
合肥学院

计算机科学与技术系

课程设计报告

2009 ~ 2010 学年第二学期

课 程	数据结构与算法
课程 设计 名称	简单文本编辑器
学 生 姓 名	何云龙
学 号	0804012022
专 业 班 级	08 计科 (2)班
指 导 教 师	王昆仑、张冠红

2010 年 6 月

实验题目：简单的文本编辑器

1、 题目及要求

名称：简单的文本编辑器

内容：输入一页文字，程序可以统计出文字、数字、空格的个数。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）删除某一字符或者子串，并将后面的字符前移。

（4）插入某一字符或者子串。

（5）查找某一字符或者子串。

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。

2、对题目的大概理解：

本程序应实现以下功能：

（1） 文章内容的输入：包括字母、标点符号、数字、空格等；

（2） 文章内容的统计：包括文章中大写字母、小写字母、数字、标点符号、空格以及文章所有字数的个数的统计；

（3） 文章内容的处理：包括对文章内容的查找、删除以及对指定位置进行插入操作，其中在查找的过程中统计出该字符或字符串在文章中出现的次数；

1、 问题分析

本程序是对一段英文文章的内容进行处理，存储方式采用链式存储，没有文件操作，故本程序对其文本内容的所有操作都是在链表中进行的。对于文本的输入，采用头插法将文本信息存储到链表已申请好的存储空间中，在此部分设计中最大的问题在于输入文章过程中输入的字符数大于 80 时如何换行；对于文本内容的统计，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、数字数直接通过比较即可得到，标点符号通过 ASCII 比较即可得到；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者字符串是否与文章中某部分内容相同，如果存在相同的记录相同的个数及位置并输出个数及位置。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。插入部分为通过输入的插入位置（行、列）将字符或者字符串插入到文章制定位置。

● 本人对此程序的功能的扩展与完善：

本程序为简单文本编辑器，如果加入磁盘文件的操作会使得该程序更加合理与完美。因此，在输入文本时可以分为：

- ★从键盘直接输入英文文本
- ★从磁盘文件中读取英文文本

经过文本的输入后，可以把经过处理过的文本保存到原来的或新建的磁盘文件中去！加入文件的操作后，使该程序的可用性更加满足用户的需求！

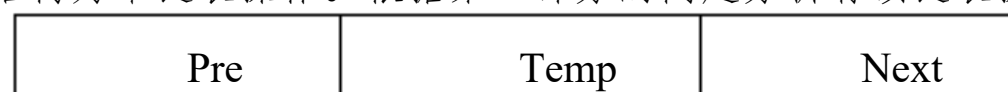
一、 数据结构选择和概要设计

数据结构选择：

本程序是对输入的文字进行操作，并要求静态存储一页文本。

由于是一页文本，包括多行，而且相邻两行的字符内容是依次读入该页的。因此，两行之间的内容要建立相应的联系。通过一个单链表，来控制该文本的行数，并额外设置一个先驱指针 `Linklist *pre` 来与上一行进行有效连接，通过 `Linklist *next` 来指向下取得与下一行的连接。为了控制每行的字符个数，在该链表的数据结构中设置一字符数组 `data[80]`，来存储每行的字符。设置 `row` 和 `length` 分别记录行数与列数；

使用的数据结构为单链表操作。根据第一部分的问题分析有该链表操作有 3 部分：



另有全局变量 `*head`，作为文章的头指针。

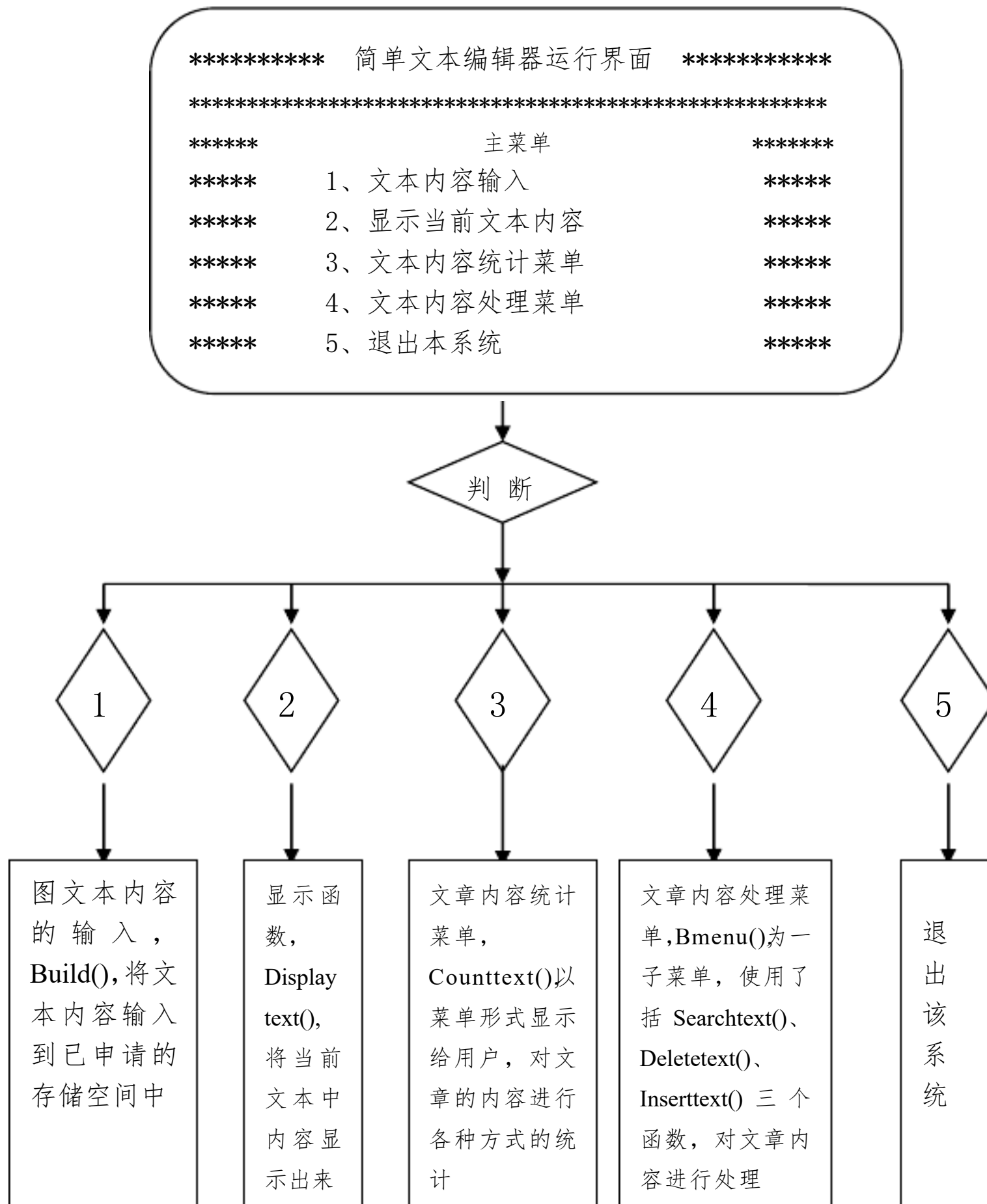
故创建了以下结构体：

```
typedef struct _node
{ char data[80];          /记录一行字符
  int length;           /记录一行字符长度
  struct _node *next;   //后继指针
  struct _node *pre;   /前趋指针
  int row;             /记录整篇文章的行数
}LinkList;
```

在文章内容创建部分中使用线性表的链式存储来实现对行的操作，实用行指针所指向的字符数组对每行应输入的字符进行存储，并使用全局变量对文本的各种信息进行存储；

文章的内容统计、删除、查找、插入都通过对行指针所指向的一维字符数组的操作来完成。

概要设计：
程序框架：



如果选择功能 1，则有：

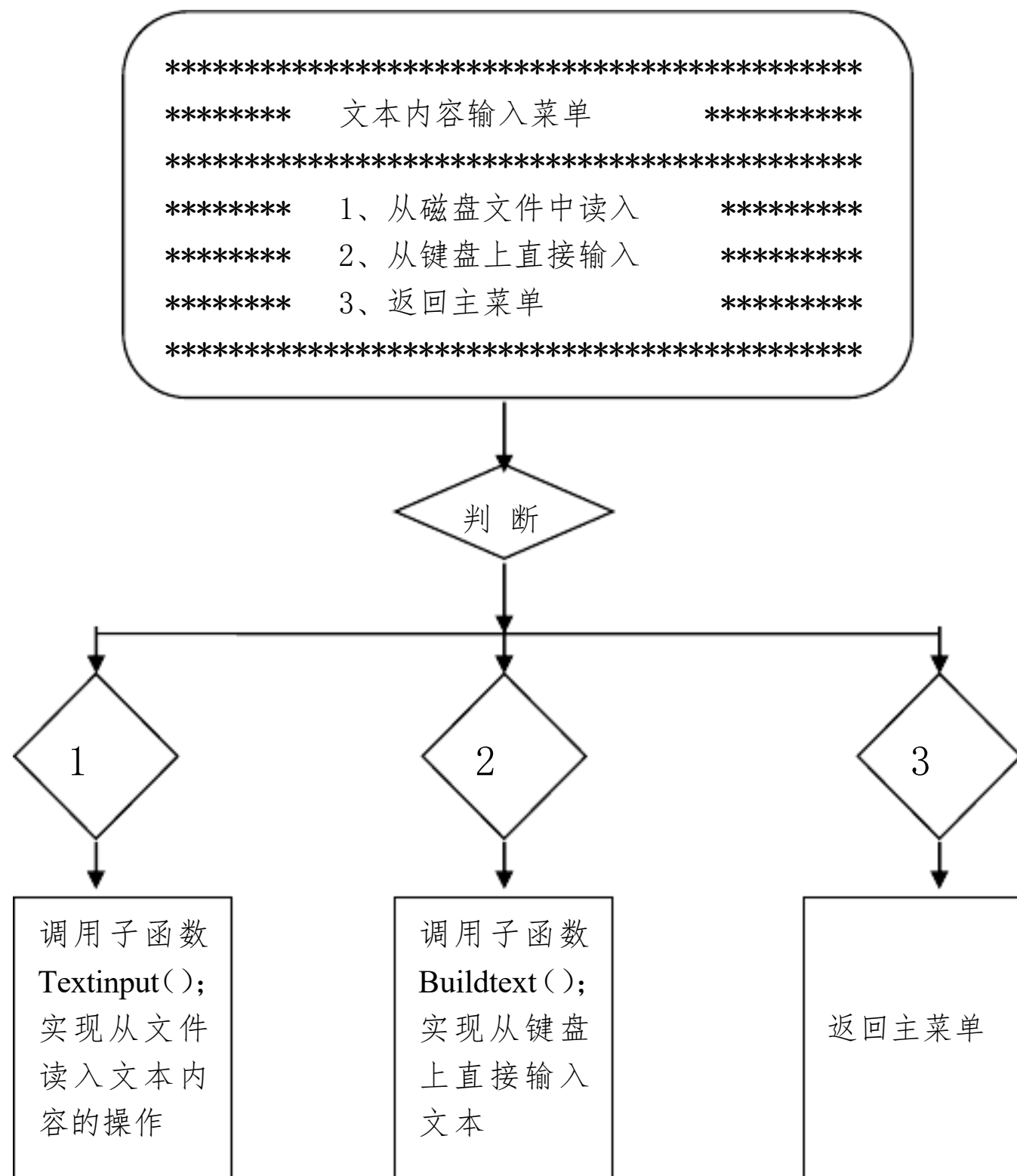


图 2 简单文本编辑器——文本内容输入菜单

如选择上图中功能 3，则有：

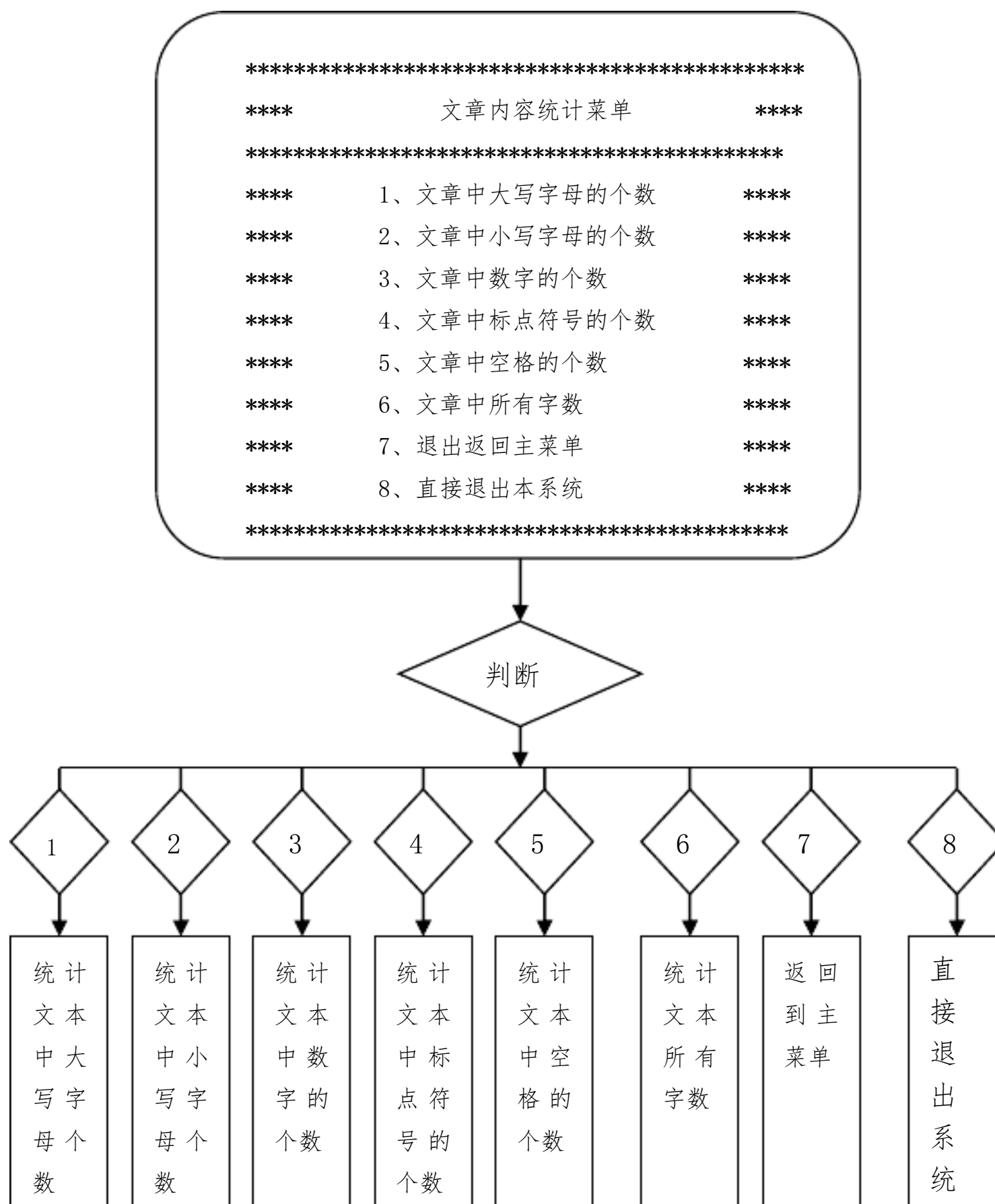


图 3 简单的文本编辑器----文本内容统计菜单

如选择功能 4，则有：

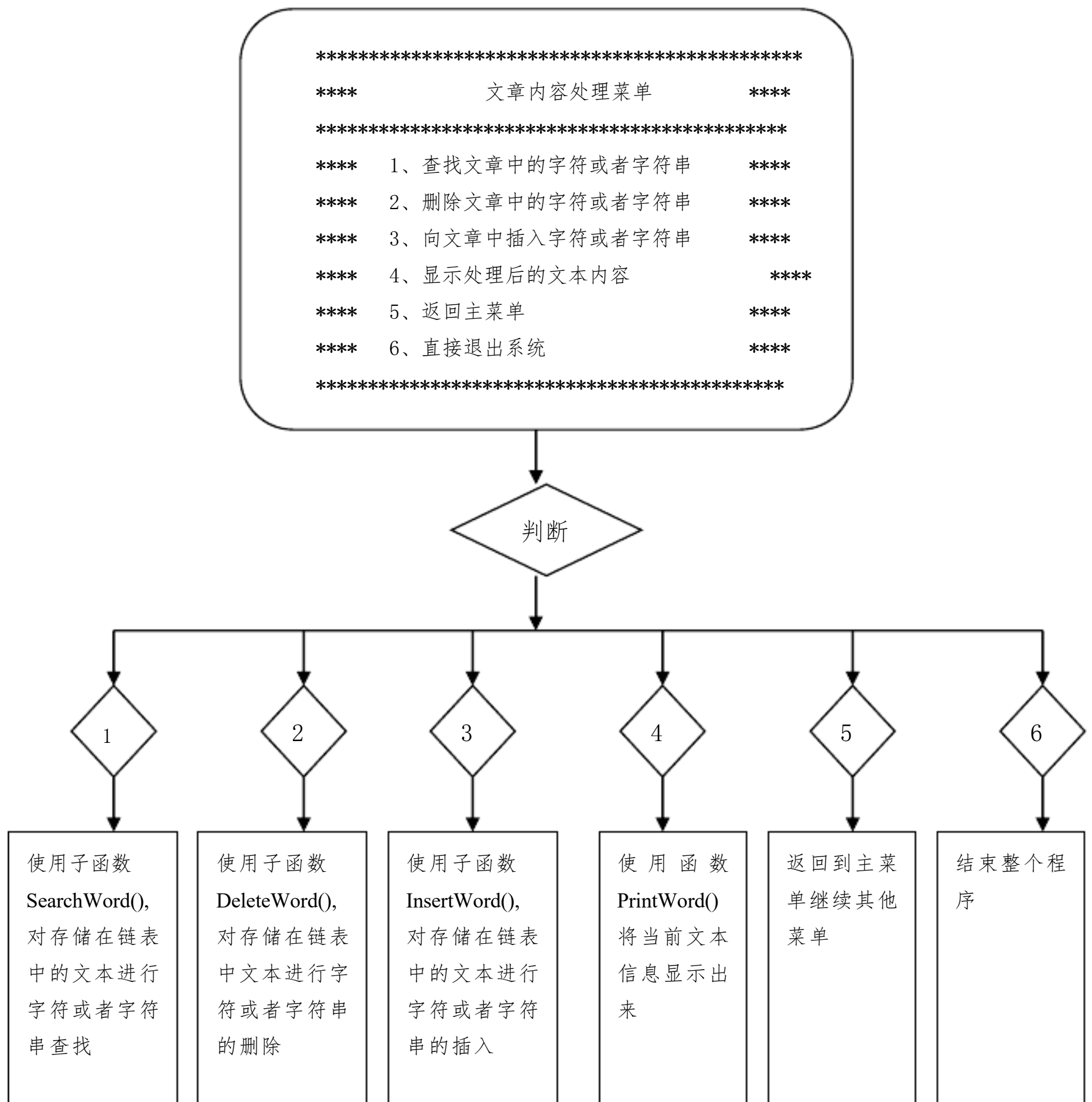


图 4 简单的文本编辑器-----文章内容处理菜单

根据以上各图，本程序共设计了 12 个函数。

- 1、Welcome() 标题函数，即一个输出标题，永远出现在程序的最顶端。
- 2、Textinput() 从指定的磁盘文件读入文本；
- 3、Buildtext() 从键盘直接输入文本内容；
- 4、Build () 文本输入菜单；实现主菜单中的功能 1，调用 Textinput () 和 Buildtext () 分两种方式输入文本
- 5、Displaytext() 当前文本内容输出函数，实现上图 1 菜单功能 2 以及图 4 中功能 4，将当前存储在链表中的文本内容输出；
- 6、Counttext() 文章内容统计函数，实现上图 1 菜单功能 3，对存储在链表中文本内容进行统计，包括对文本内容中的大写字母、小写字母、数字、标点符号、空格以及文章所有字数的个数的统计；图 2 是对其完整描述；
- 7、Searchtext() 文章内容查找函数，实现上图 4 菜单功能 1 中查找部分；
- 8、Deletetext() 文章内容删除函数，实现上图 4 菜单功能 2 中删除部分；
- 9、Inserttext() 文章内容插入函数，实现上图 4 菜单功能 3 中插入部分；
- 10、Bmenu() 第二子菜单函数，实现上图 1 菜单功能 4，图 4 是此函数实现的结果，它将 7、8、9 各子函数集合在此函数中；
- 11、menu() 主菜单函数，其结果为图 1 所显示部分，将 1、2、3、4、8、9 等函数集合。
- 12、main () 主函数

各函数关系用流程图形式绘制如下：

一、详细设计及编码

1、定义单链表结点类型

```
typedef struct _list /行表结构
{
char data[80];      /记录一行字符
int length;       /记录一行字符长度
struct _list *next; //后继指针
struct _list *pre; /前趋指针
int row;         /记录整篇文章的行数
}LinkList;
```

2、全局变量的定义

```
int NUM,C,N; //定义全局变量，Num 用来记录行号，C 用来记录子串在主串中出现的总次数
LinkList *head; //定义全局变量*head，文章首行头指针
```

3、各子函数的伪码

A) Textinput () 读文件文本内容输入函数

具体创建过程如下：

- a、定义 LinkList 指针变量*temp: LinkList *temp;
- b、定义文本输入变量 ch，记录文本行数变量 j，记录每行字符数变量 i;
- c、申请动态存储空间: head->next=(LinkList *)malloc(sizeof(LinkList));

d、首行头指针的前驱指针为空：`head->pre=NULL;`
首行指针：`temp=head->next;`
首行指针的前驱指针也为空：`temp->pre=NULL;`
定义没输入字符时文章长度为 0：`temp->length=0;`
初始化为字符串结束标志，防止出现乱码：`for(i=0;i<80;i++)`

e、从文本“abc.txt”文本文件中读取字符：

```
                                打开一个”abc.txt”文件*/
{
    exit(0);
}
利用循环进行文本输入
    for(j=0;j<LINK_INIT_SIZE;j++)// 控制一页
    { for(i=0;i<80;i++) //控制一行
      { ch=fgetc(fp);
        putchar(ch);//输出到屏幕
        temp->data[i]=ch;          temp->data[i]=ch; //给 temp 指向的行赋值
        ....
        temp->length++;//行中字符长度加 1
        if(ch==EOF)//文本结束符
        {NUM=j; break; //文章结束时，Num 来记录整个文章的行数
          }
        }
    }
```

在字符输入的过程中，如果在单行输入的字符超过了 80 个字符，则需要以下操作：

输入字符数大于 80，重新分配空间建立下一行

```
temp->next=(LinkedList *)malloc(sizeof(LinkedList));
```

给 temp 的前驱指针赋值：`temp->next->pre=temp;`

temp 指向当前行：`temp=temp->next;`

将下一行初始化为字符串结束标志，防止出现乱码：`for(i=0;i<80;i++)`

记录整个文章的行数：`temp->row=NUM+1;`

返回指向最后一行指针：`return temp;`

文本输入部分到此结束。

B) Buildtext () 键盘输入文本内容输入函数

具体创建过程如下：

a、定义 LinkedList 指针变量*temp：`LinkedList *temp;`

b、定义文本输入变量 ch，记录文本行数变量 j，记录每行字符数变量 i；

c、申请动态存储空间：`head->next=(LinkedList *)malloc(sizeof(LinkedList));`

d、首行头指针的前驱指针为空：`head->pre=NULL;`

首行指针：`temp=head->next;`

首行指针的前驱指针也为空：`temp->pre=NULL;`

```
0: temp->length=0;
```

```
初始化为字符串结束标志，防止出现乱码： for(i=0;i<80;i++)
```

e、利用循环进行文本输入

```
for(j=0;j<LINK_INIT_SIZE;j++)// 控制一页
```

```
{ for(i=0;i<80;i++) //控制一行
```

```
{ ch=getchar(); //接收输入字符
```

```
temp->data[i]=ch; //给 temp 指向的行赋值
```

```
....
```

```
temp->length++; //行中字符长度加 1
```

```
if(ch=='#')
```

```
{ NUM=j; break; //文章结束时，Num 来记录整个文章的行数
```

```
}}}
```

在字符输入的过程中，如果在单行输入的字符超过了 80 个字符，则需要以下操作：

输入字符数大于 80，重新分配空间建立下一行

```
temp->next=(LinkedList *)malloc(sizeof(LinkedList));
```

给 temp 的前驱指针赋值： temp->next->pre=temp;

temp 指向当前行： temp=temp->next;

将下一行初始化为字符串结束标志，防止出现乱码： for(i=0;i<80;i++)

记录整个文章的行数： temp->row=NUM+1;

返回指向最后一行指针： return temp;

文本输入部分到此结束。

)、Build () 文本输入菜单函数

a、建立一个文本输入菜单；

b、通过调用文本输入函数实现文本输入的功能

D)、Displaytext () 当前文本输出并写入到相应磁盘文件函数

本子函数功能为将当前存储在链表中的文本信息输出在屏幕上，具体伪码算法如下：

a、定义文本行数变量 j，每行字符数 i: int i,j;

b、定义指针变量:LinkedList *p;

c、将指针 p 指向链表表头: p=head->next;

d、输出链表中的信息，并写入到“bcd.txt 文本文件中

```
exit(0);}
```

利用循环输出链表中信息：

```
for(j=0;j<=NUM&& p!=NULL;j++)
```

```
{for(i=0;i<80)&&(p->data[i]!='#';i++)
```

写入到文件

```
}
```

```
p=p->next; }
```

文本输出函数到此结束。

、 Counttext() 文本内容统计函数

本子函数是对文本中内容进行统计。具体伪码算法如下:

- 定义指针变量 `temp:LinkList *temp;`
- 定义记录文本行数变量 `j`, 每行字符数 `i`;
- 定义记录文本大写字母数、小写字母数、空格数、数字数、标点数和总字数的变量:

```
int WORD=0,word=0,space=0,num=0,punct=0,sum=0;
```

- 将指针 `temp` 指向链表表头:`temp=head->next;`
- 利用循环对链表中信息进行匹配判断, 将大写字母数、小写字母数、空格数、数字数、标点数和总字数统计出来:

```
for(j=0;j<=NUM;j++)
{
    for(i=0;(i<80)&&(temp->data[i]!='#';i++)
    {
        ch=temp->data[i];
        if((ch>='A')&&(ch<='Z'))    WORD++;
        else if((ch>='a')&&(ch<='z'))    word++;
        else if((ch>='0')&&(ch<='9'))    num++;
        else if(ch==' ')    space++;
    else if(ch==33||ch==34||ch==39||ch==44||ch==46||ch==58||ch==59||ch==63)
        {punct++;}
    }
sum=WORD+word+num;
}
```

- 本程序对统计项目设计了菜单提供给用户选择。

菜单的编写使用 `do while` 语句进行循环操作, 此部分不再多谈。

F) Searchtext() 文本内容查找函数

本子函数是对文本内容进行查找并且对查找的内容进行统计其出现的次数。

具体伪码算法如下:

- 定义一个数组, 用来记录需要查找的字符内容:`char Data[20];`
- 定义定义文本行数变量 `j`, 每行字符数 `i`, 记录字符出现的次数变量:
`int i,j,m=0,sum=0;`
- 对形参中定义的指针变量进行操作, 使其指向链表表头:`temp=head->next;`
- 使用 VC++ 中的拷贝函数, 将形参中 `str1` 的值赋给 `Data`: `strcpy(Data,str1);`
- 利用循环进行查找操作, 核心算法为:

```
if((temp->data[j])==Data[k])
```

```
    k++;//将输入的查找字符与链表中信息比较, 找到第一个相同的字符
```

```
    {j=j-k; k=0;} //从主串第 j-k 个位置重新查找
```

```
    {sum++;//此字符出现的次数加 1
```

```
    j=j-k+1; //j 记录下该字符串出现的位置
```

```
        第%d次出现在第%d行第%d列
```

```
    l++;
```

```
    k=0;
```

```
    continue;}
```

```
temp=temp->next; //
```

对文本内容查找的部分完成。

)、Deletetext() 文本内容删除函数

此子函数是对文本内容进行删除。具体伪码算法如下：

f、定义一个数组用来存储需要删除的字符或者字符串：`char Data[20];`

g、定义指针变量：`LinkList *temp, *term;`

h、定义整形变量用来控制行数、字符数：`int i,j,k,m,y,num;`

i、使用 VC++ 中拷贝函数讲形参中需要删除的字符或字符串赋值给已定义的数组：

```
strcpy(Data,str2);
```

j、使用循环进行删除操作：

其核心算法为：

```
for(i=0;i<=NUM;i++)
{ for(j=0;j<80;j++)
{ if((temp->data[j])==Data[k]) k++;
```

```
{ num=j; break; } }
```

```
if(num<80) break;
```

```
}
```

首先是使用循环查找到需要删除字符或者字符串的字符数以及字符或者字符串的位置，以便于删除；

```
for(;i<=NUM;i++)
```

```
{ for(;j<80;j++)
```

```
{ if(j+1<k)
```

```
{ term->data[80-k+num]=temp->data[j+1]; } //删除的字符串不在最后一行，将下一行的字符（由 temp 指向）前移到前行
```

```
else temp->data[j-k+1]=temp->data[j+1]; } //当要删除的字符串在最后一行只要将最后一行的字符前移
```

```
term=temp;
```

```
temp=temp->next;
```

```
j=0; } //在使用循环，从查找到的字符或者字符串开始进行删除，在一行删除完毕之后，转至下一行进行删除。
```

删除部分到次完成。

H)、Inserttext() 文本内容插入函数

此函数为向文本中插入字符或者字符串。具体伪码如下：

a、定义需要插入的字符或者字符串的输入数组：`char Data[20];`

b、定义需要插入的字符或者字符串的插入的行和列：`int h,l;`

c、进行输入操作，将上述两个部分进行输入

d、定义一个整形变量，确定插入的具体位置：`int i=(h-1)*80+l;`

e、定义一指针变量：`LinkList *a;`

f、对输入的字符进行插入操作，具体核心算法如下：

```
int n=strlen(Data);
```

```
int m ;
```

```
int insertRow=i/80+1;//确定插入位置的行数
```

```

int row=temp->row;//          row
int j;
if(insertRow==row)//判断插入位置是否在最后一行
{for(m=temp->length-1;m>=(i%80)&& n>0;m--)
temp->data[m+n]=temp->data[m];//将最后一行插入位置后面的所有字符向后移 n 位
for(m=(i%80),j=0;m<n+(i%80);m++,j++)
{temp->data[m]=Data[j];    }/将要插入的字符或者字符串赋值到要插入的位置
}
else//如果插入的位置不再最后一行
{    int r=0;
for(int p=insertRow; p<row;p++)
{if(p == insertRow)    r=0;
else    r=n;
for(m=temp->length-1-r;m>=0&&n>0;m--)
temp->data[m+n]=temp->data[m];//将最后一行整体后移 n 位
a= temp; //p 指向前一行
temp = temp->pre;//temp 指向前一位
temp->length = 80;
for(m = temp->length-n,j=0;m<temp->length;m++,j++)
a->data[j]=temp->data[m]; }//将前一行后 n 个字符移到下一行的前 n 个字符位置
for(m=temp->length-n-1;m>=(i%80);m--)
temp->data[m+n]=temp->data[m];//插入行位置后的字符后移 n 位
for(m=(i%80),j=0;m<(i%80)+n;m++,j++)
temp->data[m] =Data[j];}//将要插入的字符赋值到插入的位置
return temp;}

```

、 Bmenu () 文本内容处理菜单

本函数为文本内容处理菜单，即使用循环语句对文本内容处理部分的子函数集中在此函数中。

J)、 Welcome () 是将一标题输出在 DOS 节面的最上端；

K)、 menu () main ()

主菜单函数以及主函数；

本人将主菜单单独创建一个子函数，方便观看。仍然是很简单的循环语句实现。

主函数为：

```

void main()
{head=(LinkedList *)malloc(sizeof(LinkedList));
LinkedList *temp;
menu(temp);
}

```

先为 head 申请存储空间，然后创建指针变量 temp，然后直接在主菜单函数中进行其他操作。

四、 上机调试

1、 上机调试过程中遇到的问题及解决方法：

a) 由于本程序的操作大部分都为经常使用的查找、删除、插入，基本的算法及代码都

函数中复杂操作过程中出现的小问题。

- b) 删除及插入过程中需要在子函数中定义一个数组变量，用来存储输入的信息，然后用来与链表中的信息进行比较以达到删除或者插入。另为达到将输入的信息赋值给子函数中定义的数组，特使用了 c++ 中的 strcpy () 函数。
- c) 为了将程序界面更加的完美，做了 3 个菜单，还写了一个子函数专门将标题显示在程序最上端。清屏函数使用 system ("cls")。

2、时间，空间性能分析：

本算法的空间复杂度很低，只需要文本的行数 $(N-1) * 80 + L$ (最后一行字符数) 的数组存放结果，因此空间复杂度为 $O(N)$ 。但是本算法的时间复杂度比较高，由于输出函数、查找函数算法时间复杂度均为 $O(N)$ ，但是删除函数、插入函数算法的时间复杂度较高。但是，我暂时还问想到时间复杂度更小的算法，因此在这里我无法对该算法进行优化。

3、经验和体会：

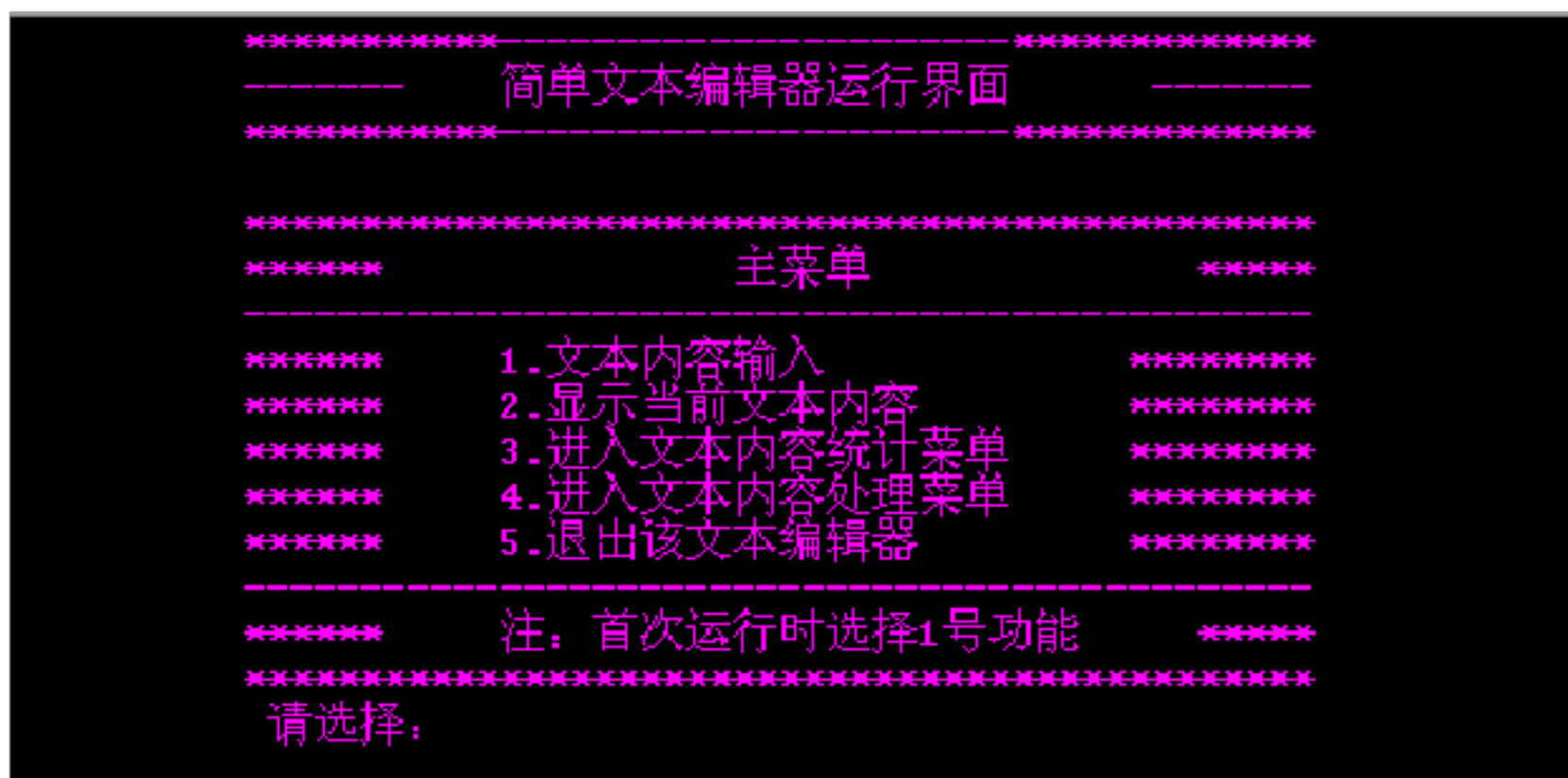
得到文本编辑器这一课程设计题目时，自己还比较害怕是制作向 windows 系统中记事本一样的程序。看了任务书之后自己有了些欣慰，任务书上要求是对输入的信息进行操作，这是我们经常练习的内容，虽然是比平时练习的更为难，更有深度，但自己毕竟知道该向什么方向去努力，自己应该如何去编写这个程序。所以自己在整个课程设计过程中还是比较轻松的，编写过程中遇到的困难及问题都通过查阅资料、向老师提问得以解决。这都是基于对课题有一个明确的了解，清楚向什么方向去写。所以本次课程设计自己最大的体会就是不管写什么程序，自己首先得对这个问题要分析透彻，要知道自己要干什么，然后才能让自己干什么。

五、测试结果及分析

在 abc.txt ”中存有以下一段文本：

How To Chat Someone Up On The Train, Bus Or Tube
Remember that person you saw on the tube that you thought was devastatingly attractive but didn't have the nerve to talk to?
With our help, you may never have to wonder if the one that got away was actually 'The One'. Follow these easy steps and you will have the confidence to chat someone up on public transport !

开始在 VC++6.0 中运行此程序：



，有如下结果：

```
*****-----*****
----- 简单文本编辑器运行界面 -----
*****-----*****
*****-----*****
*****          文本内容输入菜单          *****
*****-----*****
*****          1.从磁盘文件中读入          *****
*****          2.从键盘上直接输入          *****
*****          3.返回主菜单                *****
*****-----*****
请在菜单中选择需要的操作：_
```

在这里我们选择功能 1，从“abc.txt”中读取文本，并输出到屏幕：

```
*****-----*****
----- 简单文本编辑器运行界面 -----
*****-----*****
从磁盘文件中读出的文本：
How To Chat Someone Up On The Train, Bus Or TubeRemember that person you saw on
the tube that you thought was devastatingly attractive but didn't have the nerve
to talk to? With our help, you may never have to wonder if the one that got awa
y was actually 'The One'. Follow these easy steps and you will have the confiden
ce to chat someone up on public transport # 按回车键继续....._
```

经与“abc.txt”文本中的内容对照，读取磁盘文件操作时成功的，并把文本中的信息存
储到申请的由行指针指向的一维数组中！

选择 3，返回主菜单后，选择操作 2，显示当前文本内容：

```
*****-----*****
----- 简单文本编辑器运行界面 -----
*****-----*****
当前文本的内容是：
How To Chat Someone Up On The Train, Bus Or TubeRemember that person you saw on
the tube that you thought was devastatingly attractive but didn't have the nerve
to talk to? With our help, you may never have to wonder if the one that got awa
y was actually 'The One'. Follow these easy steps and you will have the confiden
ce to chat someone up on public transport
按回车键继续....._
```

再次证明文本中的内容存到所申请的空间中！

返回到主菜单后，选择功能 3，进入文本内容统计菜单：

```
*****-----*****
----- 简单文本编辑器运行界面 -----
*****-----*****
*****          文本内容统计菜单          *****
*****-----*****
*****          1.文本中大写字母个数          *****
*****          2.文本中小写字母个数          *****
*****          3.文本中数字的个数            *****
*****          4.文本中标点符号的个数        *****
*****          5.文本中空格的个数            *****
*****          6.文本中总的字数              *****
*****          7.退出返回主菜单              *****
*****          8.直接退出本系统              *****
*****-----*****
请在主菜单中选择需要的操作：_
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/178112002011006133>