字数，文本总数\*\*\* \*\*\*\n");

printf("\*\*\*\* \*\*\*\*\*———————删除，查找相应字符串出现的次数————\*\*\* \*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

while(1)

{

printf("-----------请输入您的选择：-----------\n");

scanf("%d",&x);

switch(x)

{

case 1:

head = InputKeyboard();

printf("-----------您输入的是：-----------\n");

OutPut(head);

break;

case 2:

head = InputText();

printf("-----------您导入的是：-----------\n");

OutPut(head);

break;

case 3:

printf("非常感谢使用本系统，欢迎再次使用。");

exit(-1);

break;

default:printf("-----您的输入不合法,请重新输入!------\n"); continue;

}

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*|------欢迎选择功能-----|\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\* \*\*|1、输出当前文本的空格数|\*\* \*\*\*\*\n");

printf("\*\* \*\*|2、输出当前文本的字母数|\*\* \*\*\*\*\n");

printf("\*\* \*\*|3、输出当前文本的数字数|\*\* \*\*\*\*\n");

printf("\*\* \*\*|4、输出当前文本的总字数|\*\* \*\*\*\*\n");

printf("\*\* \*\*|5、查询相应字符出现次数|\*\* \*\*\*\*\n");

printf("\*\* \*\*|6、删除当前文本相应字符|\*\* \*\*\*\*\n");

printf("\*\* \*\*|7、退出功能重新录入文字|\*\* \*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*|-----------------------|\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

while(1)

{

printf("-----------请输入您的选择：-----------\n");

scanf("%d",&y);

switch(y)

{

case 1:

printf("本文的全部空格数为: %d \n",Out\_Space(head));

break;

case 2:

printf("本文的全部字母数为: %d \n",Out\_Alpha(head));

break;

case 3:

printf("本文的全部数字数为: %d \n",Out\_NUm(head));

break;

case 4:

printf("本文的全部总字数为: %d \n",Out\_Text(head));

break;

case 5:

printf("请输入统计的字符串:\n");

scanf("%s",&st);

printf("%s在文本中出现的次数为: %d \n",st,Search\_Strings(head,st));

printf("\n");

break;

case 6:

printf("请输入删除的字符串：\n");

scanf("%s",&ss);

head = Del\_Strings(head,ss);

printf("删除%s后的文本为:\n",ss);

OutPut(head);

break;

case 7:

printf("-----请选择文本的输入方式------\n");

break;

default:printf("-----您的输入不合法,请重新输入!------\n"); continue;

}

break;

}

}

}