

**实 验 报 告**

|  |  |  |
| --- | --- | --- |
| **课程名称** | **:** | **c语言程序设计** |
| **专业班级** | **:** | **软工一班** |
| **姓名** | **:** | **郑航** |
| **学号** | **:** | **212406348** |
| **指导老师** | **:** | **陈欣** |
| **日期** | **:** | **2024/12/14** |

**福州大学至诚学院**

## 综合实验：图书馆管理系统

### 一、实验目的与要求

1. 掌握C语言的基本语法。
2. 掌握链表、数组、循环、选择、字符串结构、结构体的知识

### 二、实验准备与环境

1. Windows，Office Word
2. C语言编译器

### 三、实验内容

1. 任务：图书馆管理系统的开发
2. 主要功能：注册、登录、借书、还书、添书、删书、查询等操作。
3. 阶段步骤

### 四、实验方法与步骤

步骤与截图：

实验是按照什么步骤进行的，每一个功能的实现要求截屏放在下方，并写出通过什么方式实现的（比如一些基础操作，增删改查）

/\* 检查账密(pure) \*/

int check(const char\* user\_name, const char\* user\_password) // const的目的是提醒变量只读不写

/\* 注册账密入库(pure) \*/

int user\_in(char\* user\_name, char\* password)

/\* 图书基本结构(pure) \*/

typedef struct book // 定义图书结构体 // 用struct不用typedef，香味少一半呀,新类型和老类型都可以用哦，我都写一样就不会错了哈哈

{

int id; // 图书ID

char title[100]; // 图书标题

char author[100]; // 作者

int year; // 出版年份

struct book\* next; // 指向下一个图书的指针

} book;

/\* 记录当前用户名(pure) \*/

void temp\_name(char\* name)

/\* 数据进链表(pure) \*/

void data\_intolink(book\* temp)

/\* 创建链表 \*/

book\* creatlink() // 返回结构体指针类型的函数

/\* 打印链表 \*/

void printlink(book\* head)

/\* 删除admin库的书 \*/

void book\_out(int out\_id)

/\* 删除当前用户库的书,同时入admin库 \*/

void check\_outin(int out\_id)

/\* 查看admin库的书 \*/

void print\_admin()

/\* 按名字入库 \*/

void book\_in(book\* head) // 对于admin入库就是进书库了；对于user入库就是被他们借走的记录，避免乱还

/\* 用户和管理员的二层选择实现 \*/

void core\_choose(int role)

/\* 用户和管理员的二层页 \*/

void all\_core(char role)

/\* 登录界面 \*/

void login()

/\* 注册界面 \*/

void signup()

/\* 超级登录 \*/

void admin()

/\* 首页选择实现(选择放在对应的上面) \*/

void welcome\_choose(char choose)

/\* 首页 \*/

char welcome() // 欢迎界面

/\* 梦开始的地方 \*/

void start()

### 五、实验总结

通过这次实验收获了什么，有什么注意点等都可以写入总结。

还是模块多写几个吧，这样子可以相对独立，因为省了一些内容让代码维护难度提升太多

1链表应该适合一次性大量数据吧

2多次数据直接用用文件也好

3文件要及时关闭

4（我不维护随意文件打开失败基本不考虑）

5 打印上面最后改就是了，这种比较好改，嵌套多，同时还有传参就很难改了，所以我基本没有把会变化的指针作为函数的参数。

6最后修界面的时候虽然不是很快，但是很愉悦很轻松呀！

7 加了一些//system("cls");会好看一，不些过有点麻烦

8 最后调的时候发现模块有问题，最后发现是把字符串直接进行比较了晕了，不过用适当加入print函数可以帮助顺藤摸瓜接近bug

9对了,我根据要求就没做防用户乱借（只能检测id是否正确，书籍内容是错的就乱了），只做了防乱还,再做防乱还就要重推掉两个模块，个人感觉我写的有点屎山，不大好改

**注意：**

**一．建议先构思（很重要哦！）再写。**

**二．要达到下面阶段的要求，如何实现可以运用自己学会的知识，但必须包含上方实验要求的全部内容。**

**三．实验要求独立完成（每个人的代码风格都不一样，网络上的代码一看便知，不要投机取巧哦！），实验报告第四部分是考察是否独立完成和程序完成度的一个重点，越详细越好。**

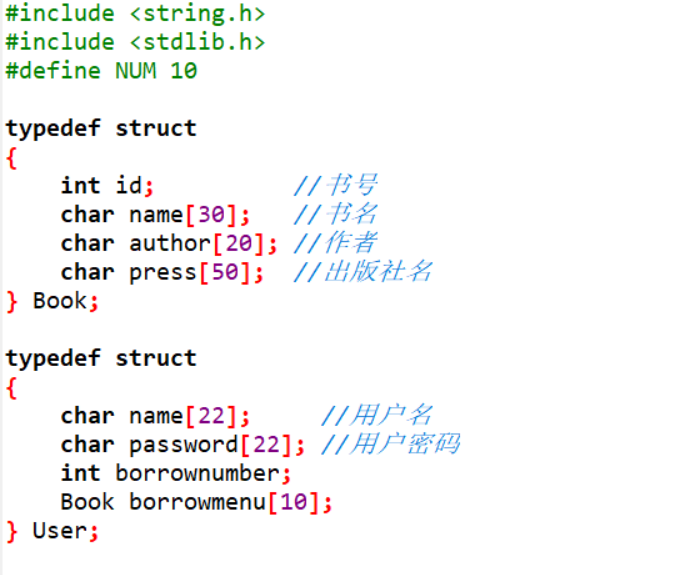
**四．在写的过程中难免有BUG，要学会耐心的去找BUG，修改BUG，这个过程对自己编程能力的提升帮助很大！！！**

**五、尽量使用链表和文件，正确使用有加分！！！**

提示：

程序的逻辑可以参考一下平时去图书馆借书的流程。比如：用户要有账号密码，然后才能进行属于用户的功能。

**提示一：可以用结构体**

****

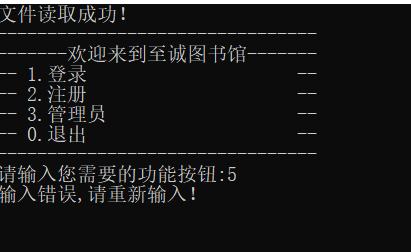
**提示二：方便将用户和自己借的书对应起来，可以用结构体数组（参考提示一图）**

**提示三：所有的退出应该是返回上一个页面，比如用户界面退出应该回到封面，继续运行**

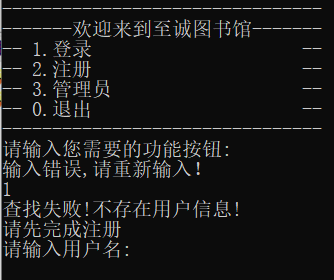
**提示四：想完成可循环运行，在main里面套一个while**

**提示五：实验报告在这个文档上面将自己的信息填写完整，然后把第四部分和第五部分补充完整，重要的事情多读三次！！！**

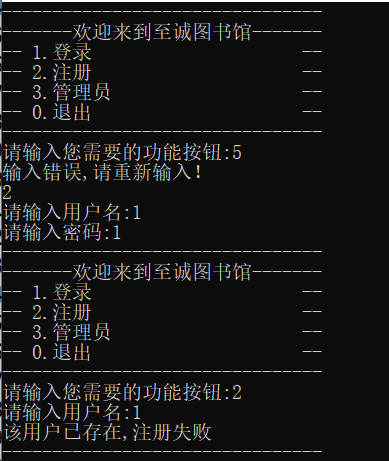
**阶段1：进行输入错误处理，整个系统要求能够可循环运行！**

****

**阶段2：没有账号进入登录，要想办法解决，不能直接结束程序**

****

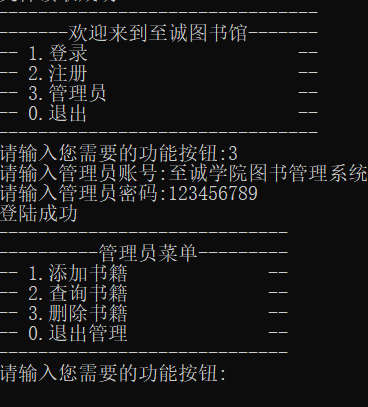
**阶段3：不能重复注册同一账号**

****

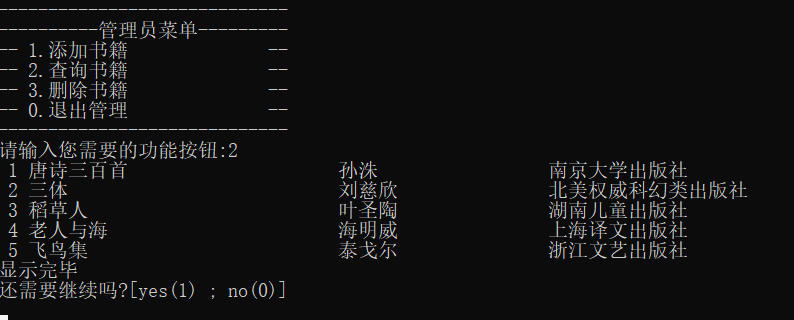
**阶段4：书写管理员系统，里面要求可以对书籍进行增删，以及遍历查询**

要求：

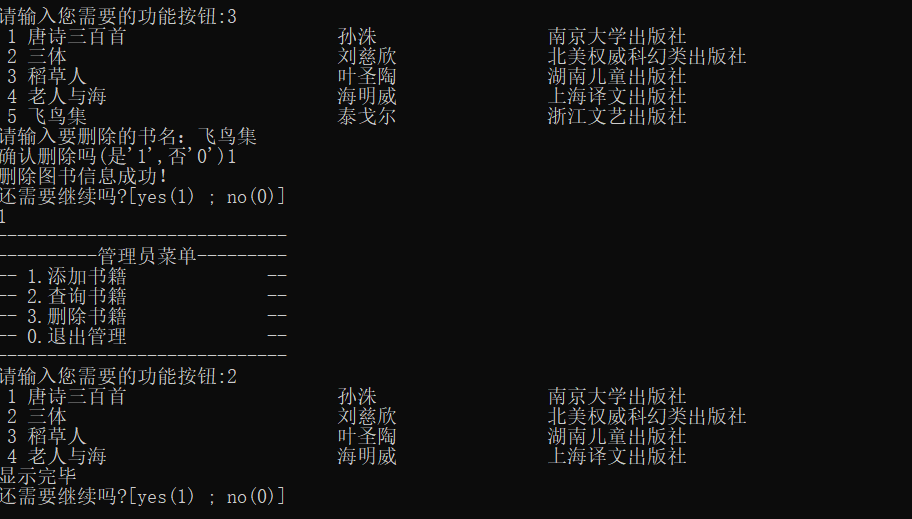
1. 查询 2. 删除 3. 添加

****

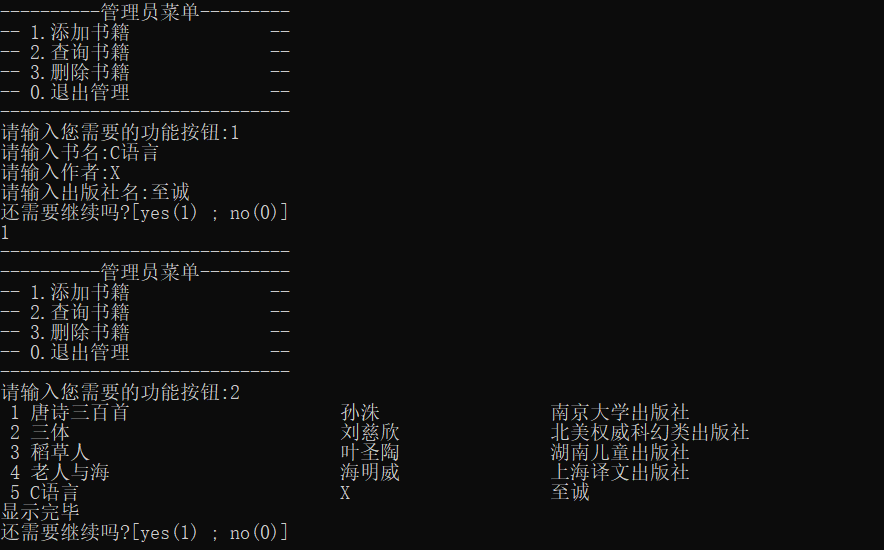
**（1）、查询**

****

**（2）、删除**

****

**（3）、添加**

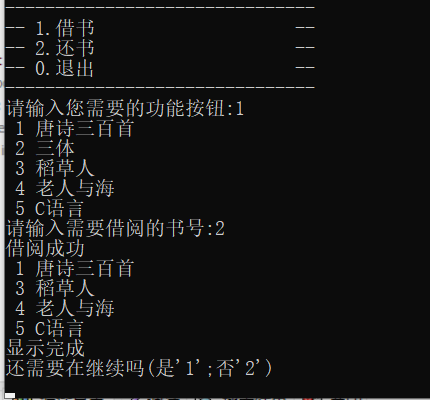
****

**阶段5：用户借书要求从书库里面删除，单独存储，不能还其他用户借的书，以及还没有借过的书**

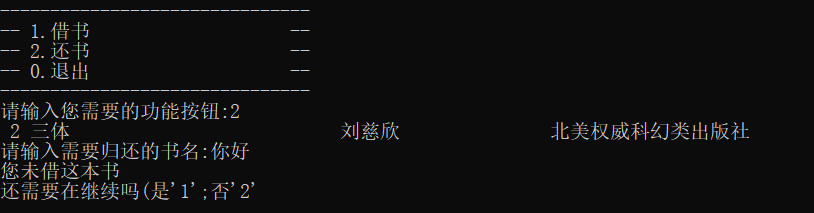
要求：

1. 从书库删除 2. 不能还没有借过的书 3. 每个人借的书独立存储

**（1）、从书库删除**

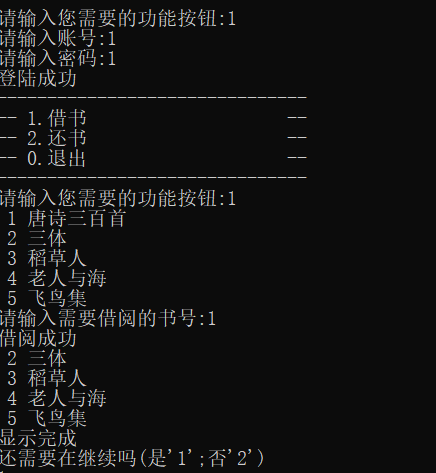
****

**（2）、不能还没有借过的书**

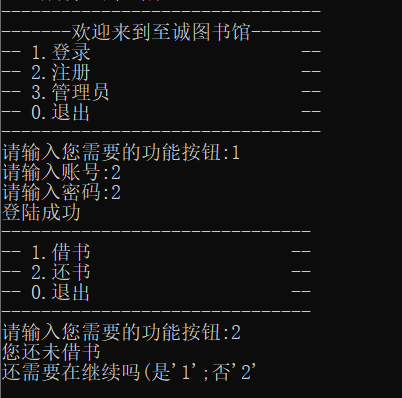
****

**（3）、每个人借的书独立存储**

例：账号一借书



账号二没有借书，所以不能还书



**阶段6：需要有使用说明（可以用记事本写）**

**例：**

