# 十三章：文件输入/输出

## 例题：

### 13-2.c

**任务：**程序通过命令行接受参数，参数为待压缩的文件名

\* 程序把文件压缩成原来的三分之一（写入每三个字符的第一个字符）,压缩后的内容保存在文件名.red中

**知识点：**\*\*\*这个程序演示了如何在程序中创建一个文件，给它命名，**添加后缀名**，并且向文件中写入内容\*\*\*

### 13\_3.c

任务：接受用户输入的单词（也就是没有分隔符的连续字符串数组），并将单词依次写入文件

### 13-5.c

任务：程序打开若干个源文件和一个目标文件，把源文件的内容追加到源文件末尾

\*

\* 知识点：1： 利用fwrite()和fread()进行拷贝（保证了拷贝的精度）

\* 2: setvbuf()函数建立缓冲区

\* 3： ferror()处理错误

\* 4: strchr()查找字符(\*\*\*利用strchr定位给字符串结尾加上\0, 是给字符串结尾的新方法\*\*\*\*)

\* 5. rewind()回到文件开头

\* 6. 其它各种输入输出函数演示

### 13-6.c

**任务**：程序演示二进制件的随机访问（fseek定位，fread/fwrite读写）

\*

\* 知识点：1.（fseek定位，fread/fwrite读写）

\*

\*

\* **算法：**

\* 二进制写模式打开文件并检测

\* fwrite把数组中的内容写入文件

\* 关闭文件

\*

\* 二进制读模式打开文件并检测

\* 用户输入要访问的位置i

\* 根据i计算fseek的偏移量pos

\* 根据pos用fseek定位

\* fread()读取该位置的值，输出

\* 关闭文件

## 复习题：

### E4.c

复习题4：

\*  **任务**：计算输入的浮点数的算数平均数

\* 如果没有命令行参数，就从终端标准输入

\* 如果有命令行参数，那么这个参数解释为文件名，从这个文件中获得输入

\* 参考答案是用分支语句来选择fp指针，显然更加高效！！！

### W\_E5.c

/\*复习题5：

\*

\* 任务：编写一个程序接受两个命令行参数，第一个参数是字符，第二个参数是文件名，程序只打印文件中包含指定字符的行

\*

\* **算法：**

\* 打开并检测文件

\* while(不是EOF ,读入文件的一行，存进缓存（数组))

\* 用strchr判断行是否包含指定的字符，如果存在就打印

**知识点**：1.strchr 判断是否包含某个字符 2. feof判断是否到文件结尾

## 习题

### W1.c

/\*程序修改例题13.1（统计文件字符数量），要求不使用命令行模式，获取用户输入的文件名

\*

\*  **知识点：**

\* 1. fgets()会把结尾的'\n'都读入并且储存！所以需要下面一点程序来处理'\n'或者是多余的字符

if(find=strchr(filename,'\n'))

\*find='\0';

else

while(getchar()!='\n')continue;\

### W2.c

**任务**：拷贝文件，文件副本的名称为源文件.cp。用命令行获得原始文件名称

\* 要求以二进制模式读写

\*

\*  **知识点**：1.为文件添加后缀（亦见例题[13\_2.c](#_13-5.c)）

\* 2. 利用二进制读写进行拷贝

\* 3. ferror检测错误

\*

\*  **算法：**

\* 检测命令行参数

\* 产生副本文件名称

\* 以二进制读模式打开并检测源文件

\* 以二进制写模式打开并检测副本文件

### 目录WWW3.1

**任务**：用户输入一个文件名，程序读取文件内容，并把所有小写转换为大写，写回到原文件中（替代掉原来的内容）

**知识点**：修改文件中的内容

**版本1**：是个坑坑~~但是其中文件位置的反复变换仍然是值得学习的

**版本2**：优点：代码简洁，不需要借助临时文件（避开权限问题）

缺点：不及版本三那么灵活

**版本3：**优点：灵活，可以任意修改文件

缺点：需要借助临时文件，如果没有建立文件的权限，那么就GG了

#### 待解决：3\_1.c

\* **算法** ： 我认为这个程序的难点在于要把转换后的内容写回到源文件中，那么就要求一下子读取文件中的所有内容，转换为大写之

后，再存储到源文件中，这样就可以避免替换掉还没有读取的内容。但是这样的问题在于~~~缓冲区有那么大吗？

\*

\* 我的解决方案(试图不借助任何临时文件！)：

\* 用r+模式打开文件

\* 位置移动到文件的结尾，记录下结尾的位置（方便之后删除前面的内容，以及知道该读到哪里）

\* 回到文件开头

\* 读文件内容，直到刚刚前面记录的结尾位置

\* 把转换为大写后的内容附加在文件的结尾后面

\* 利用之前记录的文件位置，删除转换前的内容

\*

\*  **待解决**：如何不借助临时文件删除文件的某部分内容？

#### 3\_2.c

**程序版本2（一次读取，转换一个字符,不借助临时文件）**

\*

\* 程序版本1发现无法不借助其它临时文件而实现删除文件中一部分内容的功能

\* 因此程序版本2采取一次读取一个字符，转换为大写，然后再替换掉原来字符的方法，而不像程序版本1那样先读取完文件的所有内容再进行转换

\*

\***知识点**：1.替换文件内容

\*

**\* 算法 ：**

\* 我的解决方案(试图不借助任何临时文件！)：

\* 用r+模式打开文件

\* while(程序遇到结尾之前)

\* 读取一个字符

\* 回退1位到原来的位置

\* 写入转换后的字符，替换掉原来的字符

\* 关闭文件

#### 3\_3.c

\*程序版本3：读入整个文件内容，借助临时文件

\*

\* **知识点：**  1. remove()删除函数

\* 2. rename()重命名文件

\* 3. 借助临时文件修改文件内容(指定内容)

\*

\* **算法** ：

\* 我的解决方案(借助临时文件)：

\* 用r+模式打开文件

\* w+模式（创建）打开临时文件temp

\* 把源文件内容转换为大写并且写入源文件（从头写入，替换内容）

\* 关闭两个文件

\* 删除原文件

\* 临时文件改名为源文件

### 4.c

/\***任务**：按顺序在屏幕上显示命令行输入的文件，argc控制循环

\*

\* **算法：**

\* 检验参数

\* 循环（argc-->0）

\* 打开文件

\* 读取并输出内容

\* 关闭文件

### 5.c

\*本程序将[例题13.5](#_13-5.c_1)有用户输入界面更改为命令行界面。第一个参数为目标文件，后面的参数为源文件名

\* 最少要求输入两个参数

### 待解决：6.c

本程序将[例题13.2](#_13-2.c)由命令行模式修改为用户输入模式

**问题**：为什么说使用命令行参数的程序依赖于用户的内存如何正确地使用它们

### 目录7.c

/\***任务**：用户输入两个文件名，交替打印两个文件的每一行

\*

\* **知识点**：交替输出两个文件的行

\*

\* **算法：**

\* 提示用户输入两个文件名

\* 以读模式打开两个文件并且检测

\* while(任意一个文件没有到结尾)

\* 没到结尾，按行输出文件1

\* 没到结尾，按行输出文件2

\*

\* 关闭两个文件

#### 7\_a.c：一行行输出

#### 7\_b.c：同行号的同行输出

### 8.c

**任务**：程序通过命令行参数接受一个字符和任意个文件名，一次报告每个文件中该字符出现的次数

**算法：**

for(循环访问所有文件)

打开并检测文件

while(文件结尾之前)

按字符读取文件，如果字符是指定字符，++

打印字符出现次数

初始化字符出现次数

### 9.c

/\*本程序是对例题13.3的修改。按照单词写入的顺序为单词编号。确保下次运行程序时单词序号是之前的继续

\* 程序的关键点在于：\*\*每次运行程序的时候，程序都需要知道之前已经有多少个单词，或者说知道当前的最大序号是多少

\* 以便能够在序号的基础上继续。\*\*

\*

\* **知识点**：统计文件中的单词数量(当然也可以采用之前的方法，但是比较麻烦)

//计算之前输入了多少个单词

while(fscanf(fp,"%40s",words)==1)num++;

### 10.c

/**\*任务**：提示用户输入一个文件和一个文件位置，程序打印从文件位置到换行符之前的内容

\* 用户输入非数字支付或者负数来退出程序

\*

\* **知识点**：文件的随机访问（二进制）（同见例题13.6）

\*

**\* 算法**：

\* 提示用户输入文件名

\* 以rb模式打开并检验文件

\*

\* 提示用户输入文件位置

\* while(位置为非负整数)

\* fseek定位

\* while(fread读入字符且不是'\n')

\* fwrite输出到屏幕

### 11.c

\***任务：**编写一个程序，接受两个命令行参数，第一个参数为一个字符串，第二个参数为一个文件名

\* 打印文件中所有包含该字符串的行(这个程序是通过命令行模式来输入的，所以显然有一个缺陷，那就是输入的字符串必须是

连续的，不能有分隔符！而如果是使用用户界面进行输入就可以解决这个问题)

\*

**\* 知识点：**1. strstr查找包含的字符串

\* 2. fgets()字符串输入

\*

\* **算法：**

\* 获得命令行参数

\* 判断是否为两个参数，不是则返回错误

\*

\* 以读模式打开文件

\* 按行读取并储存（假定每行255个字符），直到末尾

\* strstr检验是否出现指定字符串

\* 有，则打印整行

\*

\* 关闭文件

### 12.c

/\***任务**：将一个20行30列的包含0~9的数字矩阵转换成灰度，方法是每个数字对应不同的符号输出~~~比如0为空格，9为#

\* 可以通过命令行获得文件名，输出文件为文件名.img

\***知识点**：1.一次读取一个Int类型数据fscanf(fp1,"%1d",&n)

2. 添加后缀名

3.略微窥见图像输出原理

**答案:** 答案的优越出在于:在选择是哪个数字对应哪个符号输出的时候,首先定义并初始化了数组: const char tran[LEVEL+1]=”!@##$%”，然后将之前读入的数字作为下标以此输出，相比我自己所用的switch语句要简洁高效得多

\* **算法**：

\* 命令行获得并且检验参数

\* 打开并且检验文件

\* 创建文件.img, w模式打开

\* 逐个读取数字，并且存入20\*30的int类型数组中

\* 转换int类型数组元素，存入20\*31的字符串数组中

\* 输出字符串数组内容到.img文件和屏幕上

\* 关闭两个文件

### 13.c

/\***任务:**在练习13.c的基础上改用变长数组VLA.

\* 我这里都在main函数中完成操作，如果要调用子函数，就需要在用参数传递变长数组之前传递行列信息

\*

\***知识点：**1.变长数组

### ？14.c

# 第十四章：结构和其他数据形式

## 例题

### 14\_1.c

**任务**：让用户输入一本书的书名，作者，价格，然后显示在屏幕上

**知识点**：声明、定义和使用结构体变量

### 14\_2.c

**任务**：创建多本书的目录，包含每本书的书名、作者、价格信息

**知识点**：结构体数组定义和访问

### 14\_3.c

/\***任务：**管理自己的朋友信息,输出一封给好友的信

\* **知识点**：1. 结构嵌套

\* 2. 数组指针的使用

### 14\_4.c

//**知识点**：1.结构体指针 2. 结构体数组的初始化（包含嵌套）

### 14\_5.c

//**知识点**：结构成员作为参数传递

### 14\_6.c

//**知识点**：结构体地址作为参数传递

### 14\_7.c

//**知识点**：整个结构体内容作为参数传递

### 14\_8.c

/**知识点**：子函数中，使用指针修改结构体

### 14\_9.c

**知识点**：子函数中，传递和返回结构体来修改结构体内容

### 14\_10.c

**任务：**在程序14\_8.c的基础上，将fname和lname改为指针（原本是数组），以演示不同

**知识点：**1. 结构体中指针的\*\*安全\*\*使用 2.动态分配内存

### 14\_11.c:复合字面量

//**知识点**：复合字面量

### 14\_12.c：伸缩数组成员

//程序计算数组的平均值

2 //**知识点：**伸缩型数组成员

### 14\_13.c

//**知识点**：结构数组的数组名=结构第一个成员地址

### W14\_14.c

/\***任务：** 建立一个书单文件：1用户可以往文件中新添书单（作者，书名，价格）2.显示书单文件内容

2 \* 注意：我对原题做了修改：

3 \* 原题是把添加的书单内容放入结构数组，然后存入文件，并且输出结构数组的内容

4 \* 而我是把结构数组存入文件后，输出文件内容（这样才知道有没有成功写入）

5 \*

6 **知识点**： 1. 从文件读取/存入结构（fread/fwrite）

7 2. 把二进制文件内容输出到屏幕上(关键在于清楚文件指示器当前指向的位置！)

8 **算法：**

9 定义结构数组（用来暂时储存书单中的内容）

10

11 打开并检测书单文件（a+b模式）

12 判断书单是否已经满了

13 是：结束程序

14 否：继续

15 fread()内容读入结构数组

16 filecount记录读了多少个结构（便于在文件后面追加新内容）

17

18 判断：书单是否已经满了？

19 是：退出

20 否：继续

21

22 whiel(获得并判断用户输入的内容，直到书单满或者空行或者输入无效)

23

24 fwrite把新添加的结构数组写入文件

25

26 文件中是否有内容？

27 有=》

28 显示文件内容：

29 \*\*文件指示器指向文件开头\*\*

30 定义临时结构体temp

31 while(fread读取内容到temp,知道文件结尾)

32 printf()输出到屏幕

33 关闭文件

34

35 无=》提示没有内容

36 \*/

### 14\_15.c

//**任务：**用户输入一种颜色，程序判断程序中是否有这种颜色

2 //**知识点**：1.枚举类型enum(可以出现在所有int类型能出现的位置，提高代码可读性)

### W14\_16.c

/\***任务**：用户输入一段字符串

根据用户选择实现下面功能：

u: 转换为大写

l:转换为小写

t: 大小写转换

o:输出原字符

n:下一个字符串

**知识点**：1. 函数指针作为函数参数（用于选择哪一个函数,关键在于根据用户选择，定义函数指针指向哪个函数）

2. 菜单小技巧：判断strchr("abc",ch)==NULL =》相当于：ch!=a && ch!=b && ch!=c

3. 将用户选项转换为小写，这样就算用户输入大写对应的也是相应的小写选项

\*/

## 复习题

### E5.c

**/\*任务**：用户输入一个月份号，程序输出这一年到这一个月为止的总天数

\*

\* **知识点**：结构的定义、初始化、访问

\* \*/

## 习题

### 1.c

/\*修改复习题5

\* 任务：用户输入一个月份名称，程序输出这一年到这一个月为止的总天数

\*

\* 知识点：1.结构的定义、初始化、访问

\* 2. 字符串赋值，对比

\* 3.枚举类新

### 2.c

/\* 任务：用户输入一个月份,日期和年，程序输出这一年到这一天为止的天数。月份可以是月份名，月份号或者缩写

\*

\* 知识点：1.结构的定义、初始化、访问

\* 2. 字符串赋值，对比

\* 3.枚举类新

\*

\* 问题：判断用户输入的月份号那里，应该有更好的方法

### WW3.c

/\*对例题14\_2.c进行修改

\* **任务**：创建多本书的目录，包含每本书的书名、作者、价格信息。

\* 并且按照输入顺序，字母顺序，价格顺序（升序）进行输出

\* **知识点**：1. 利用指针类型的数组对结构体排序

\* \*/

### 4.c

对比4a.c和4b.c：

4a.c向子函数传递结构的地址

4b.c向子函数传递结构本身

1.传递结构本身一次只能传递一个结构，而传递地址却能够在子函数中操作/访问所有的结构内容，因此，显然传递地址要更加灵活，模块化更加清晰。

如果害怕在子函数中改变结构的值，完全可以在指针前面家const限定符

#### 4a.c

/\***任务**： 定义两个结构，

\* 结构1： 成员1：社会保险号 成员2： 结构2

\* 结构2： 成员1：first name 成员2：middle name 成员3：last name

\* 初始化并打印结构信息

\*

\* **知识点**：1.结构的嵌套使用

\* 2. 结构传址

\*

\* **算法**：

\* 定义结构2

\* 定义结构1

\* 初始化结构

\*

\* 定义子函数，参数为结构地址

\* 打印first name和last name

\* 如果没有中间名，就不打印中间名

\* 如果有中间名，打印中间名首字母，在后面加.

#### 4b.c

**知识点：**结构传值

### 5.c

/\*

**任务**：建立一个学生计分系统

**知识点：**1.结构操作

**算法**：

两个结构，初始化学生姓名

子函数1）获得学生成绩： 获得用户输入的学生姓名，匹配学生

匹配无效：输出无法匹配，重新输入

匹配成功：提示用户输入该学生的三科成绩（处理多余字符）；计算三科平均分，存入结构

子函数2）输出学生成绩：

子函数3）计算并输出班级信息：（各科和全部平均分）

\*/

### W6.c

/\***任务：**在例题要求的基础上分为3个功能

\* 1. 添加球员数据（如果球员号已经存在，那么直接显示球员名字，然后获取后面的信息）

\* 2. 删除球员数据（按照球员号）

\* 3. 显示球员数据以及全部统计信息

\*

\* **待实现**：1.希望用户输入球员号之后，如果列表中有这个球员，就直接确认其姓名，无需用户重新输入

\* （这样的话就需要打开文件来检查）

\* **算法**：

\* 声明结构member

\* 定义member类型的数组

\*

\* showmenu()显示选项菜单

\*

\* count\_data():统计文件中现有多少条球员数据

\*

\* 添加球员：void add\_member(struct member \*,int)

\* while(获取球员号)

\* 判断球员号是否已经存在

\* 是：显示球员名

\* 否：获取球员名

\* 继续获得球员其它数据

\* if（获取数据成功）

\* 调用add\_to\_file()，把新添数据写入文件

\*

\*

\* add\_to\_file:

\* 定义文件指针

\* a+b模式打开文件

\* fwrite()写入文件

\* 关闭文件

\*

\*删除球员：void delete\_member(void)

要求用户输入要删除信息的球员号

r模式打开原文件1

w模式创建并打开另一个文件2

循环读取文件1

如果是索引号匹配的信息，跳过

其他信息，写入文件2

关闭文件1、文件2

删除文件1

重命名文件2为FILE\_NAME

# 十五章：位操作

## 例题

### 15\_1.c

**/\*任务**：利用为操作把整数转换为二进制输出

\* 要求定义一个函数，参数为一个整数和一个字符串地址，

\* 程序把整数转化为二进制0/1序列放入字符串，然后输出字符串

\*

\* **算法**：获得用户输入的整数

\* sizeof求整数有多少字节，乘8求出位的数量size

\* 定义大小为size+1的char数组

\* 把数组名和整数传递给itobs()

\*

\* itobs():

\* while(没有到最后一位)

\* 定义掩码，只有最低位为1(说白了就是1，为了装逼，用八进制01)

\* 整数和掩码&操作获得i

\* i+'0'的值存入数组（由最大下标开始）=>因为数组中需要是字符而不是数字

\* 左移一位

**\* 知识点**：1.提取位

### 15\_2.c

\***任务**：切换一个值中的后n位，待处理的值和n都是函数的参数

\* 注：程序为自己的练习，和例题不同(例题一次生成掩码的方法值得借鉴)

\***算法**：获得用户输入的数值和n

inverse()将数值最后n位切换，返回数值

\* 用itobs()的方法求出0/1序列

\*

\* **知识点**:1. 生成掩码(使用位操作远比数学计算容易)

### 15\_3.c

/**任务**：输出窗口设置选项（包括：透明/不透明，填充颜色，边框风格，边框颜色）

//**知识点**：1.位字段的使用（节省空间）

// 2.这个程序提供了很好的“选择输出”的模板

//

**//算法**：宏定义各种选项（根据）

// 声明位字段结构模板

// 创建/初始化位字段结构

// 输出结构内容

// 改变结构内容

// 再次输出