

9.3

开始自学 C 语言的基本内容。用网课学习了 main 函数, printf,scanf 等基础内容。

9.4, 9.5

开始进行 vscode 的配置, 搜了一些教程, 稍微费了一点时间, 不过最终完成配置。

9.6 参加了 dian 团队的宣讲, 觉得蛮有意思的。

9.7 完成了 github 的注册, 建立了第一个库。

9.8 开始着手做 level-0 的第一问。

期间出现了一些问题。第一个问题是我不知道字符串应该如何进行比较。通过网络我知道了要用 strcmp 函数。第二个问题, 我忘记了 while 的循环应该怎么写。为此我在 B 站上搜了课, 学习了这方面的内容。

9.9 9.10 开始着手写代码, 首先定义了一个字符数组, 用于盛放输入值。然后构建了一个无限循环体。在这个循环体中, 首先用 scanf 读取输入字符并存储到 input 数组中。之后开始构建后续的循环步骤。主要分为 3 种互斥的情况。通过学习, 我了解到 if,else 语句适用于解决这类问题。总共分为三类。首先比较是否为 Dian, 然后比较是否为 Quit,最后如果都不是输出 Error。但是试运行并没有成功。调试提示;和花括号的使用存在问题。分号的问题解决的较快, 但花括号问题却花了很多时间。因为我之前没有搞明白花括号的具体用法。后来了解到几乎所有的语句都要用花括号, 于是进行了修正, 进过数次尝试最终运行成功, 成就感满满。

9.11

开始审第二问的题目。一开始理解有点困难, 之后大致搞懂了题意。但是对于如何引入文件一脸懵。于是借室友的 C 语言读本学习了有关打开文件, 解析文件的有关内容。并且了解了一些其他知识点。由于题干中对字符长度有要求, 要定义一些常量, 因此进行了宏定义, 由于题干中有键和值两个变量, 因此引入了一个结构体用于解决该问题。在完成基本定义之后。开始着手进行变量声明。首先需要定义一个字符数组用于存储文件。之后还要定义一个数组存储用户输入。还要定义一个 Pair 的类型用于存储读取的键值对。另外还需要定义一个整数变量用于技术已经读取的键值对数量。

9.12

在开始写代码之前, 昨天和同学交流时他们提到了指针, 指针这个东西我不太了解, 于是我用百度大致了解了一下指针的内容。明白了指针是一类指向内存地址的变量。之后开始写代码。在完成基本的定义之后, 开始进行第一步。读取用户提供的文件。用一个 printf 和 scanf 函数要求用户输入。获取文件名之后, 开始打开文件, 采用了 FILE*的文件指针, 指向文件流, 利用 fopen 形式打开, 并且以只读形式打开。为了处理错误情况, 还加入了对于无法打开文件的处理方式。

之后开始设计程序读取文件内容。这时候需要再次定义一个变量用于存储文件中每一行的内容。完成定义之后，开始设计一个 while 循环，逐行读取文件。利用之前学习过的文件读取内容，采用 fgets 函数从之前标记的文件流里逐行读取文件，用 sizeof 限定变量大小，并用之前定义的 count 函数限制最大键值对数量。完成对文件的解析过程。

但考虑到题干中还有关于空行的要求。为了是程序更加精准，处理一下空行的问题。这里我不太清楚究竟应该怎么处理，于是问 deepseek 如何处理空行问题。知道了只要保证 line[0]==空字符或者换行符就可以识别空行。而利用 continue 函数即可跳过这一行，处理下一行。

9.13 在完成文档读取问题之后。开始处理键值对。键和值之间有冒号，因此首先应该找到冒号。采用了 strchr 函数来查找冒号字符。从 line 中查找第一个匹配*colon 的指针。如果找不到，仍用之前定义的 continue 解决。

找到冒号之后我一时不知道怎么办了。下面的基本思路应该是把键和值分开。我查询了有关资料，只要把键和值只加上一个空格，再利用相关函数复制后，就可以形成键和值的割裂。（因为 strncpy 函数遇到空字符会自动停止）于是，我把*colon 指针换成了空字符。并且按照之前查询到的资料，利用 strncpy 函数进行复制。把冒号之前的东西复制到变量 pairs[count].key 中，并且留了一个位置给空字符。之前的资料提醒，为了保证程序正确终止还要加一行

```
pairs[count].key[KEY_SIZE-1] = '\0';
```

这个我不太理解，不过还是照做了。

之后如法炮制复制值的内容。复制起点应该在存储库的*colon+1 的位置，于是顺利完成值的复制。为了保证正确终止，也采用了相同的处理。

9.14

由于复制的值中可能有换行符。因此处理了一下换行符。引入了一个指针*n1 并且把这个指针换成了空字符。

那么程序的解析已经接近尾声。最后让 count 函数 count++，并且 fclose 文件。今天有一个小插曲，我们班参观了 dian 团队，观看了机器人表演，我拍了视频，还分享到了抖音上，引起了不少人评论，其中也有 Dian 团队的学长学姐，也算是提前和 Dian 团队结了缘。

9.15 完成了大体内容之后，开始处理用户端操作。首先让用户输入一个键。先确定不是 Quit，之后引入一个整形变量找到标志，并且初始化 found = 0。之后对之前存储的键值对进行遍历（遍历的方法查询了相关资料，没有理解的太透彻）。之后，利用 strcmp 函数比较键和用户输入。如果相等，就把对应的值输出并且设置找到标志。且终止一次循环。这就完成了一次键值对查找。

但是用户也有可能遇到找不到键值对的情况。鉴于此，如果! found,那么返回

error，并且提醒用户再次输入。

最后 return 0，程序结束。

历时五天利用军训闲暇时间完成了这道题目，有许多思考，体味了解决问题的基本流程。也查阅了许多资料。认识了许多新函数，认识了结构体，初步了解了程序的内存以及指针。并且对于写程序严谨化，全面化的思维有了新认识。

9.16 开始读 level1 内容。这个难度就比较大了。第一步要求我们要实现用户的登录。。用户名为全部大写字母，这个难度不算大。限定一下就行。另外，这个图书馆有层，每层还有列有行，另外还有时间这一维度。这点似乎就比较难了。

另外我还考虑了一下数据的来源。按理来说，图书馆应该有数据库，如果这是一个新的图书馆，我们也需要一个文件去储存已近获得的数据，这个虽然在第一步用不到，但是为了接下来的步骤，似乎也要构建一个数据分析储存系统。

但是当着手去写时，却发现自己的知识储备已经很难去写这个代码了，于是使用了 AI 工具帮我写了一个下水程序。

这个程序给了我两点启发。

第一，在全局函数构建上可以采用四维数组，处理以上的四个维度。这个和上道题似乎略有类似。

第二，之前考虑到的数据库的问题，凭我现在的水平很难达到要求。要求对数据加载，处理和保存的要求较高，这是之前没有接触到的。对数据的处理才是这个程序真正核心的步骤。主要分为数据初始化，数据加载，数据保存这三部分。处理用户登录不难，难的是如何保存这些数据。

9.17 开始尝试自己写代码，虽然大概率无法独立完成，需要借助 AI 来填补我的知识漏洞。但我还是尝试了一下。尝试独立写了函数头，宏定义等内容。之后尝试写了一下结构体，将 seats 采用四维数组进行构建。并将 seats, users, user-count 三个涉及的主要变量进了结构体构建。为了方便全局调用，采用了 AI 写的那个下水程序，声明了两个全局变量。

之后开始写 main 函数，显示输入指令的信息。

之后，利用循环结构处理用户输入。定义一个叫 command 的数组。用于存储用户输入。

之后利用一个无限循环结构处理用户输入。首先识别是否是 Login，如果是，输入用户名。如果正确输入了用户名 就登录成功，如果没有，就提示用户名无效。

之后，采用相同思路处理了 Exit，和 Quit 指令。在处理前者时，按照 AI 建议设计了一个 current-user 是否为 0 的判定，便于处理登录与未登录用户的不同情况。

9.18

开始写数据处理部分。

这一部分基本上都是参照 AI 写的，因为我对数据怎么处理实在知之甚少，只能尽量去理解 AI 所写的了。

首先是对数据进行一个初始化。利用一系列嵌套的 for 函数，遍历整个四维数组。将所有变量初始化为 0，并且让 seats 整个数组为 0。

并且对用户的账户进行初始化，并且规定用户的账户初始值为 A 最终只能加到 Z，有效回应了题干中要求的用户名只为大写字母。

之后进行数据的加载。以只读方式打开可能存在的图书馆文件，如果文件不存在，那么使用默认的初始化数据。

如果文件可以打开，那么遍历四维数组。从文件中利用 fscanf 函数读取座位的四维状态。完成读取之后关闭文件。

下一步是保存数据，同样打开文件，如果可以打开，仍采用遍历数组的形式。利用 fprintf 将四维数组的数据打印到文件当中，实现数据的保存。之后关闭文件。

说实在的，看完 AI 写的之后，确实感觉思路清奇，尤为精妙。但是对于四维数组遍历这块还是有点懵，知其然而不知其所以然。查了一些资料任然不甚了了。

9.20 开始进行最后的完善。

由于之前没有对 user name 这个变量进行处理，处理用户的登录。将 username 赋值给 current-user。完成登录过程。之后进行了退出登录。

然后对用户名进行有效性检查。程序考虑到还有 Admin，所以设置对于 Admin 的考虑，之后利用 ctype.h 下的 isalpha 和 isupper 两个函数限定 user name 是否为大写字母。至此，程序基本完整。

这个程序写完之后，我对于数据处理的理解上了一个台阶。也了解到了一系列的新韩淑，对于如何初始化数据，如何处理数据，如何调用数据有了一个体系化的理解。虽然还有一些盲点。

9.21 开始审 level1-2 的题

大致是将原本的变量进行一个可视化，转为数据，从而实现用户的预约。本质上还是对数据库的一个进一步处理。

这个让我不知所措，于是让 AI 加速了一下进程。

首先进行了数据化的处理，将日期字符串转化为了索引，这里面涉及到了一个映射的思想。将原本周日到周六的日期字符映射为了 0-6 七个数字。并处理了无效日期。

之后进行某一天某一层的座位状态显示，将某一天和某一层进行了协同处理。然后限定了楼层范围，之后用一个叫做 seats 的四维数组进行了对 status 变量的一个界定。

可以正确显示某一天某一层的状态。

9.22

之后开始处理管理员和用户的可视化权限问题。普通用户只能看到 0-2，而管理员可以看到具体哪些用户。

9.23 开始进行座位预约系统的编写。处理了一些错误的情况。定义了 floor-index 等一系列内容。如果 library.seats 的数据不等于 0，则这个数据可以预约。通过一系列赋值，存储用户的编号，从而实现预约。在成功预约之后，还要正确显示预约的内容。设置用户的 id，之后仍然以遍历数组的形式打印出每个用户的预约内容。

9.24, 9.25

对这个程序进行再理解。对于一些细节的内容进行了学习。整个程序分为四个系统，首先进行日期处理，之后通过 4*4 的格式显示实现座位的展示。还根据实际情况进行了用户与管理员分开处理的情况。之后构建了一个预约管理系统，为用户设置了 id，有效实现了先到先得的预约逻辑。最后，还有效地进行了查询反馈。

如果要我设计的话，我觉得座位和楼层还可以进行细分，不同楼层的座位可能会不同，还应该设计静音间什么的，这或许会让预约系统更加全面。