**哈尔滨工程大学**

**《计算思维（二）》实验报告**

**基础实践二**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 班级： |  |  |
|  | 学号： |  |  |
|  | 姓名： |  |  |

实验时间： 2024 年 第02~12周

|  |  |
| --- | --- |
| 成绩 |  |

**哈尔滨工程大学计算机教育与实验创新中心**

实验五：数组

实验题目1：寻找子串

有两个字符串str1和str2，它们的长度都不超过100个字符。请编程实现在str1中查找str2的初始位置。

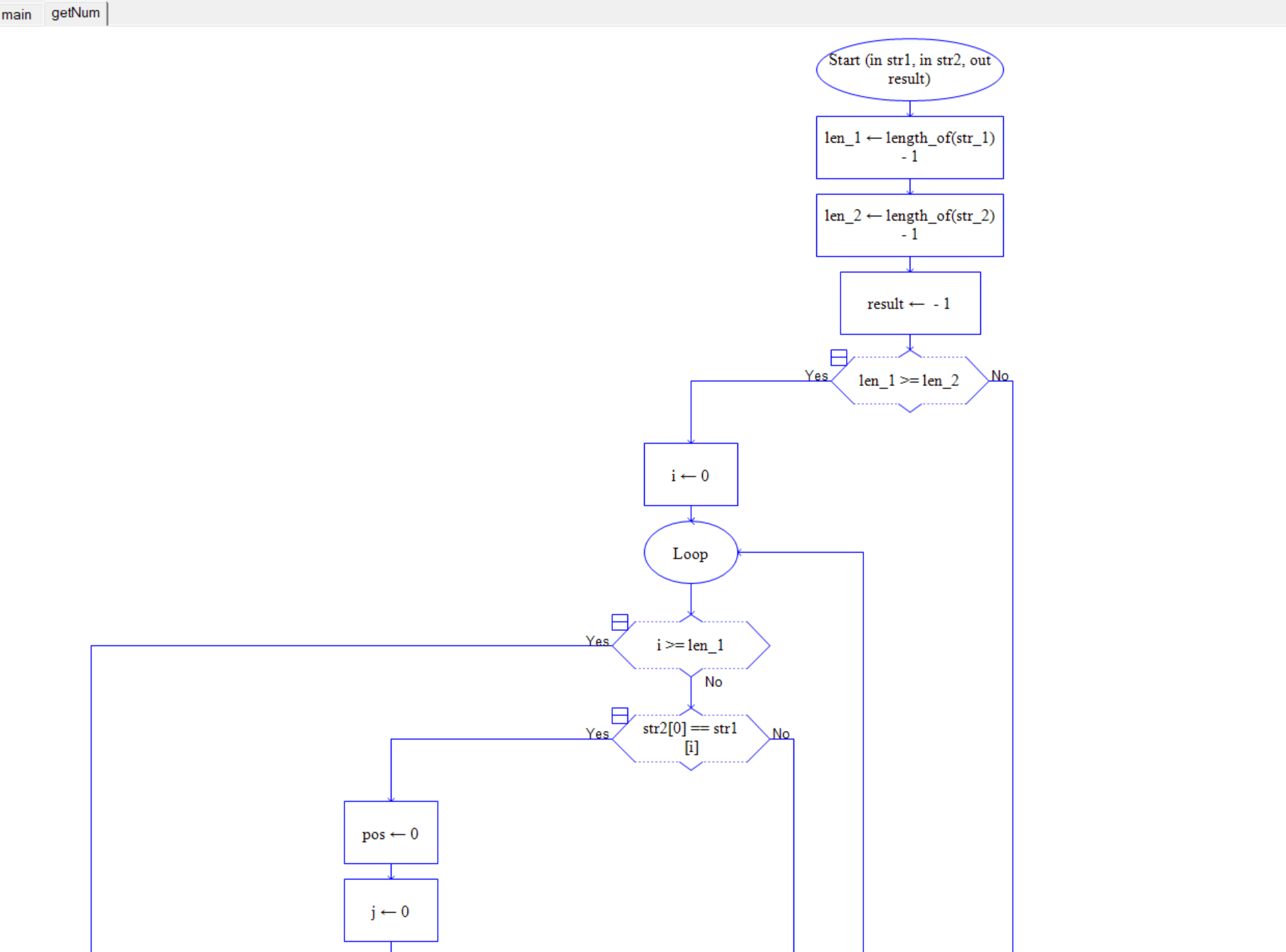
设计思想：

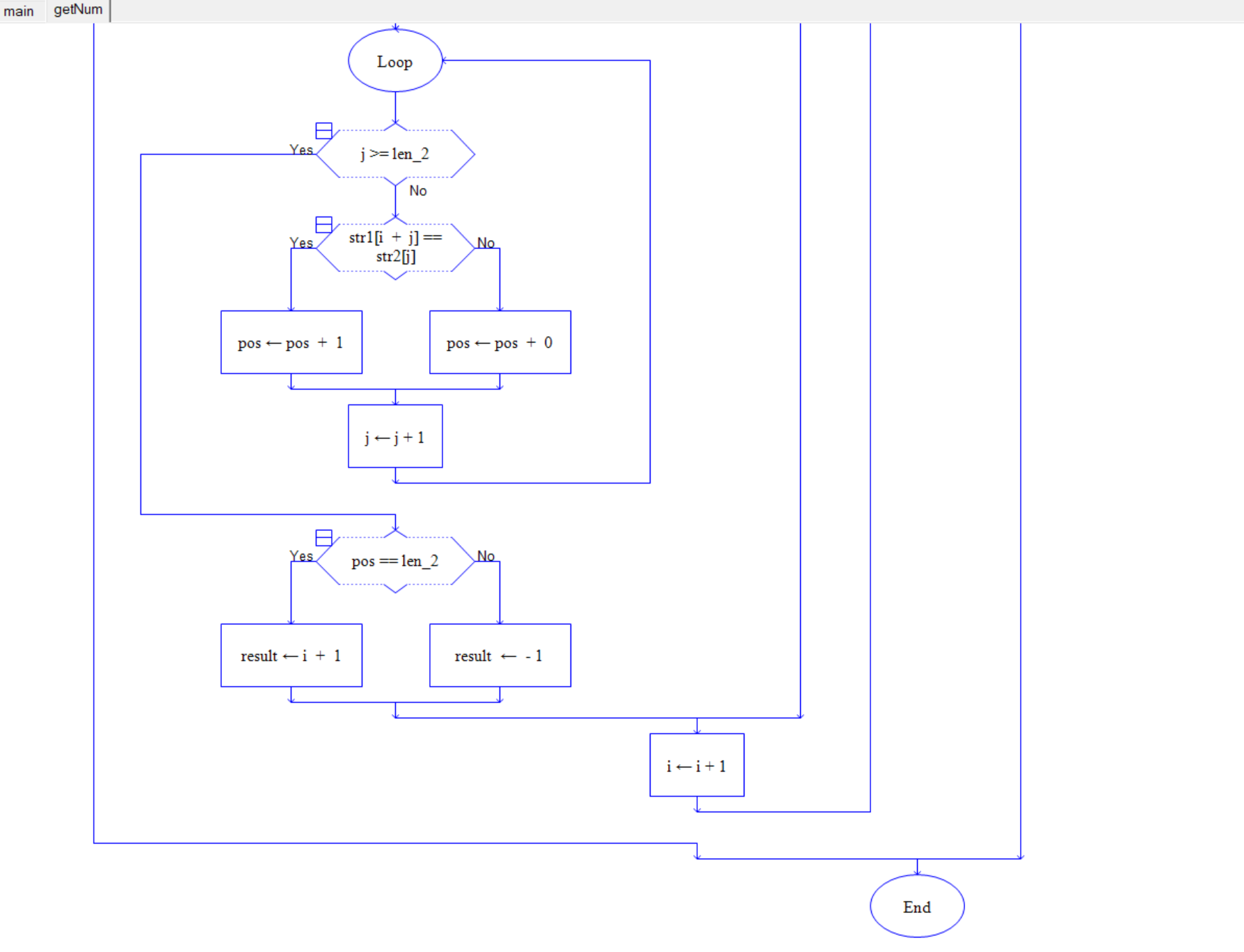
输入两个字串，用双重 for 循环来检测相同的子串，最后判断长度是否一致即可。

下面的流程图没有体现的是 getNum 函数内我将两个字串长度减一的操作，因为 fgets 函数会连同换行符一起输入，所以长度比预想的多1，要减掉。

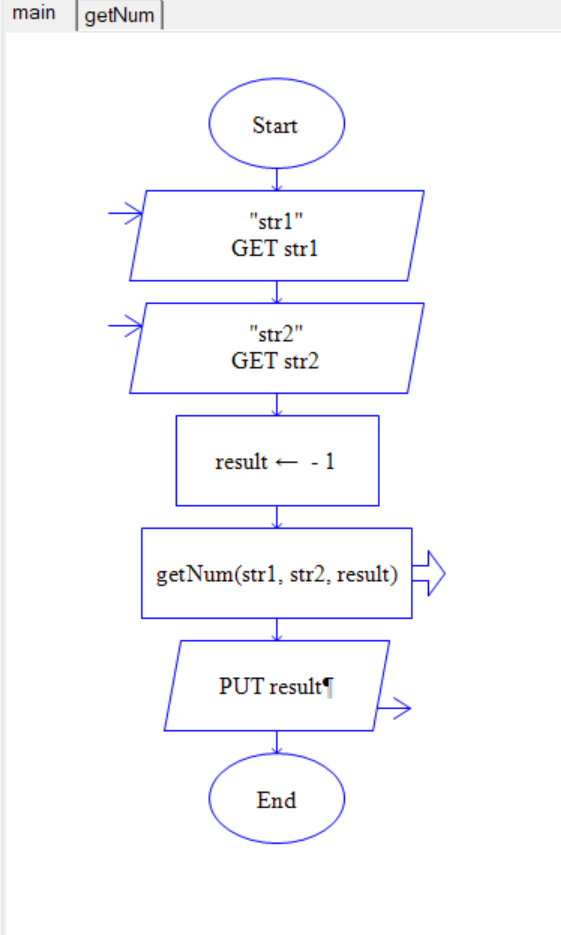
Raptor 算法流程图如下，

getNum 函数：

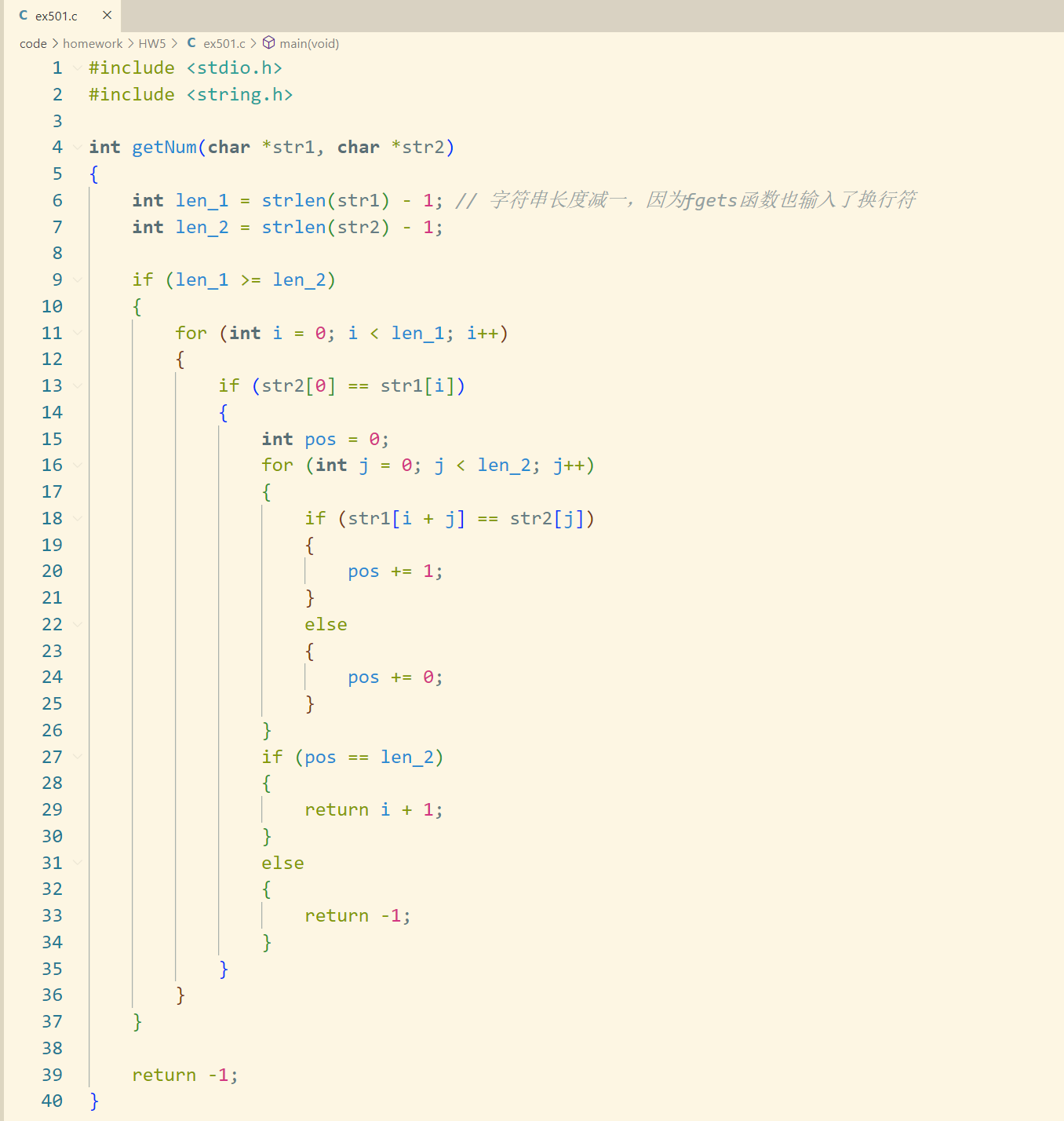


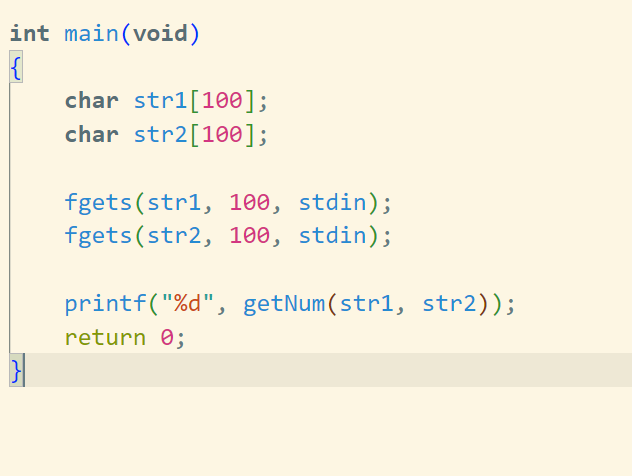


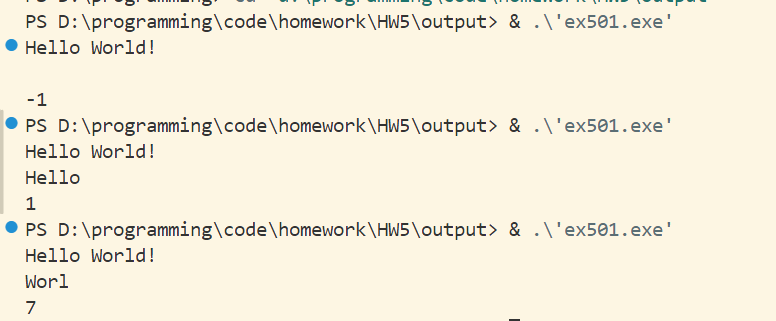
main 函数：



实验代码、验证与结论：









总结与心得体会：

首先困扰我好久的就是那个 fgets 函数，因为他把换行符也加进去了，导致长度多了 1，也就导致了我用 gets 函数时一切正确，但换成 fgets 就出问题了。

后来我先是上网查找二者的区别，有把每个变量都输出了一遍（当时还没有学会调试），最终才找到了这个问题。

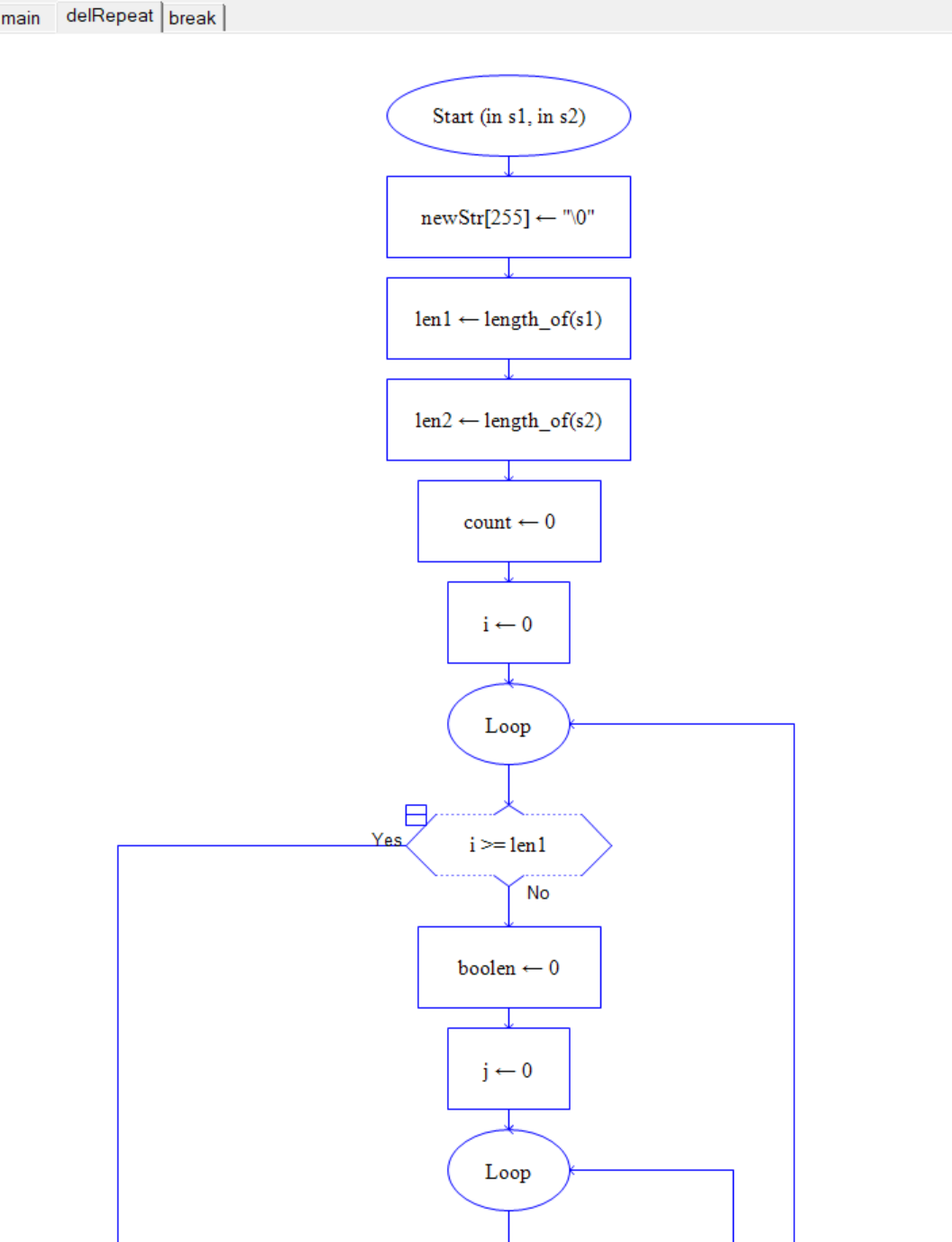
实验题目2：字串处理

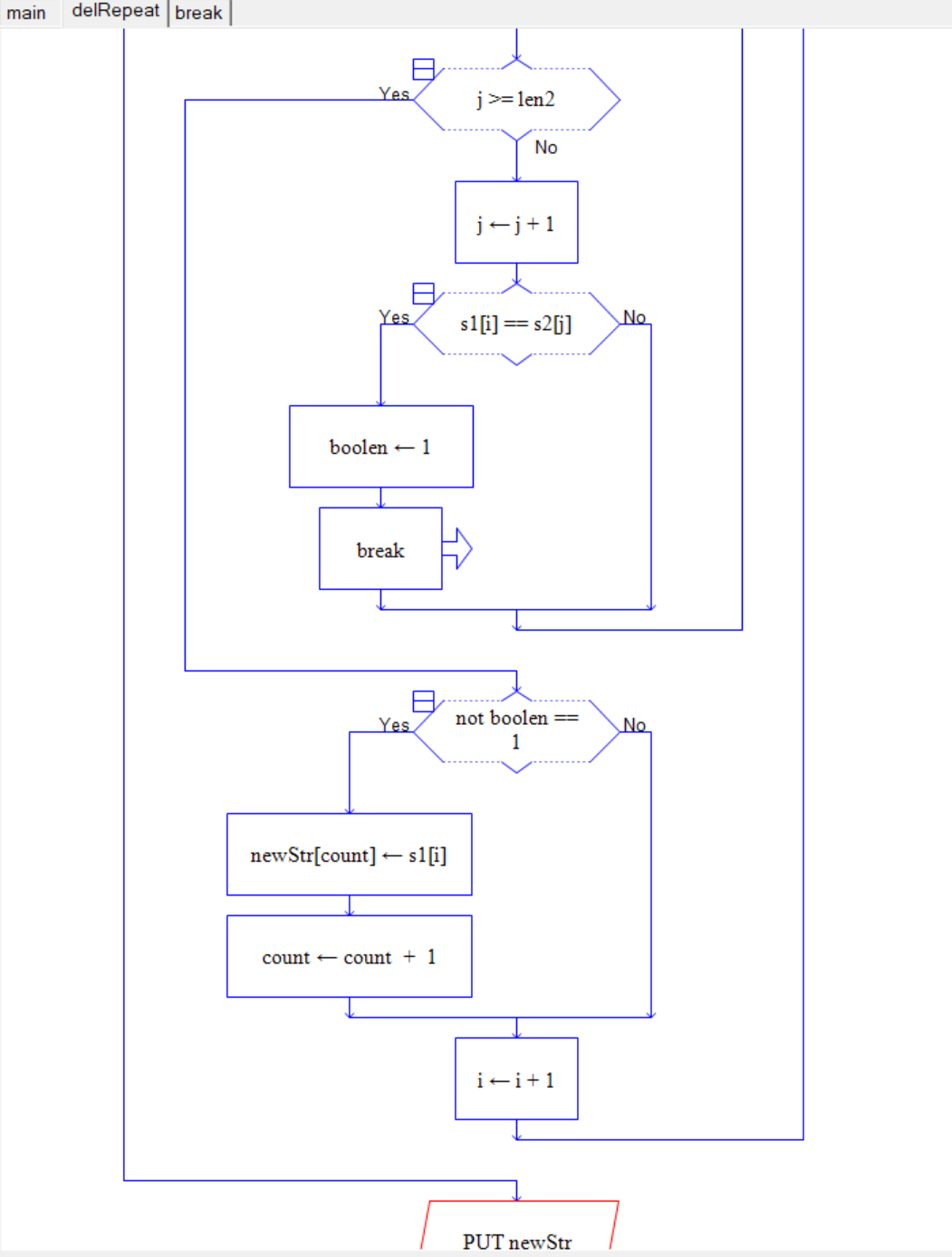
现有两个字符串s1和s2，它们最多都只能包含255个字符。编写程序，将字符串s1中所有出现在字符串s2中的字符删去，然后输出s1。

设计思想：

Raptor算法流程图如下，

delRepeat 函数：

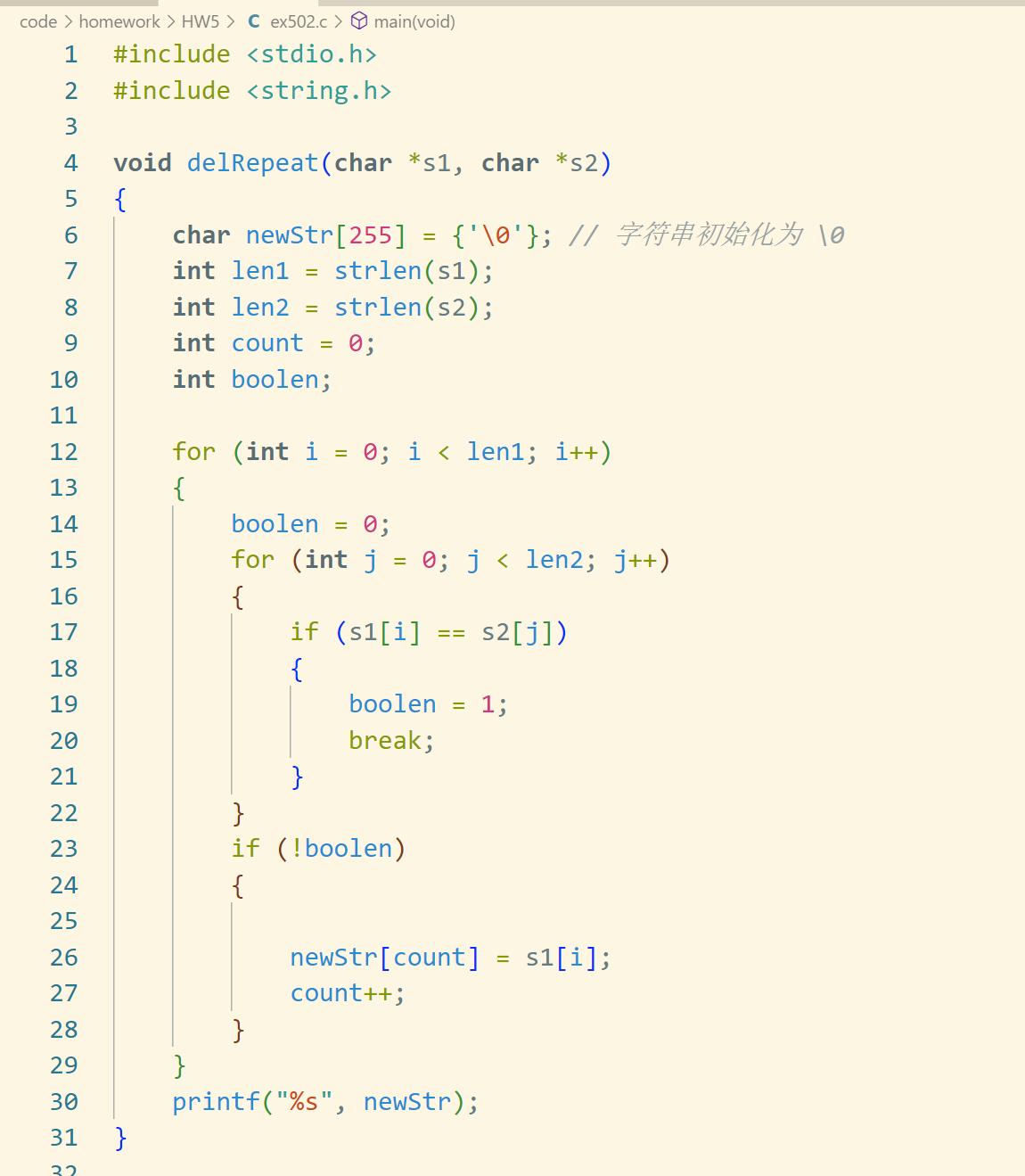


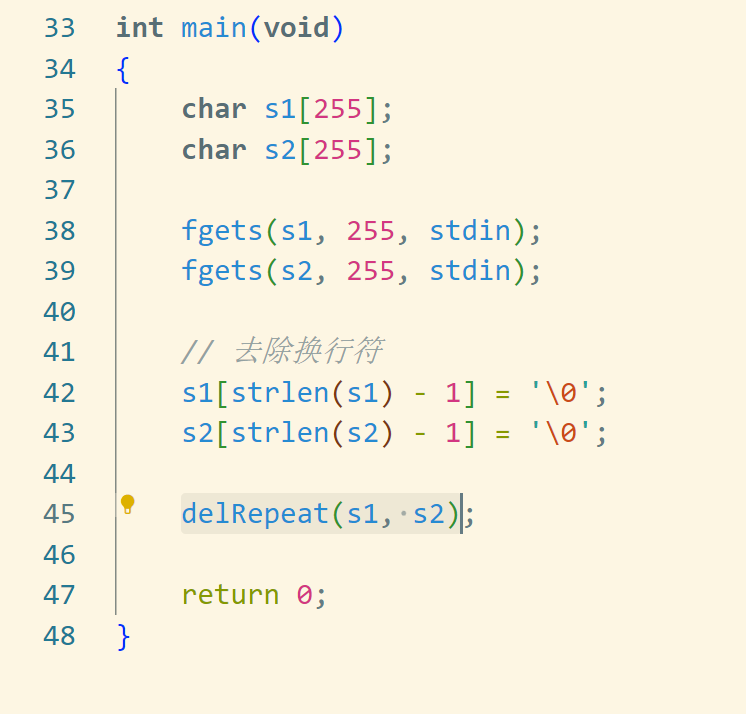


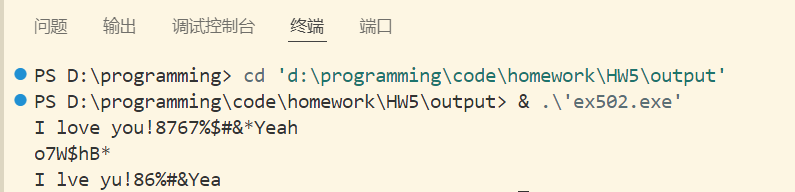
main 函数：



实验代码、验证与结论：









总结与心得体会：

这道题一开始发现处理完的字符串有乱码，后来尝试对字符串进行初始化为 ‘\0’ 就好了。

字符串去除换行符我换了与第一题不同的另一种方式来处理的。

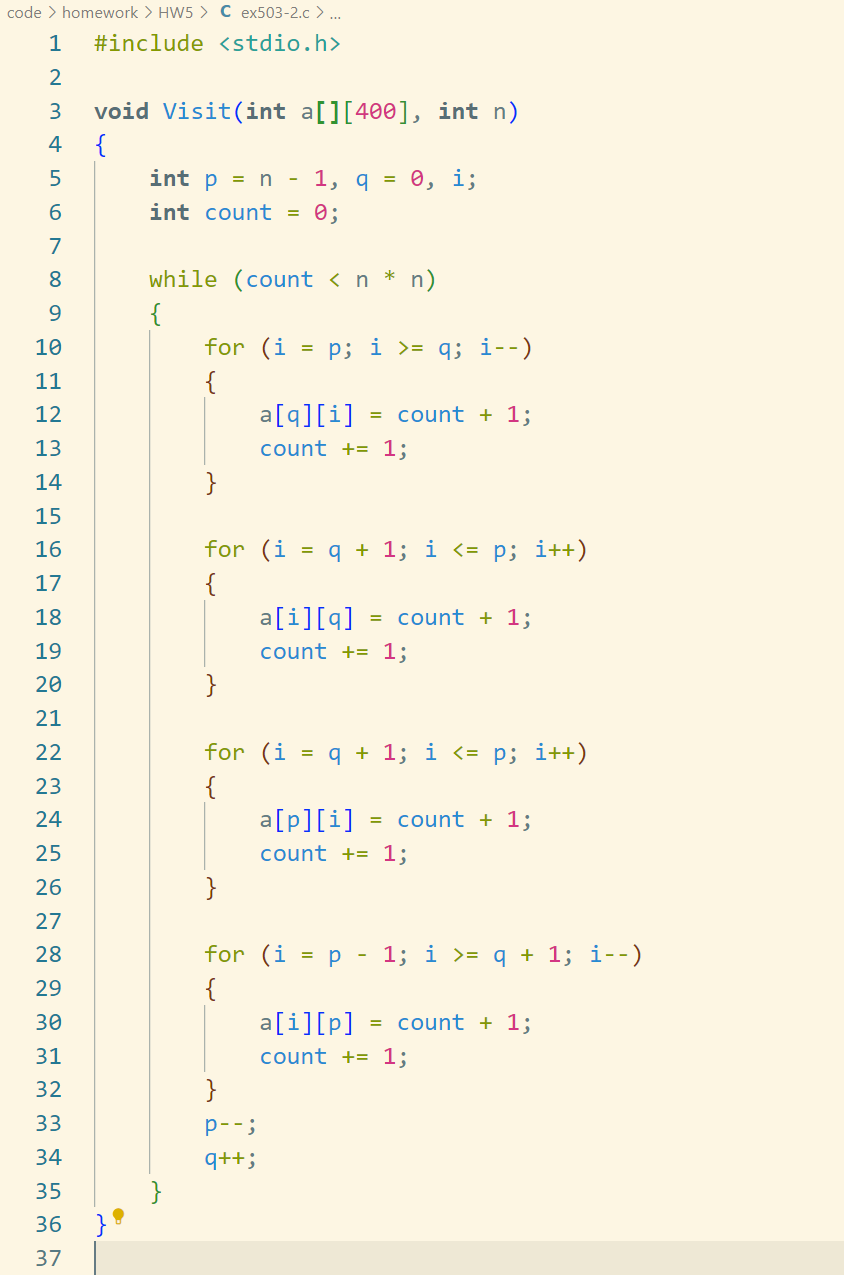
实验题目3：填充矩阵

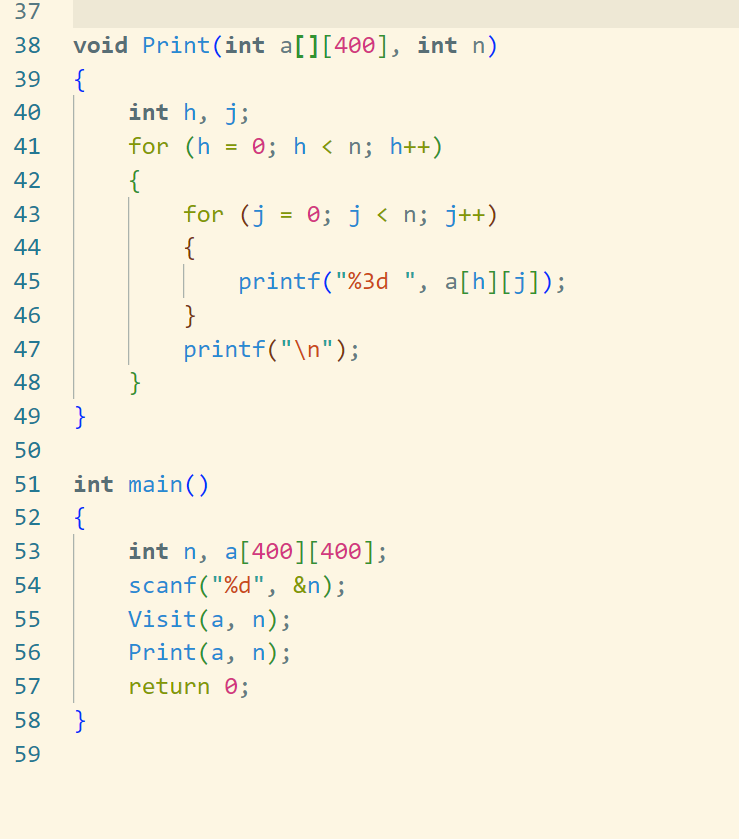
编程实现自动填充n×n矩阵元素数值，填充规则为：从第一行最后一列矩阵元素开始按逆时针方向螺旋式填充数值1，2，…，n×n，其中：n从键盘输入且3≤n≤20。最后向显示器输出该矩阵所有元素。

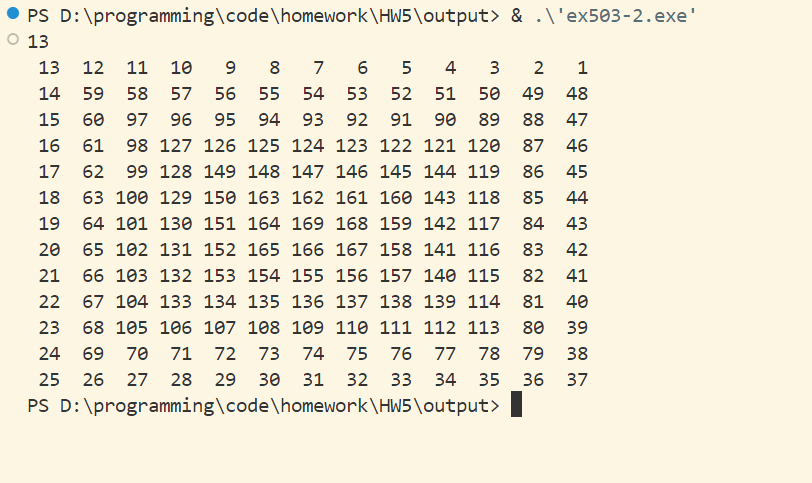
设计思想：

这个是通过一个外围边界向内收缩的方式，以螺旋的方式来填充矩阵元素的。

实验代码、验证与结论：









总结与心得体会：

我最开始的思路是改变二维数组索引值来完成上下左右自由填充的效果，并加上边界的判断条件，但在填充的时候总是弄错方向，不是螺旋填充而是以其他各种各样的方式来填充的，改了好久，最后借鉴同学的才写出来。

那么新的代码是不断缩小填充的范围来完成的，非常符合正常的逻辑。

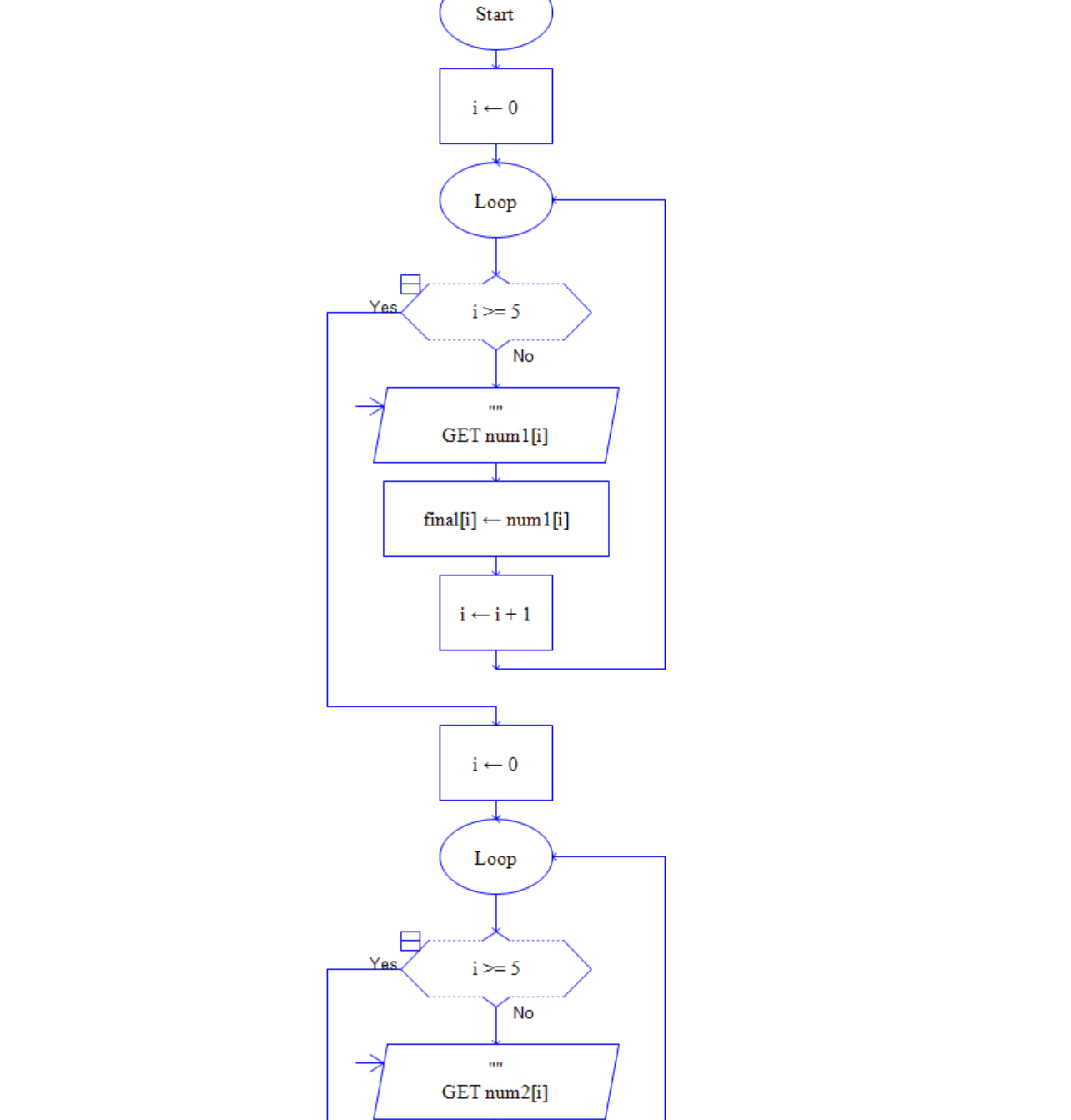
在想代码的时候用演算纸写一写，会更容易弄懂。

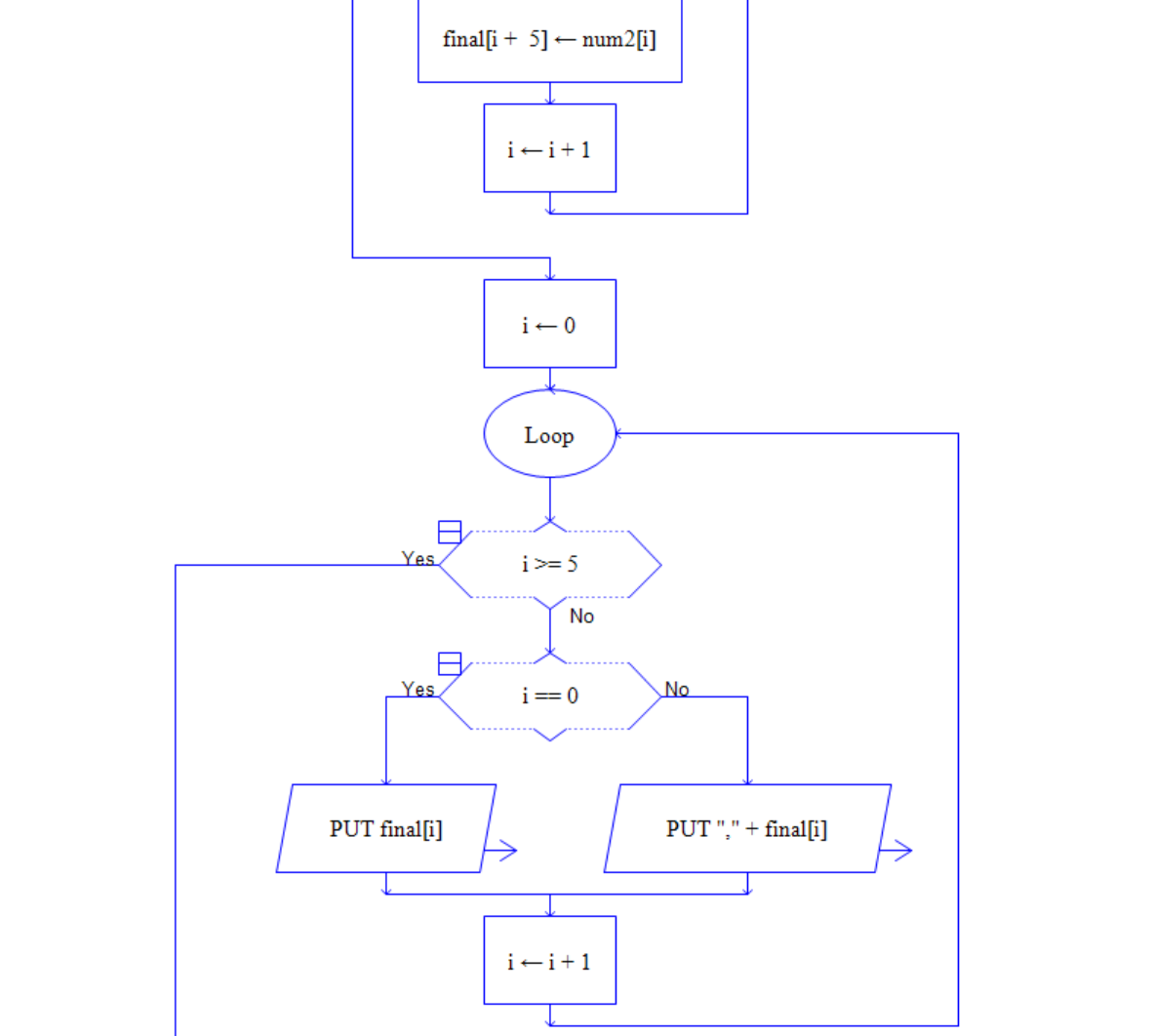
实验题目4： 数组合并与排序

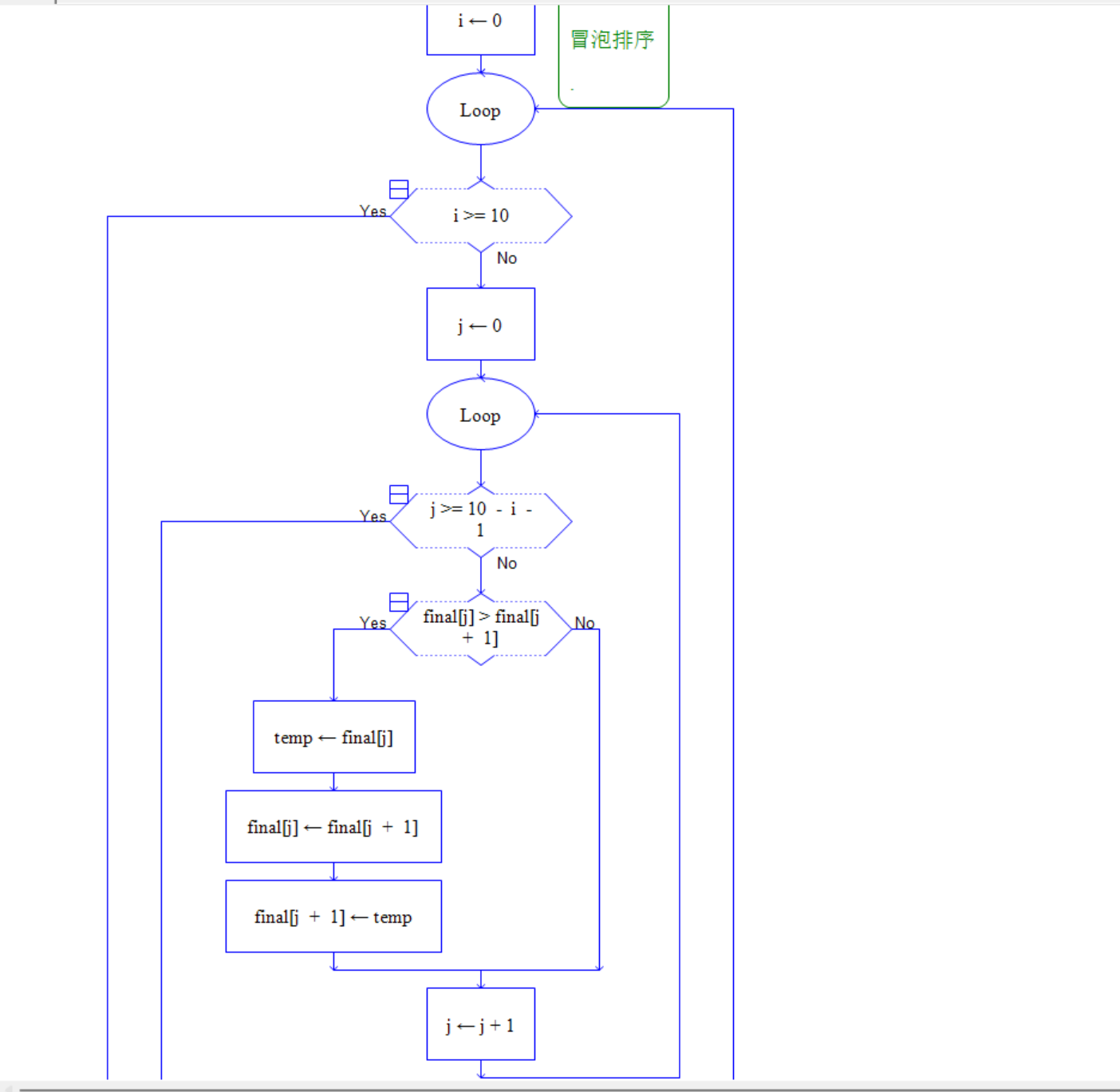
输入两个整数数组，每个数组有5个整数，将二者进行合并，然后按照数值从小到大排序输出。

设计思想：

Raptor 算法流程图如下：

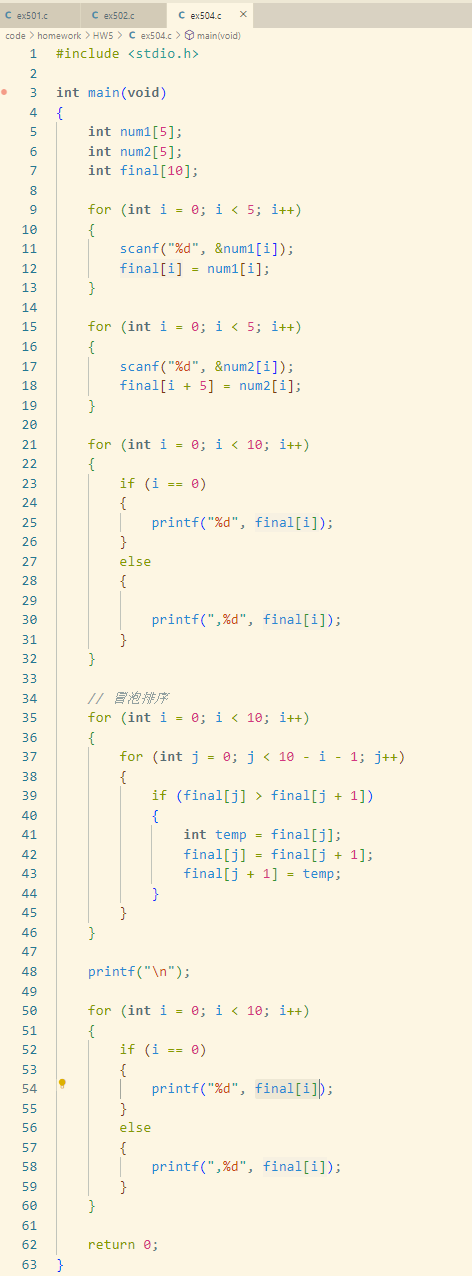


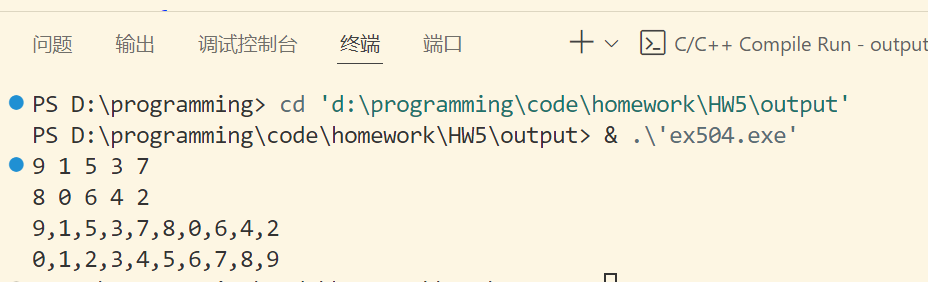






实验代码、验证与结论：







总结与心得体会：

这道题中用到了冒泡排序，我也是在演算纸上画了每一个步骤才开始理解双重 for 循环的意义。

其余的地方都是以前常规的内容。

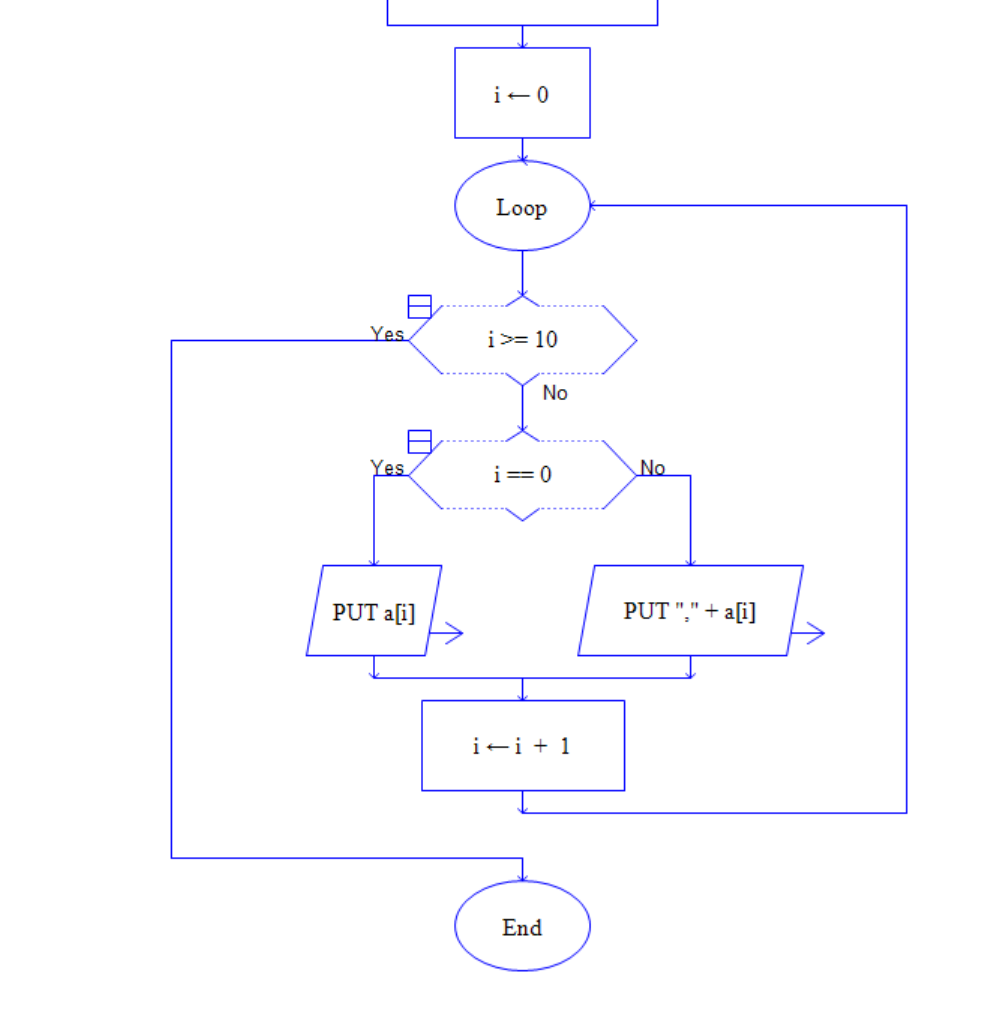
实验题目5： 数组元素遍历

输入一个整型数组a[10]，并计算其中的前9个元素的平均值，然后将这个值替换a[9]中的内容，最后输出该数组的所有元素。

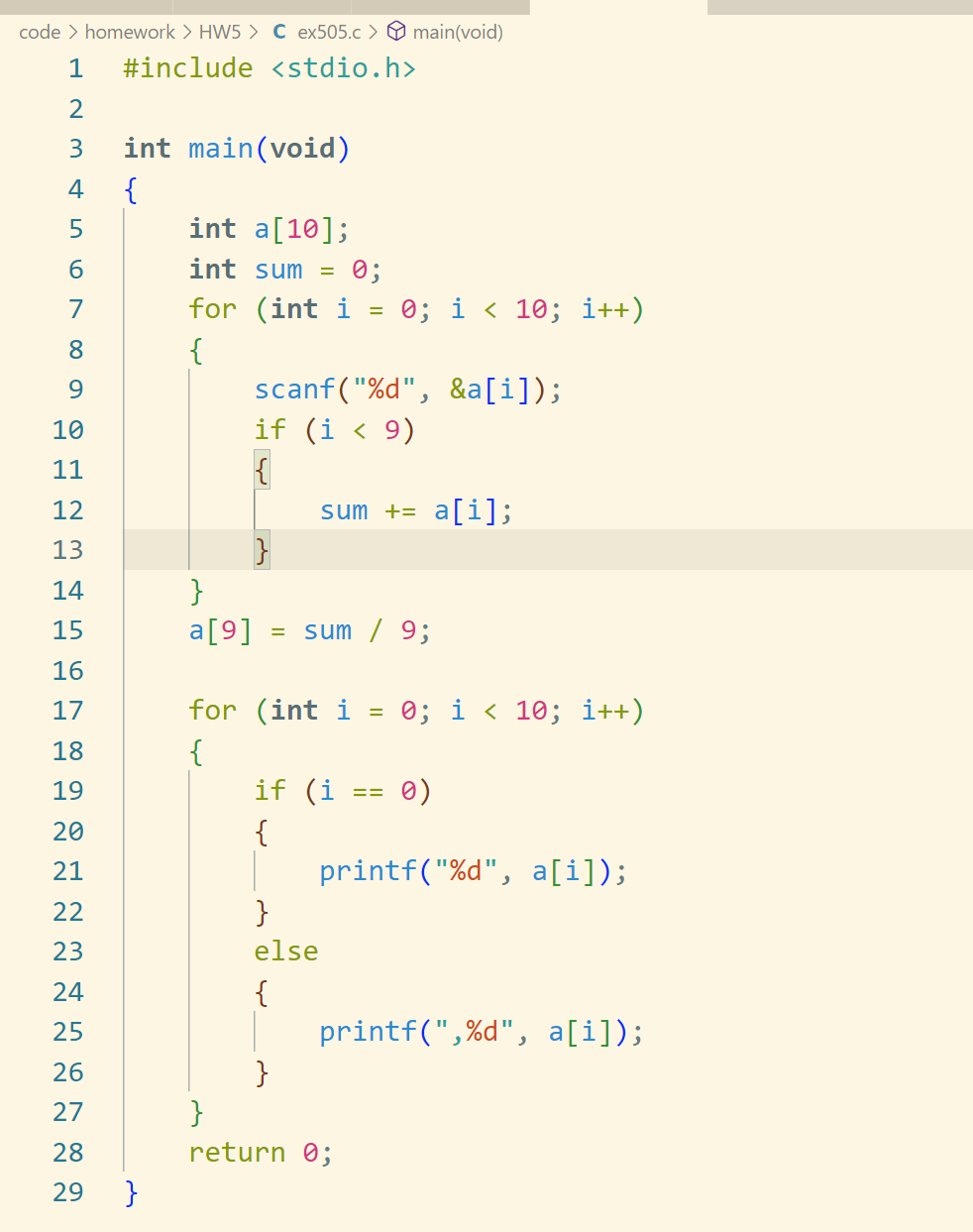
设计思想：

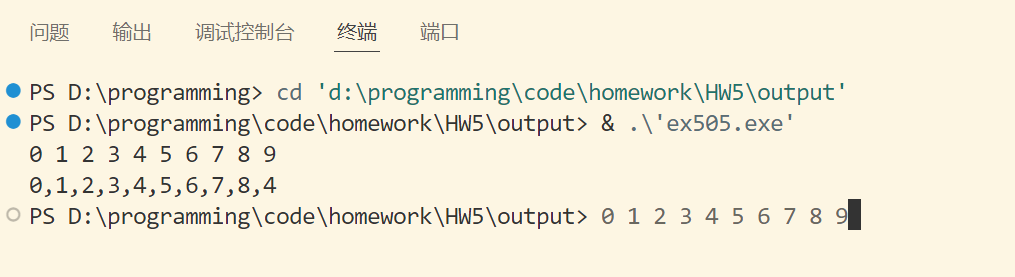
Raptor 算法流程图如下：





实验代码、验证与结论：







总结与心得体会：

完成数组啦！

实验六：指针

实验题目1：循环移动数组元素

定义一个长度为10的一维整型数组，并编写三个函数分别实现如下功能：

（1）函数input：对数组的各元素实现从键盘输入赋值。

（2）函数output：将数组的所有元素向屏幕打印输出。

（3）函数moveToRight：向右循环移动数组元素

（4）函数moveToLeft：向左循环移动数组元素

（5）函数move：将数组元素向左或向右循环移动m个位置。移动规则为：

● 若m>0，调用moveToRight函数，执行向右循环移动；

● 若m=0，不做任何移动；

● 若m<0，调用moveToLeft函数，执行向左循环移动。

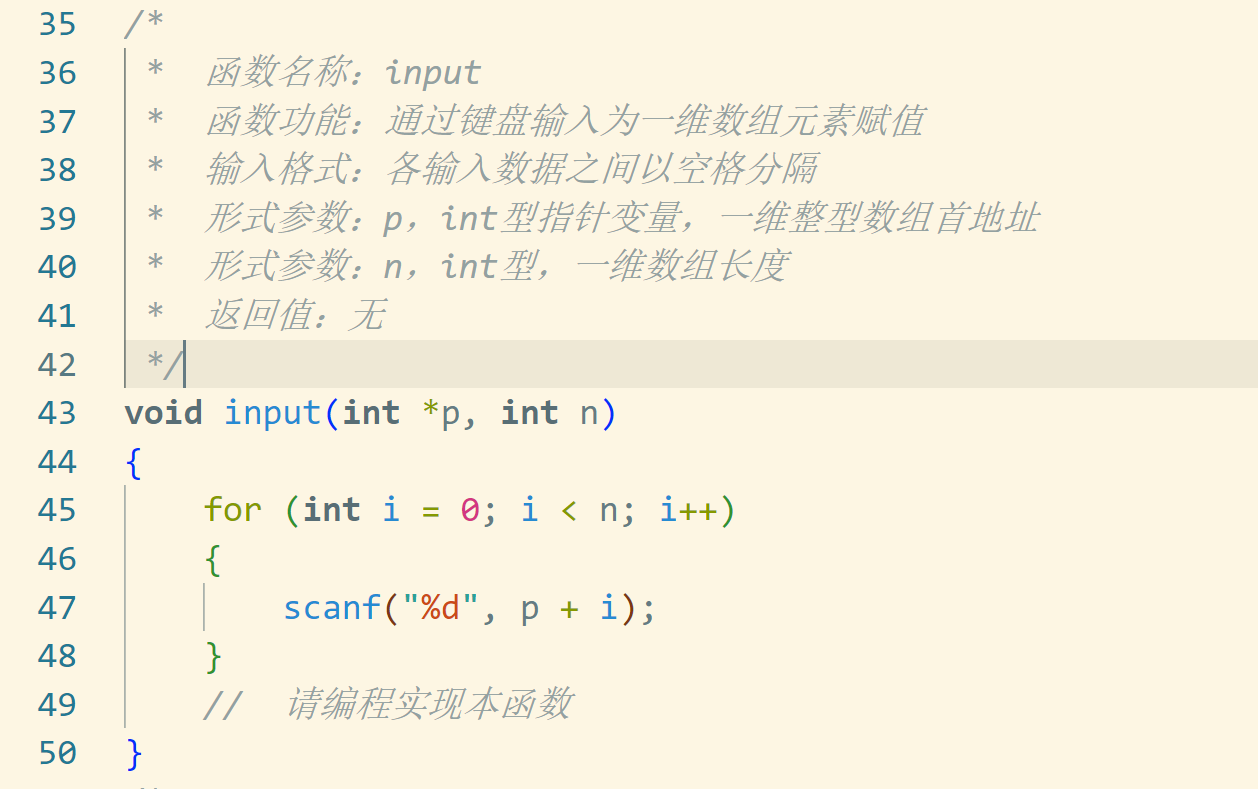
目前已编写完成main函数和move函数，请编程实现input函数、output函数、moveToRight函数和moveToLeft函数。

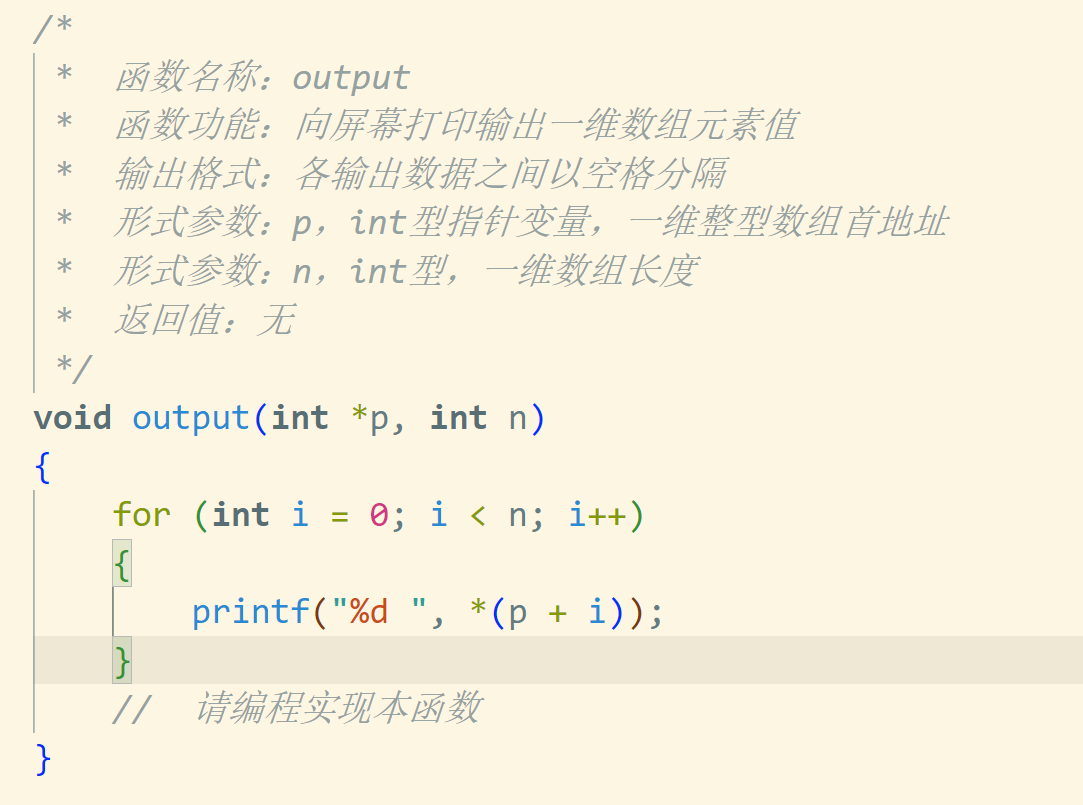
设计思想：

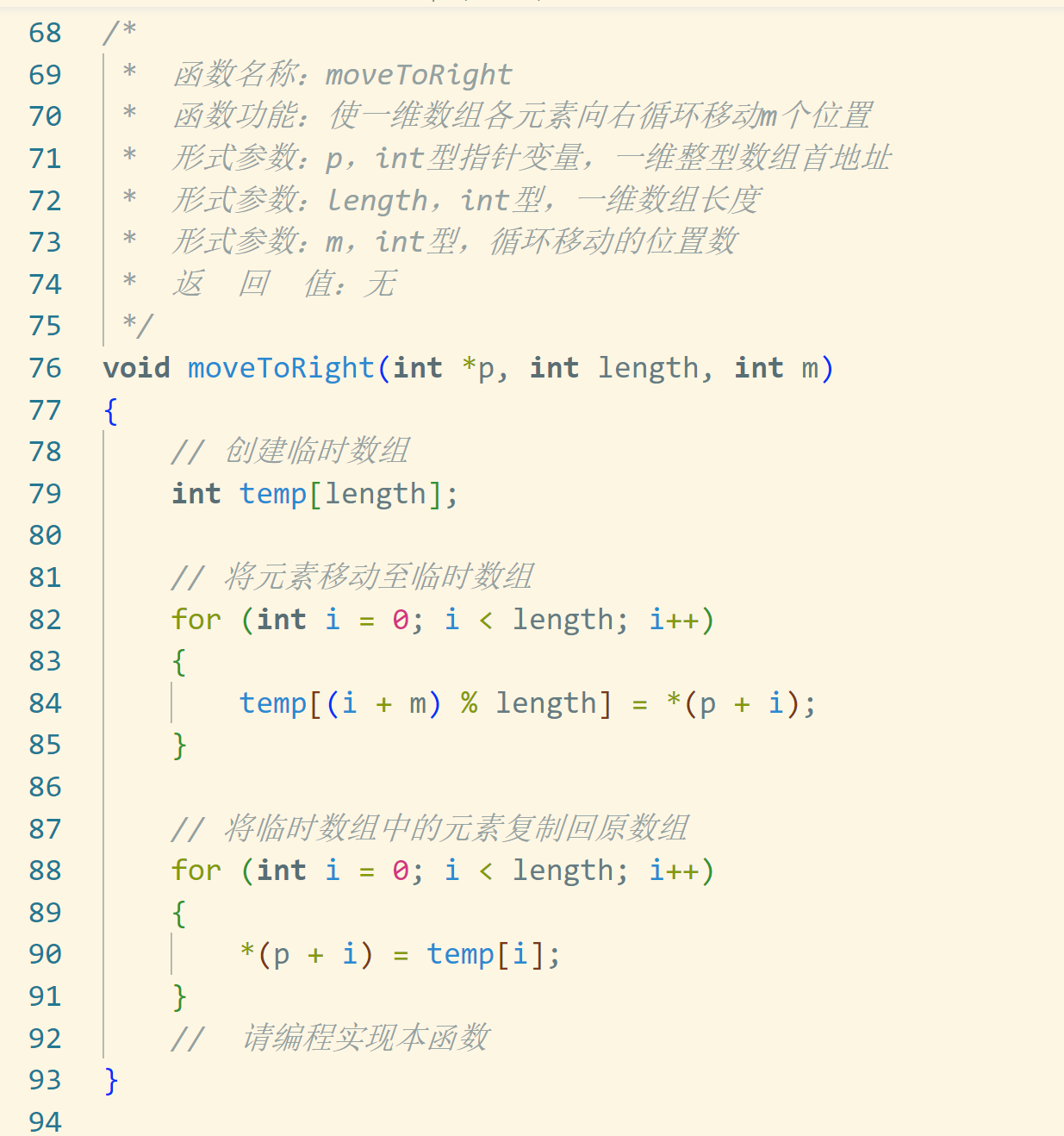
得益于老师写的很规范的函数前的注释，不需要联系全文就可以完成填空了，非常轻松。

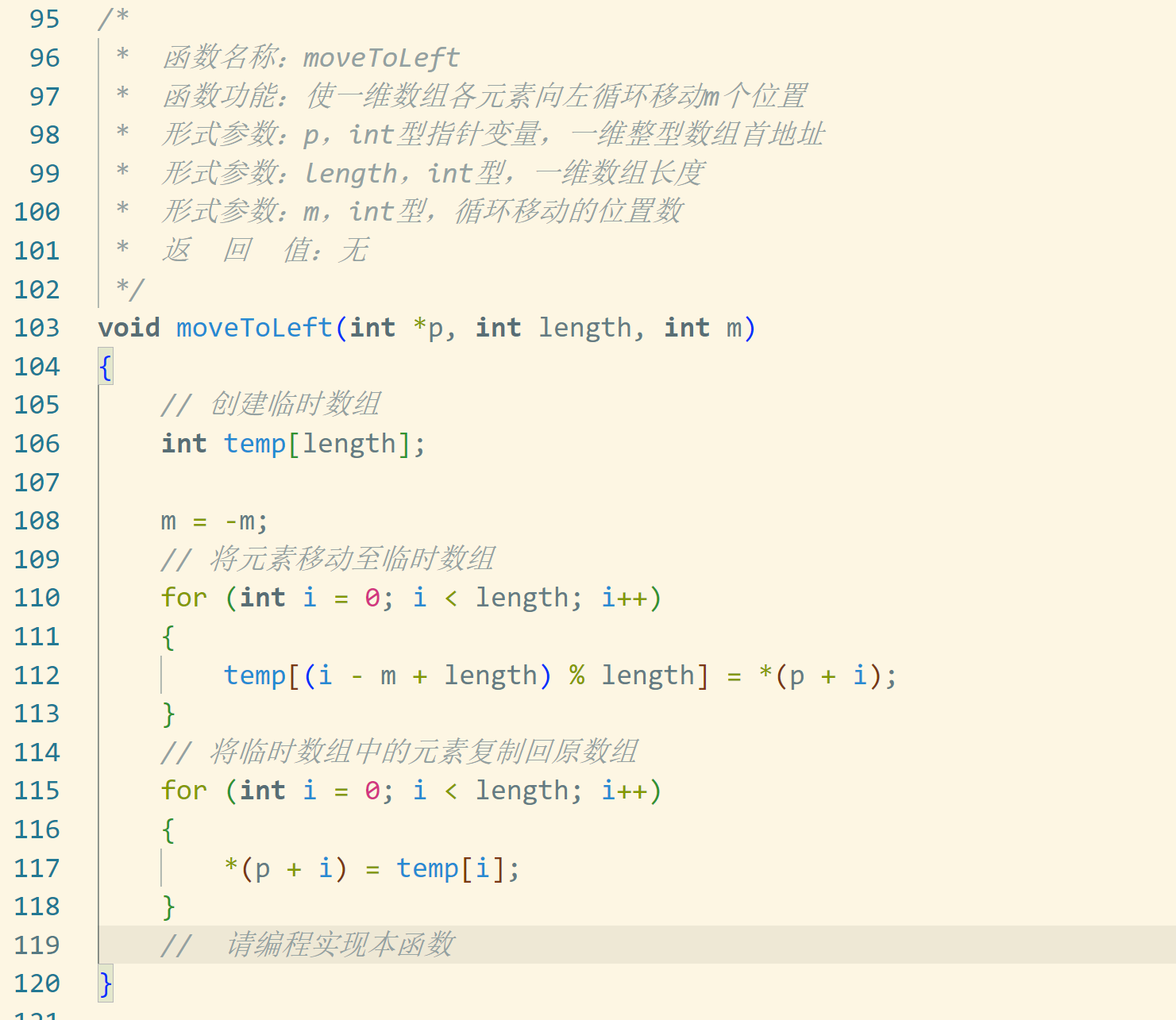
不过第一题的命名是 ex605.c，第五题的是 ex601.c，看来老师将题目顺序打乱了。

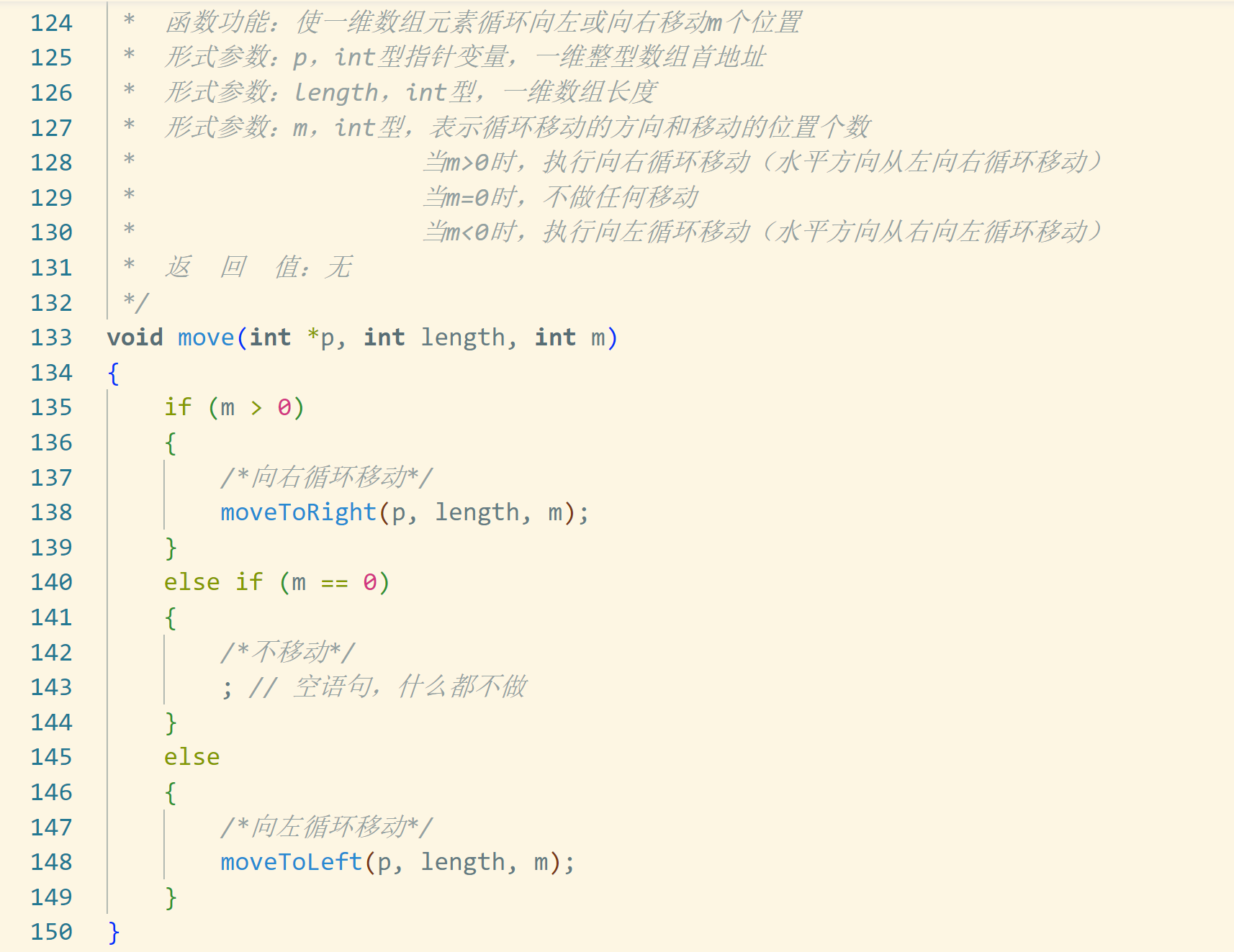
实验代码、验证与结论：













总结与心得体会：

这次收获是非常大的！老师的注释将函数的功能、参数的作用以及变量的含义写明后，我在填空时就会非常轻松。

在看到了这份注释的样例后，我以后的代码注释也要写成这种规范的格式。

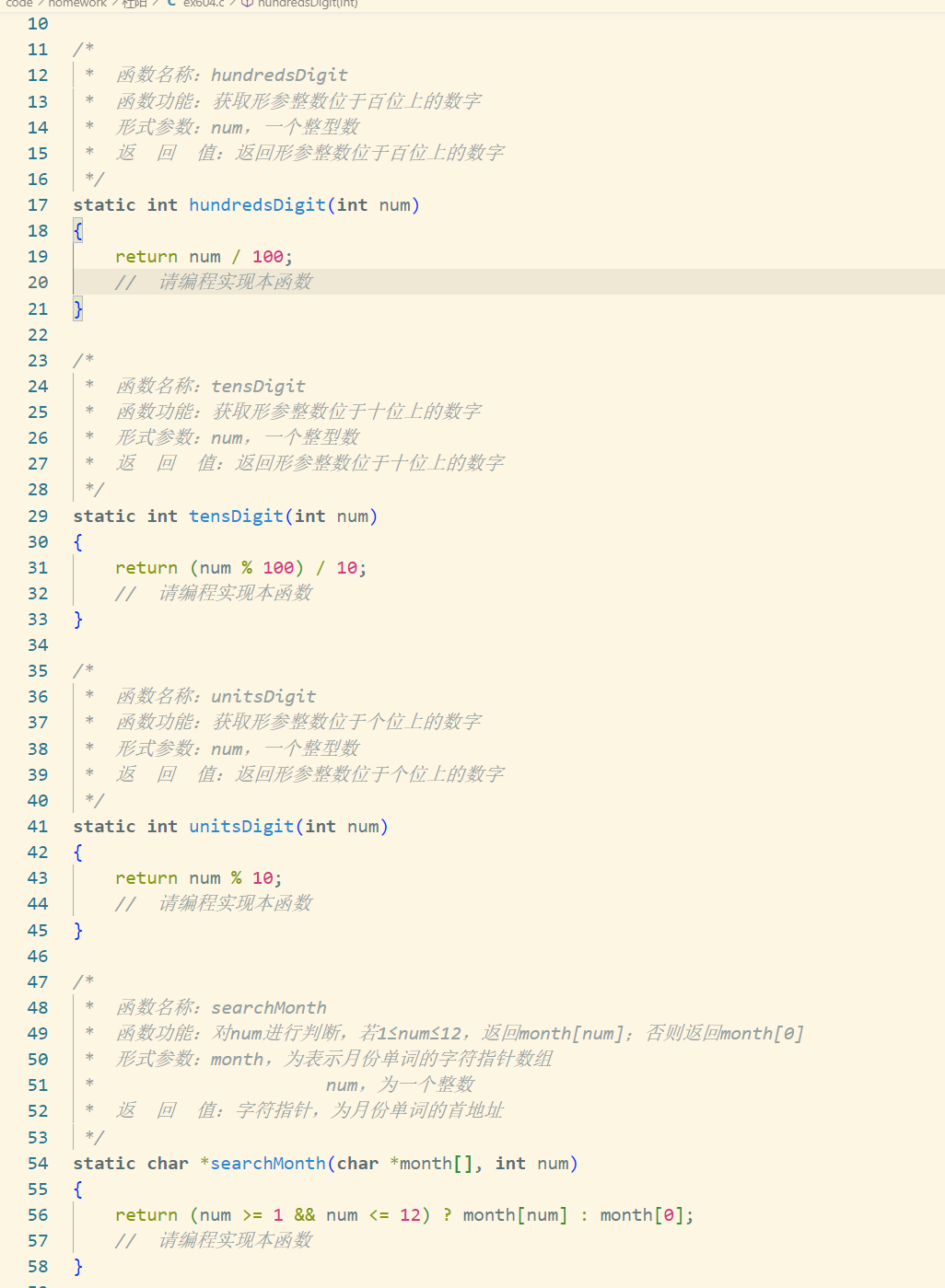
实验题目2：数字与月份

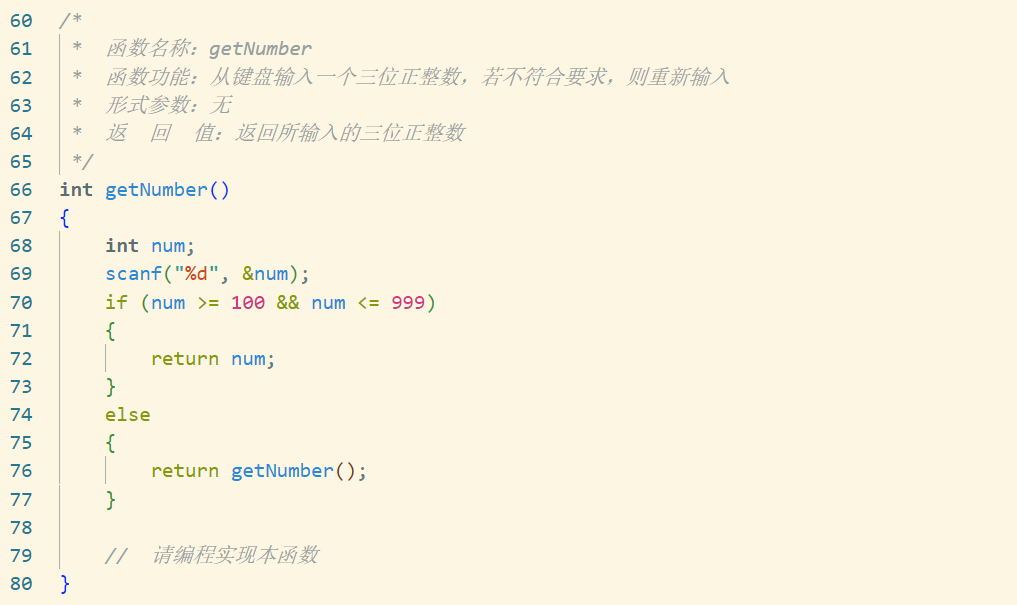
输入一个三位数，计算该数各位上的数字之和，如果在[1, 12]之内，则输出与数字之和相对应的月份的英文单词，否则输出\*\*\*。要求：用指针数组记录各月份英文单词的首地址，且不得使用全局变量。目前仅完成了main函数和bizProcess函数，请编程实现hundredsDigit函数、tensDigit函数、unitsDigit函数、searchMonth函数和getNumber函数。

设计思想：

这里面用到了 static 关键字控制了函数的作用域，又学到了新东西。

实验代码、验证与结论：







总结与心得体会：

结合老师课上讲的内容，又上网查找了相关的资料，这次学会了 static 这个新的知识点。

除了指针的传参，其余的知识点都是之前的内容了，有了老师的注释，写起来特别方便。

实验题目3：单词排序

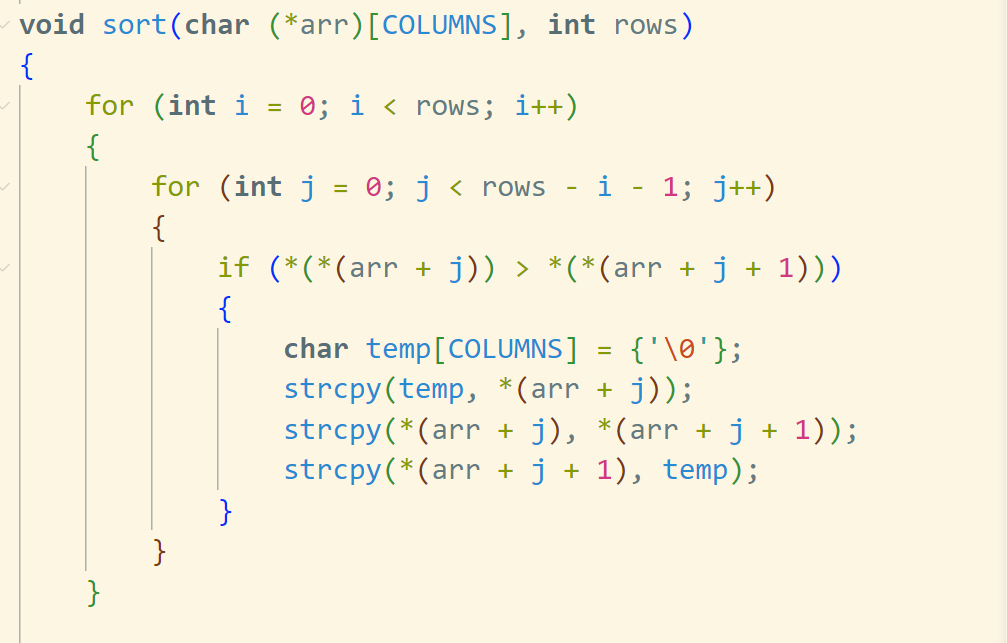
编程实现从键盘输入5个英文单词（每个单词不超过20个字母），然后将它们按照字典序（即字母从小到大的顺序）排序后打印输出。要求：编写assign函数、sort函数、show函数，分别用于实现单词的键盘输入、单词的排序、单词的输出。目前，已完成main函数的编写，请编写assign、sort和show函数，对这三个函数的定义及要求如下所示。

设计思想：

主函数里只有变量的定义和函数的传参，剩下的所有功能都写到各个函数里了，统一由主函数调用，非常的模块化。

实验代码、验证与结论：







总结与心得体会：

之前我习惯于把两三个代码写到一个函数里，但读老师的代码发现，一个函数只写一个功能，每个函数的作用更加明确，也便于修改。

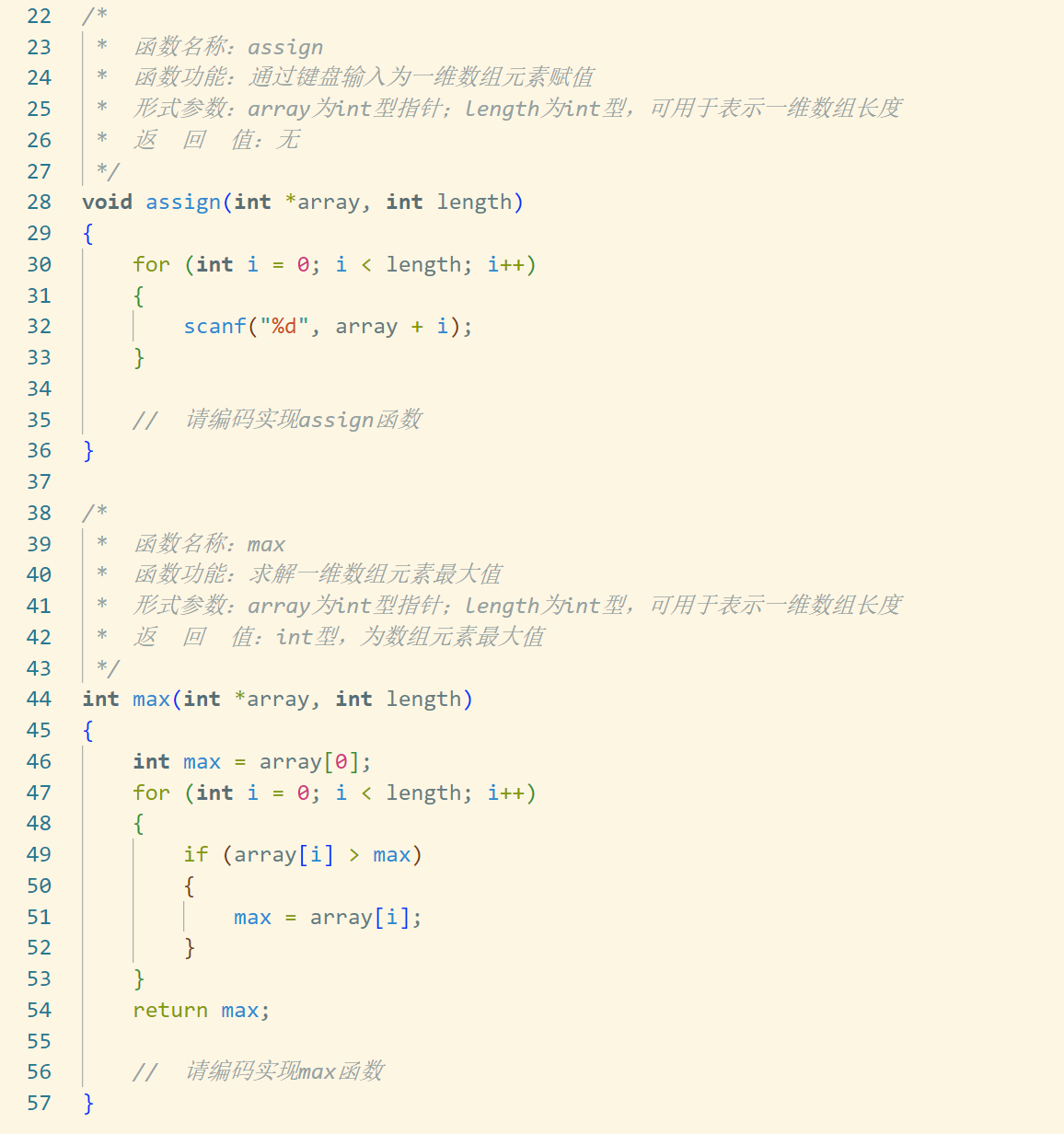
实验题目4：指针作为函数参数

一维整型数组a有5个元素，请编写assign函数和max函数分别实现为数组元素赋值和求解数组元素最大值。目前，除了assign和max这两个函数以外，其他代码已编写完成，如下所示。请编码实现上述两个函数。

设计思想：

取数组最大值的算法我记得上学期的导论里介绍过，这次就用到了：先取第一个元素作为最大值，如果后面有比它更大的，就

实验代码、验证与结论：





总结与心得体会：

由于给好了形参，完成起来还是非常容易的。

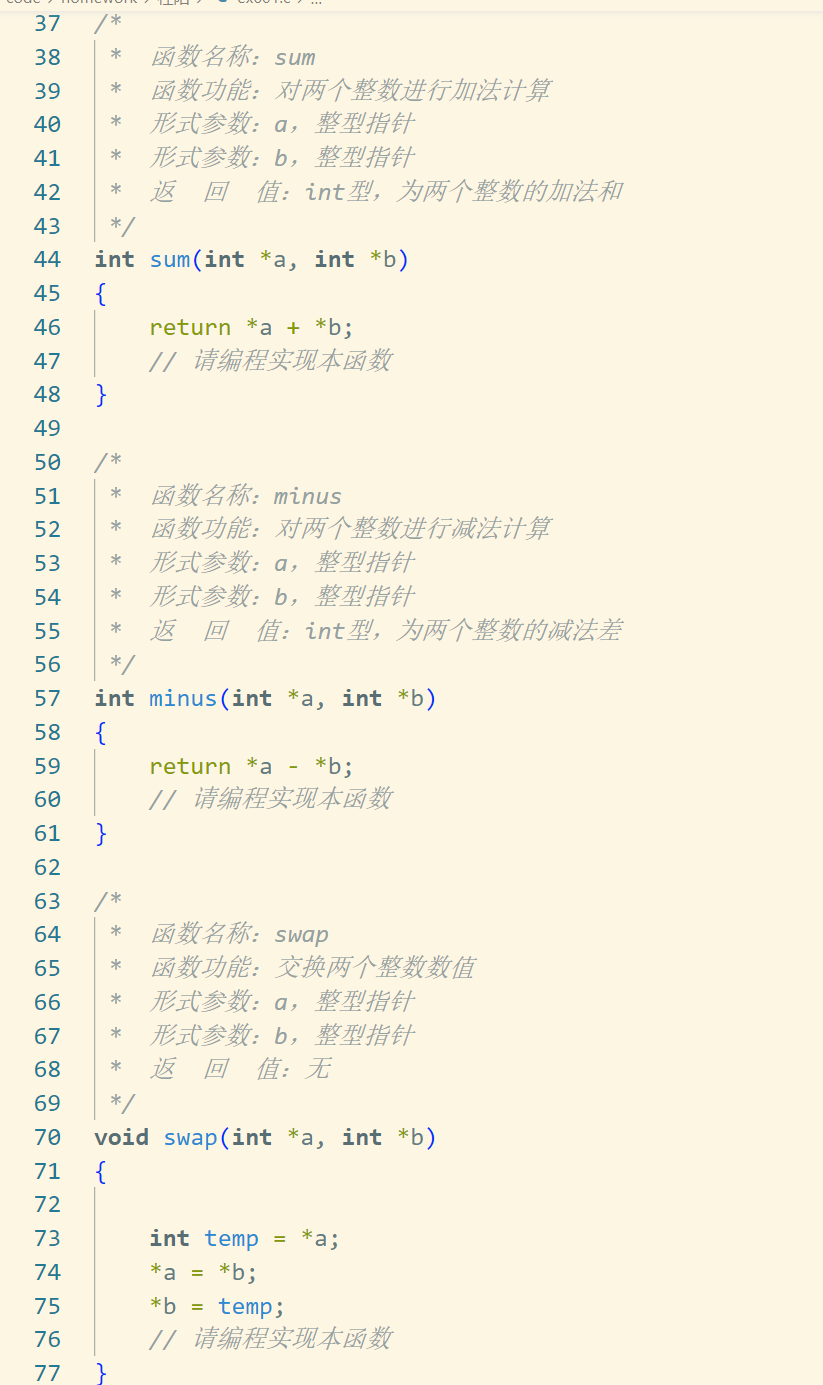
实验题目5：指针形参及函数指针应用

任意输入两个整数，编写三个函数分别实现：（1）计算两个数的加法和；（2）计算两个整数的减法差；（3）交换这两个整数的数值。要求用“函数指针”调用这三个函数，结果在主函数中输出。目前已编写完成main函数，请编程实现sum函数、minus函数和swap函数，函数功能和要求如下所示。

设计思想：

这个交换变量的函数，是书上讲的例子，其余的也只是导入函数进行相加和相减，比较容易一些。

实验代码、验证与结论：



总结与心得体会：

这些题目做完后，对指针更加了解了，可以更熟练地运用了！

实验七：结构体

实验题目1：教师信息管理

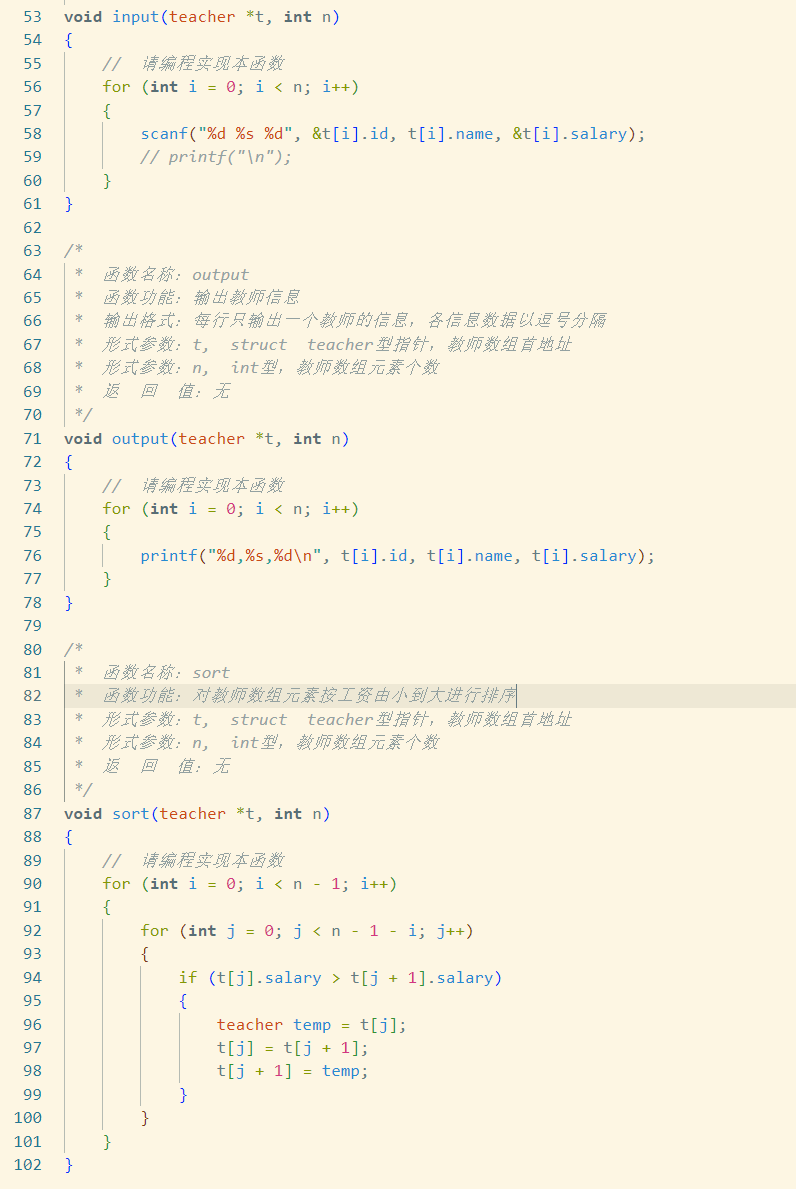
定义一个包含5个教师成员的结构体数组，教师信息包括职工号、姓名和工资。要求从键盘输入所有教师的信息，并设计一个sort函数实现结构体数组按教师工资由少到多排序。目前，已编辑完成main函数，请编程实现input函数、output函数和sort函数，分别实现教师信息的录入、输出与排序。

设计思想：

这里面用到的结构体可以把它当成一个新的数据类型，有点像之前我学过的一些编程语言的面向对象里面的类的属性，也可以给它创建一个实例。

这样理解起来，感觉就比较直观了。

实验代码、验证与结论：



总结与心得体会：

面对一个新的知识点，可以从已有的知识点上去推理，把未知换成已知。

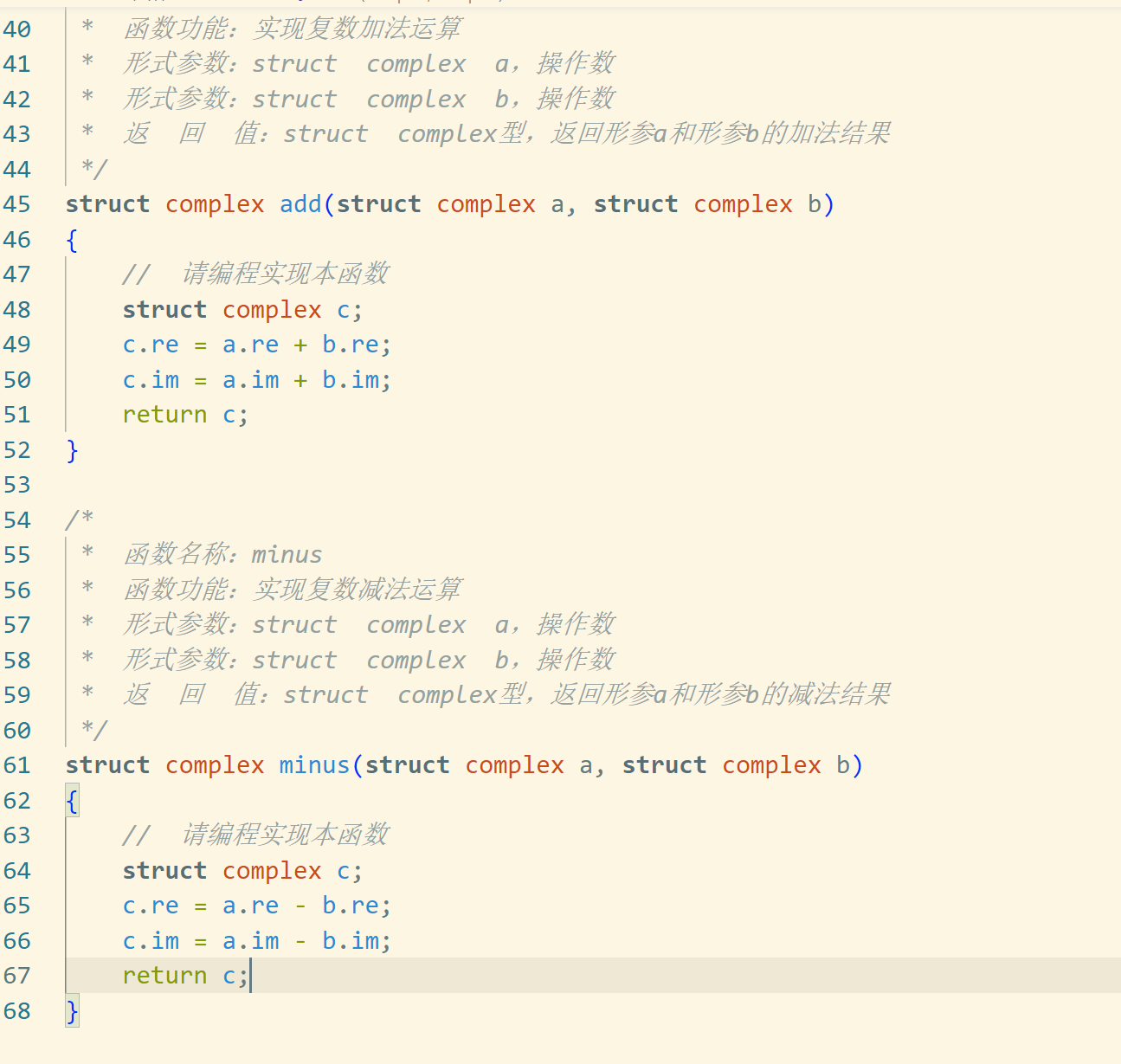
实验题目2：复数加减法运算器

定义一个结构体类型，用于描述复数结构数据。分别编写函数实现复数的加法和减法运算，在主函数中调用这些函数进行计算并输出计算结果。目前已完成main函数的编写，请编程实现add函数和minus函数。具体要求如下所示。

设计思想：

同样的思路，代码比较简单。不过这里要注意，两个函数的返回值仍然是结构体。

实验代码、验证与结论：



总结与心得体会：

这道题也是让我见到了结构体的另一个应用吧，积累到了更多用法。

实验题目3：简单的图书管理程序

现有3本图书，每本图书的信息包括：书号、书名、作者、价格。

（1）编写input函数，实现从键盘输入若干本图书信息，存放在结构体数组中。

（2）编写output函数，遍历结构体数组，输出所有图书信息。

（3）编写findBookByID函数，实现按书号检索图书。

（4）编写findBookByName函数，实现按书名检索图书。

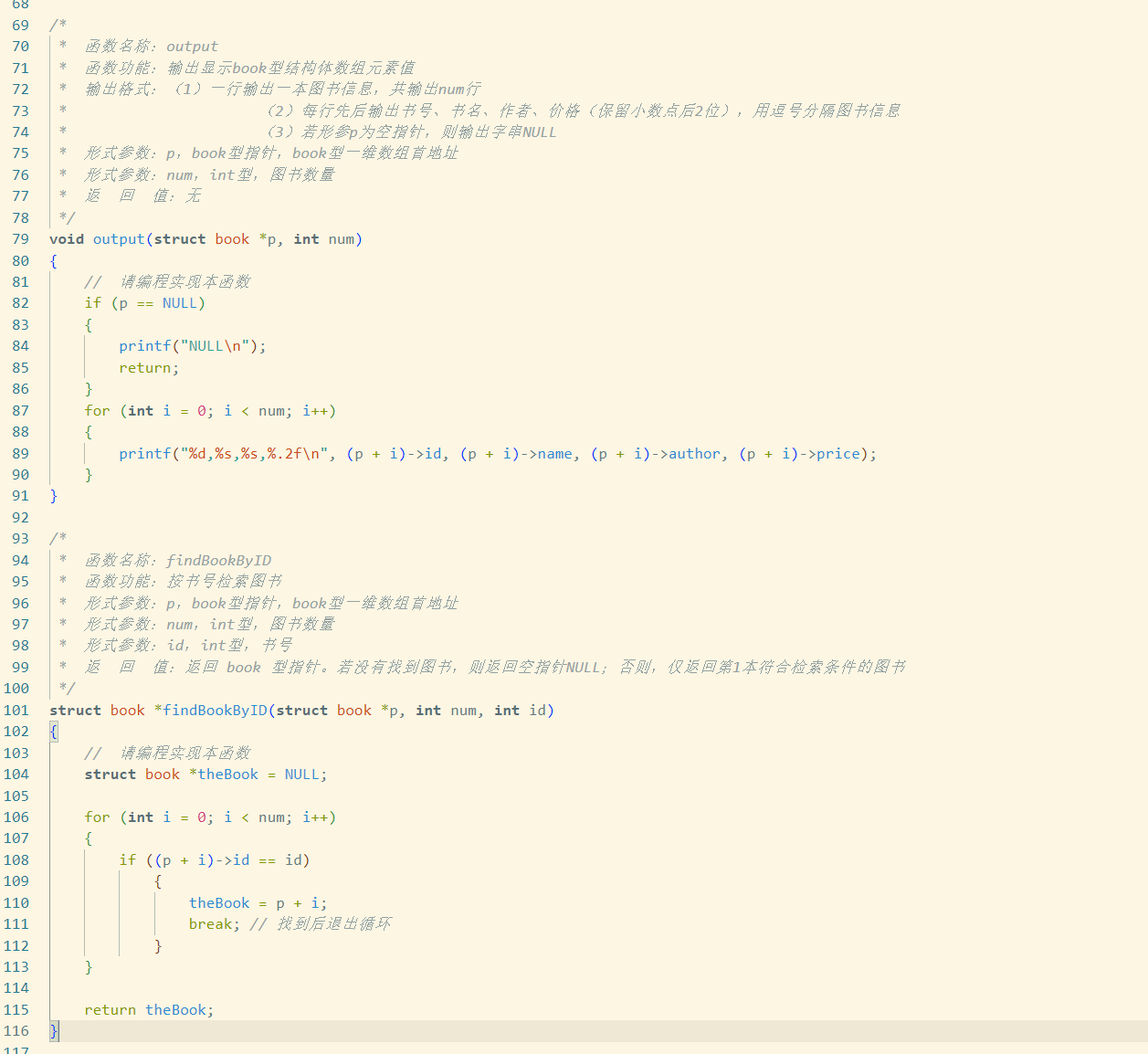
目前，已编写完成main函数、input函数和findBookByName函数，请编程实现output函数和findBookByID函数。

设计思想：

这里面多了一个新的运算符 ->，类似于 . 运算符 ，只不过是针对指针类型。

理解了这个，那剩下的就和之前的差不多了。

实验代码、验证与结论：



总结与心得体会：

学会了结构体指针，又掌握一个知识点！

实验题目4：反转单链表

编程实现对一个单向链表的结点进行反转，也就是将链表结点按逆序排列，排序后链头将成为链尾，而链尾将成为链头。自定义函数名称及函数功能要求如下：

（1）定义“createNode”函数，实现创建一个新结点。

（2）定义“createList”函数，实现一个单向链表的创建。

● 在创建结点时通过键盘输入为每个结点的数据成员value进行赋值。

● 使用“头插法”将每个结点逐一插入到链表中。

（3）定义“display”的函数，实现对链表所有结点的遍历输出。

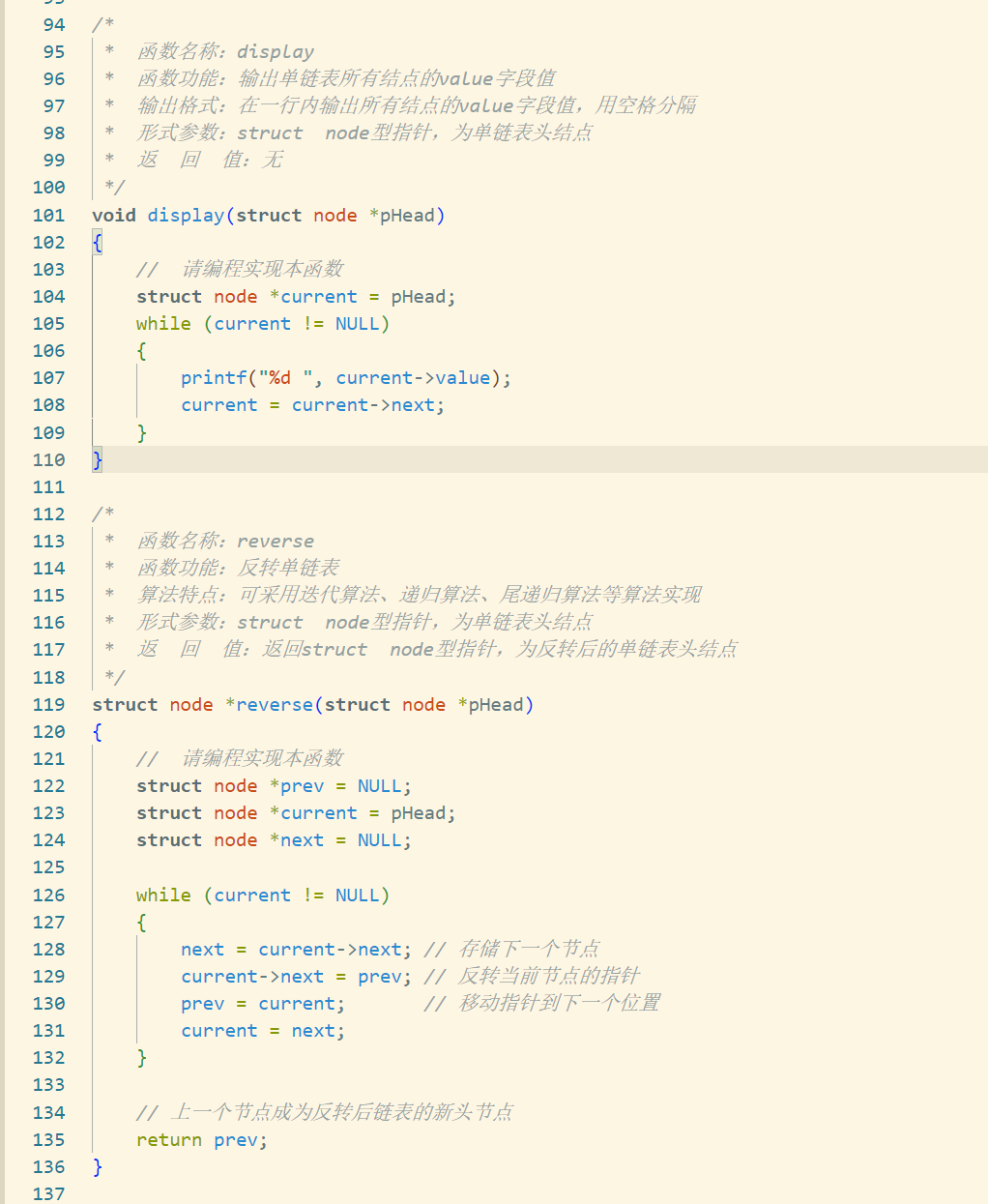
（4）定义“reverse”的函数，实现对链表结点的反转。

目前，已经编辑完成“createNode”函数、“createList”函数和“main”函数，

设计思想：

这道题用到了链表，其实也就是结构体里包裹数据和指向下一个数据的指针。

实验代码、验证与结论：



总结与心得体会：

老师把代码的注释写的非常详细，基本上可以理解，不过很多地方还需要再看一看。

实验八：文件

实验题目1：写入文本文件

从键盘输入一个长度不超过100个字符的字符串，然后做如下操作：

（1）将字串中的小写字母转为大写，大写字母转为小写，而其它字符不作处理。

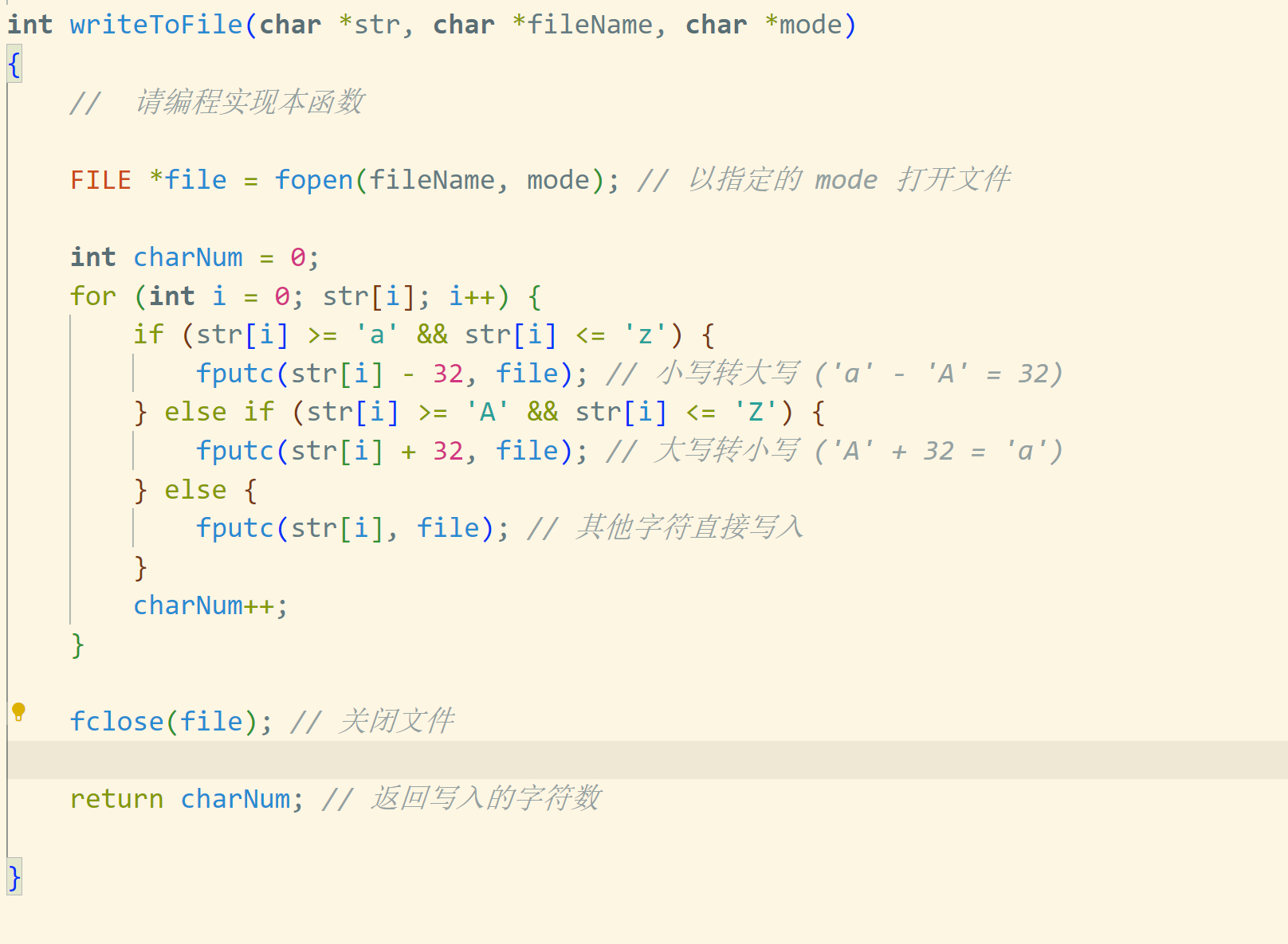
（2）将字串输出保存到一个名为“ex801.txt”的文本文件中。注：文本文件ex801.txt应与源码文件ex801.c保存在同一个文件夹中。

设计思想：

这道题需要注意的是大小写转换，要遍历字符串然后依次判断。

注意文件一定要关闭。

实验代码、验证与结论：





总结与心得体会：

第一次用 C 语言尝试了文件读写，如果以后用 C 语言做个小小的游戏的话，可以尝试用文件来存档保存游戏进度。

实验题目2：统计文本文件

请编程实现读取并统计文本文件“ex802.txt”中字母、数字和其它字符的个数，然后将统计结果输出到屏幕。注：该文本文件与源码文件保存在同一个文件夹中。

设计思想：

这里读到文件末尾会返回EOF，可以用来判断结束条件。

只要不到末尾，就可以一直计数。

实验代码、验证与结论：





总结与心得体会：

报告写了 7 个小时了，终于要写完了！

这道题用选择和循环结构就完成了。

实验题目3：格式化写文件

创建一个学生结构体，如下所示。

struct stu{

int id; //学号

char name[20]; //姓名

int age; //年龄

int score; //成绩

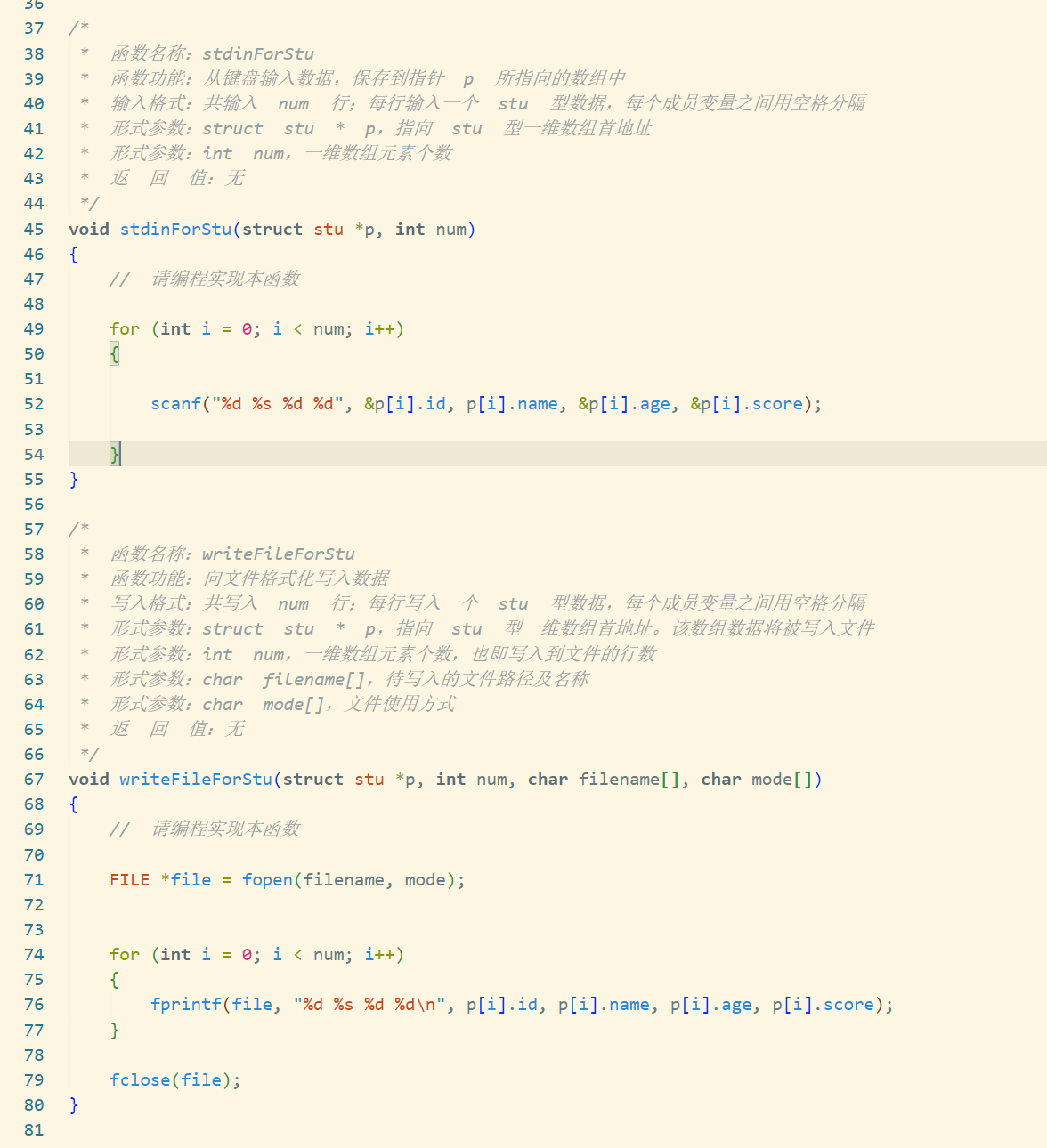
};

然后从键盘录入两名学生的信息，最后将上述信息格式化写入到文件。写入格式要求：将每个同学的信息写入到文件中的一行，每行中的各个数据之间以空格分隔。目前，已经完成了main函数的编写，请编程实现“stdinForStu”函数和“writeFileForStu”函数。

设计思想：

一个从键盘上接收传入的字符，一个写入到文件。

实验代码、验证与结论：





总结与心得体会：

从键盘上输入消息写到文件，正好用到了 scanf 函数将之前的知识点联系起来了。

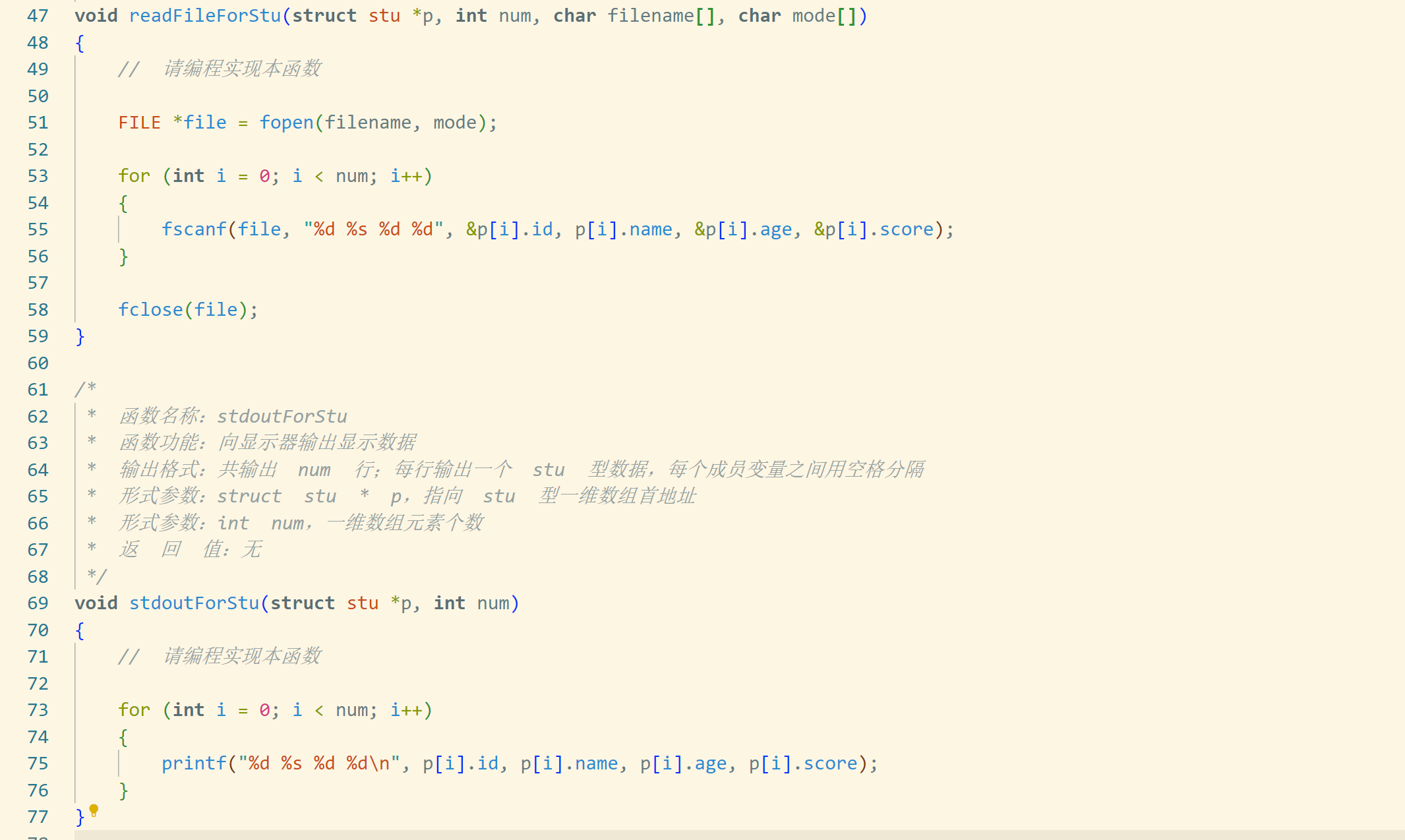
实验题目4：格式化读文件

从文本文件“ex804.txt”格式化读取学生信息，然后将学生信息封装到结构体数组中，最后在标准输出设备中输出显示。

设计思想：

就是打开文件、输入信息、然后输出。

实验代码、验证与结论：

**



总结与心得体会：

这次又用到了结构体数组来完成，再一次熟练。

实验题目5：读写二进制文件

现有一个结构体用来描述学生信息，结构体数据成员包括：学号、姓名、三门课程成绩和平均成绩。

typedef struct student{

char sno[SNO\_LENGTH]; //学号

char name[NAME\_LENGTH]; //姓名

int score[COURSE\_NUMBER]; //课程成绩数组

float average; //所有课程的平均成绩

}STU;

（1）从键盘输入3个学生的学号、姓名和三门课成绩，然后计算每个学生的平均成绩。之后，将每个学生的结构体数据存放在二进制磁盘文件“ex805.dat”（与源文件“ex805.c”存放于同一个文件夹）中。

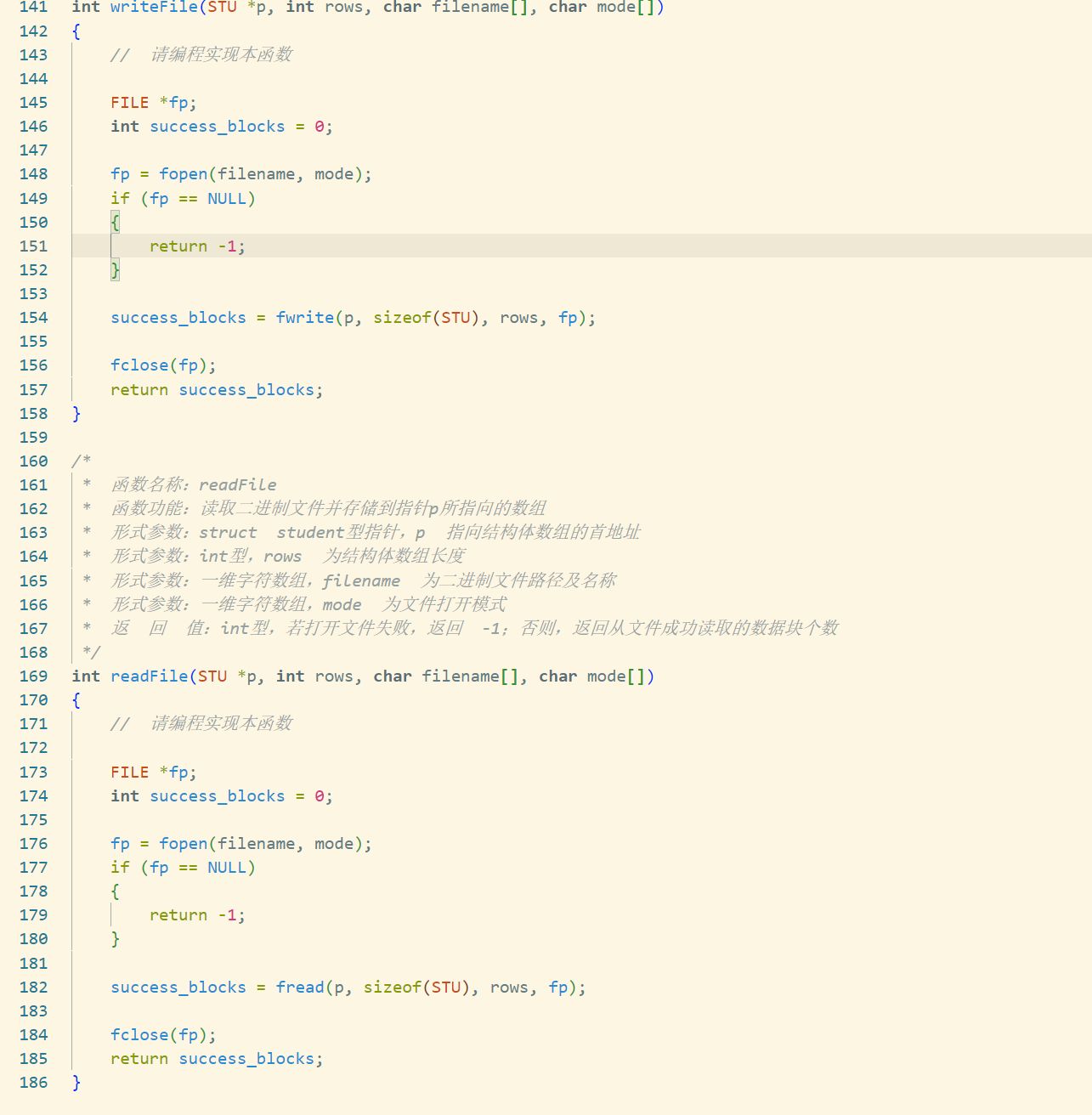
（2）读取二进制磁盘文件“ex805.dat”，然后输出到显示器。

目前，已编辑完成main函数、input函数和output函数。

设计思想：

与前面的相比，这次要注意二进制的格式。

实验代码、验证与结论：





总结与心得体会：

终于完成了！！！