



**QG工作室中期考核**

考核名称 图书管理系统

分组类别 嵌入式组

学生学院 信息工程学院

专业班级 电子信息类18级(3 )班

学 号 311800XXXX

目录

[一、项目说明 1](#_Toc7162)

[1.1设计背景 1](#_Toc11856)

[1.2设计内容 1](#_Toc32716)

[二、设计目的与要求 4](#_Toc21073)

[三、整体设计 4](#_Toc9354)

[四、详情设计(基于链表) 6](#_Toc13429)

[4.1头文件模块 6](#_Toc1309)

[4.2函数模块 6](#_Toc5464)

[4.3结构体成员 6](#_Toc5059)

[4.4主函数界面代码 6](#_Toc10785)

[4.5主函数界面 11](#_Toc5465)

[4.6 book函数界面交互 12](#_Toc24361)

[五、结果分析 18](#_Toc23554)

[六、总结 18](#_Toc8329)

[七、附录 19](#_Toc1003)

# 一、项目说明

## 1.1设计背景

[图书](https://baike.baidu.com/item/%E5%9B%BE%E4%B9%A6" \t "https://baike.baidu.com/item/%E5%9B%BE%E4%B9%A6%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/_blank)管理系统，是一个由人、[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA/140338" \t "https://baike.baidu.com/item/%E5%9B%BE%E4%B9%A6%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/_blank)等组成的能进行管理信息的收集、传递、加工、保存、维护和使用的系统。利用信息控制企业的行为；帮助企业实现其规划目标。

图书馆管理系统，能进行图书馆管理系统能实测国民经济和企业的各种运行情况；利用过去的数据预测未来；从企业全局出发辅助企业进行管理决策；利用信息控制企业的行为；帮助企业实现其规划目标。

图书馆管理系统合运用了管理科学，系统科学，运筹学，统计学，计算机科学等学科的知识。可以通俗的简化的描述图书馆管理系统的三要素：系统的观点、数学的方法以及计算机的应用。

图书馆管理系统概念结构主要由四大部分组成即信息源、信息处理器、信息用户、信息管理者组成。

## 1.2设计内容

1. 选择执行身份（管理员和借书者）

2. 新进图书基本信息的输入。

3. 图书基本信息的查询。

4. 对撤消图书信息的删除。

5. 修改已有图书的信息。

6．登记借阅书。

7．登记时还书。

信息描述：有关该系统基本信息的描述，如：图书名称、图书编号、作者、存在状态、图书库存等

1. 课程设计的要求与数据
2. 进一步掌握和利用C语言进行程设计的能力；
3. 掌握数据结构的原理及初步运用

2、进一步理解和运用结构化程序设计的思想和方法；

3、初步掌握开发一个小型实用系统的基本方法；

4、学会调试一个较长程序的基本方法；

5、掌握实现界面的基本框架搭建和代码的规范性

三、课程设计应完成的工作

1、编写完成相应题目的程序；

2、编写课程设计报告,课程设计报告的内容应包括以下6个部分：

1) 需求分析：包括设计题目、设计要求以及系统功能需求分析；

2) 总体设计：包括系统总体设计框架和系统功能模块图；

3) 详细设计：包括主要功能模块的算法设计思路以及对应的工作流程图；

4) 调试分析过程描述：包括测试数据、测试输出结果，以及对程序调试过程中存在问题的思考（列出主要问题的出错现象、出错原因、解决方法及效果等，适当的包含结果截图）；

5) 总结：课程设计完成了哪些功能，有没有什么扩展功能？还有哪些地方需要改进？课程设计过程中的学习体会与收获、对本次课程设计的认识以及自己的建议等内容；



1图书管理系统

# 二、设计目的与要求

1. 建立图书管理并实现目的

2.图书管理基本业务活动包括：对一本书的采编入库、清除库存、借阅和归还等等。试设计一个图书管理系统，将上述业务活动借助于计算机系统完成。

[基本要求]

    （1）每种书的登记内容至少包括书号、书名、著者、现存量和总库存量等五项。

    （2）系统应实现的操作及其功能定义如下：

    ①采编入库：新购入一种书，经分类和确定书号之后登记到图书账目中去。如果这种书在帐中已有，则只将总库存量增加。

    ②清除库存：某种书已无保留价值，将它从图书账目中注销。

    ③借阅：如果一种书的现存量大于零，则借出一本，登记借阅者的图书证号和归还期限。

    ④归还：注销对借阅者的登记，改变该书的现存量。

⑤显示图书状态

1. 调试图书管理系统并修改解决方案

图书馆系统应以用户为中心，考虑好用户在运用时候可能需要的需求，处理好界面框架，处理好语句的执行对用户的说明，让用户在少操作的情况下让程序实现更多的功能，

功能要精悍，直接切入图书管理系统的主题，不符合主题的内容不附加，不造成用户的使用麻烦，不造成程序的无关复杂度，把空间复杂度高的程序用到用户理想的功能应用。

对参数进行改正，对于不同方案在执行，在过程中找出最佳方案，提出可行性并初步进行整改，重新规划方案，在有限时间花多心思去解决。

# 三、整体设计

**一、身份选择：1.管理者 2.借书人**

**管理者权利：**

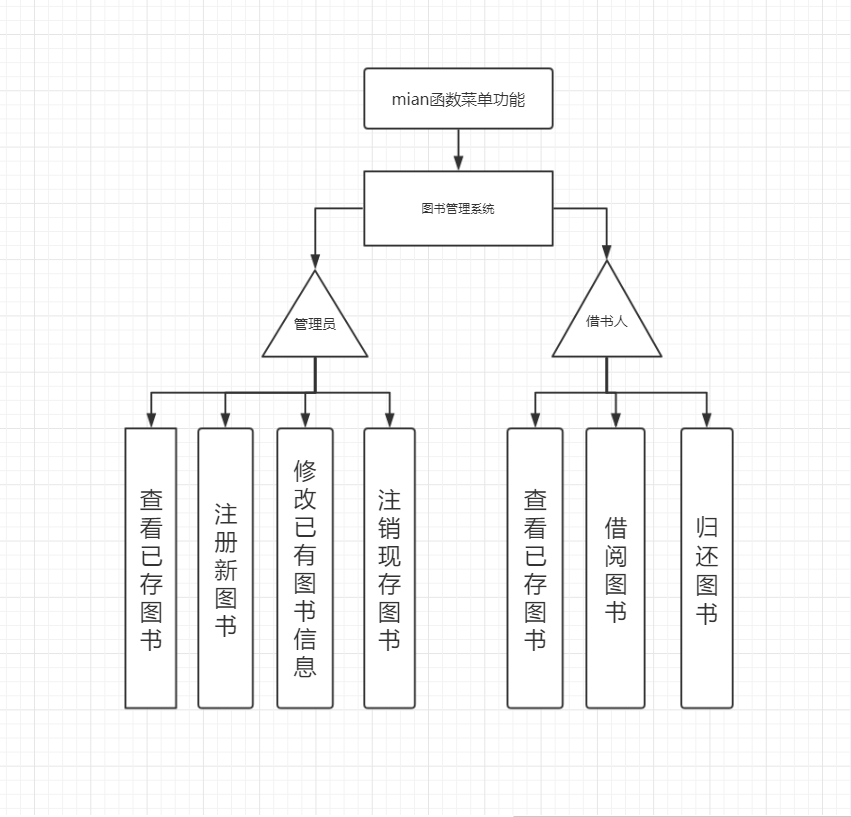
**1.查看已存图书 2.注册登记新图书 3.修改图书信息 4.注销（删除）现存图书**

**借书人权利：**

1. **查看已存图书 2.借阅图书 3.归还图书**

**二、增加退出选项和对应的输入检查**

**设计框架和功能图如下：**



2设计框架和功能图

# 四、详情设计(基于链表)

## 4.1头文件模块

#include <stdlib.h>

#include<stdio.h>

#include<string.h>

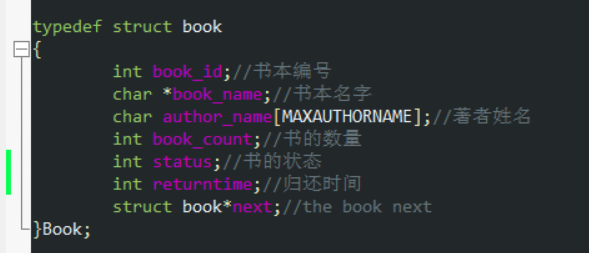
#include "book.h"

## 4.2函数模块

main.c

book.c

## 4.3结构体成员



结构体成员

## 4.4主函数界面代码

//身份选择界面初始化

int idnumber;

int id\_init()

{

int choice, confirm;

printf (

"\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\t\n"

"\t\t|【1】:\t管理员 |\t\n"

"\t\t|【2】:\t借书者 |\t\n"

"\t\t|【3】:\t退出 |\t\n"

"\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\t\n\n"

);

printf ( "\t\t请输入您的选择：" );

scanf ( " %d", &choice );

while ( ( confirm = getchar() ) != '\n' ) //用户再次输入空格键才表示其真正确认

continue;

//输入检测

while ( choice != 3 && ( choice < 1 || choice > 2 ) ) //判断执行1-2，并且检测合法输入

{

printf ( "\t\t您的输入: '%d' 有误!请重新输入：", choice );

scanf ( " %d", &choice );

}

return choice;

}

/\*管理者界面\*/

int usermanger()

{

int manger\_choice;

printf ( "\t\t-------------------------------------\n" );

printf ( "\t\t请输入项目前编号执行相关操作：\n\n" );

printf ( "\t\t【1】 查看已存图书\n" );

printf ( "\t\t【2】 注册新图书\n" );

printf ( "\t\t【3】 修改已有图书信息\n" );

printf ( "\t\t【4】 注销现存图书\n" );

printf ( "\t\t【0】 返回主页\n" );

printf ( "\t\t-------------------------------------\n" );

printf ( "\t\t输入您的选择:" );

scanf ( "%d", &manger\_choice );

while ( manger\_choice != 0 && ( manger\_choice < 1 || manger\_choice > 4 ) )

{

printf ( "\t\t您输入的：'%d' 有误！请重新输入：", manger\_choice );

scanf ( "%d", &manger\_choice );

}

return manger\_choice;

// switch ( manger\_choice )

// {

// case 0:

//

// return 0;

// case 1:

// printf("\t\t查看已存图书\n");

// break;

// case 2:

// printf("\t\t注册新图书\n");

// break;

// case 3:

// printf("\t\t修改已有的图书信息\n");

// break;

// case 4:

// printf("\t\t注销现存图书\n");

// break;

//

// }

}

/\*借书者界面\*/

int userborrower()

{

int borrower\_choice;

getchar();

printf ( "\t\t-------------------------------------\n" );

printf ( "\t\t请输入你要执行的操作\n" );

printf ( "\t\t【1】查看已存图书\n" );

printf ( "\t\t【2】借阅图书\n" );

printf ( "\t\t【3】归还图书\n" );

printf ( "\t\t【0】返回主页\n" );

printf ( "\t\t-------------------------------------\n" );

printf ( "\t\t请输入你的选择:" );

scanf ( "%d", &borrower\_choice );

while ( borrower\_choice != 0 && ( borrower\_choice < 1 || borrower\_choice > 3 ) )

{

printf ( "\t\t您输入的：'%d' 有误！请重新输入：", borrower\_choice );

scanf ( "%d", &borrower\_choice );

}

return borrower\_choice;

// switch ( borrower\_choice )

// {

// case 0:

// return 0;

// case 1:

// printf("\t\t查看已存图书\n");

// break;

// case 2:

// printf("\t\t查看已存图书\n");

// break;

// case 3:

// printf("\t\t查看已存图书\n");

// break;

//

// }

}

int main()

{

printf ( "\t\t\t\t\t\t\t\t『制作：宏 时间：2019年4月10日 名称：图书管理系统』\n\n" );

printf ( "\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*欢迎来到图书管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" );

int choice;

int borrower\_choice;

int manger\_choice;

//选择身份

while ( ( choice = id\_init() ) && choice != 3 )

{

system ( "cls" );

switch ( choice )

{

case 1:

printf ( "\t\t你是管理员\n" );

while ( ( manger\_choice = usermanger() ) && manger\_choice != 0 )

{

switch ( manger\_choice )

{

case 1:

//printf ( "\t\t查看已存图书\n" );

viewbook();

break;

case 2:

//printf ( "\t\t输入图书\n" );

addbook();

break;

case 3:

//printf ( "\t\t修改已有图书信息\n" );

modifybook();

break;

case 4:

//printf ( "\t\t注销现存图书\n" );

deletebook();

break;

}

}

//system ( "cls" );

break;

case 2:

printf ( "\t\t你是借书者\n" );

printf ( "\t\t请输入你的借书证号:" );

scanf ( "%d", &idnumber );

getchar();

while ( ( borrower\_choice = userborrower() ) && borrower\_choice != 0 )

{

switch ( borrower\_choice )

{

case 1:

//printf ( "\t\t查看已存图书\n" );

viewbook();

break;

case 2:

//printf ( "\t\t借阅图书\n" );

borrowbook();

break;

case 3:

//printf ( "\t\t归还图书\n" );

returnbook();

break;

}

}

idnumber = 0;

//system ( "cls" );

break;

}

}

printf ( "\t\t您已退出图书管理系统，感谢您的使用!\n" );

system ( "pause" );

}

## 4.5主函数界面

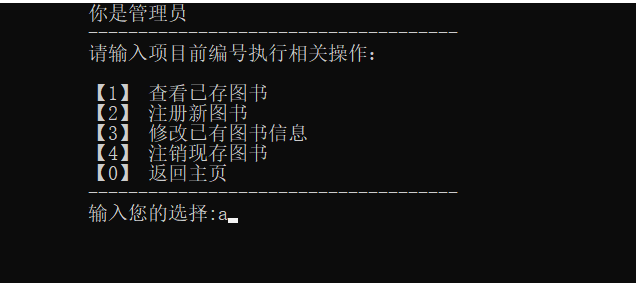
思路：选择身份，选择管理员或借书者，或者退出整个图书管理系统，并有错误输入判断



3主函数界面

1. 管理者操作界面

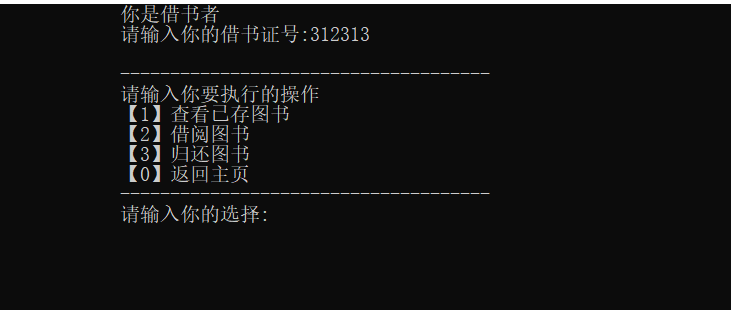
思路：选择1-4的功能执行，0返回主界面



4管理者操作界面

2.借书操作界面

思路：选择1-3功能执行，0返回主界面



5借书者操作界面

### 4.6 book函数界面交互

思路：通过十个函数交互实现功能

book.c封装10个执行函数，分别是：

int viewbook();查看图书信息

int addbook();增加注册图书

int deletebook();注销删除图书

int modifybook();修改图书信息

void printbook(Book \*bookpack);遍历图书链输出图书信息

Book \*lookupbookname ( char \*bookname, int id );检验信息

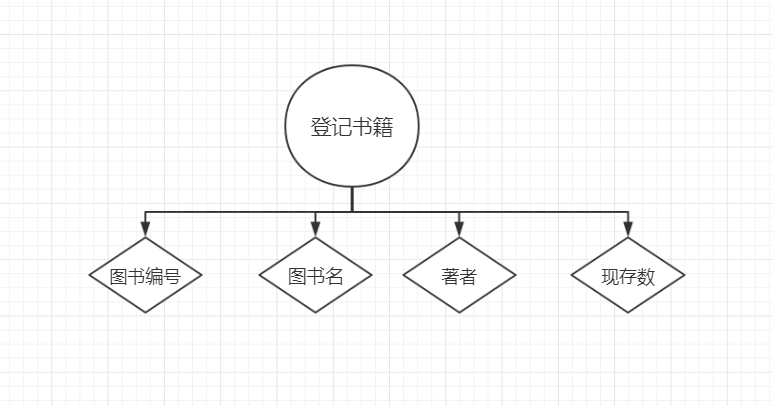
Book \*lookupauthorname ( char \*authorname, int id );检验信息

int borrowbook();借阅图书

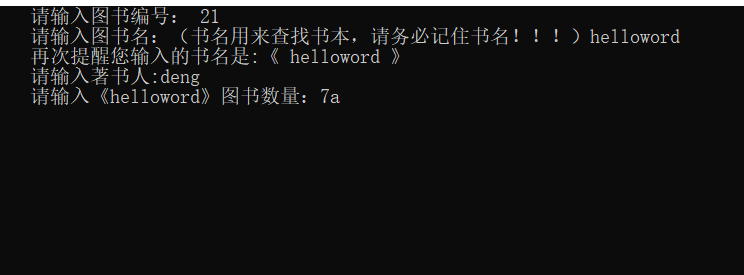
int returnbook();归还图书

**交互过程**

注册图书：思路：登记图书的编号、图书名、著者和图书数量



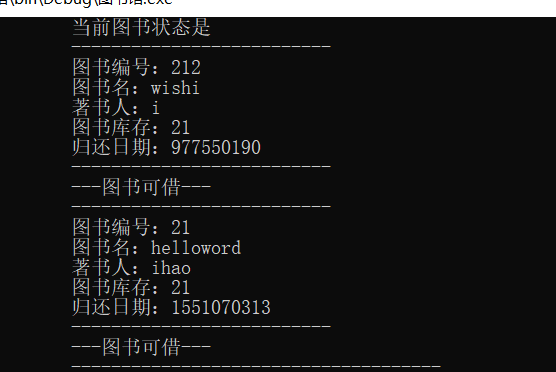
6注册图书流程



7注册图书流程界面

查看图书

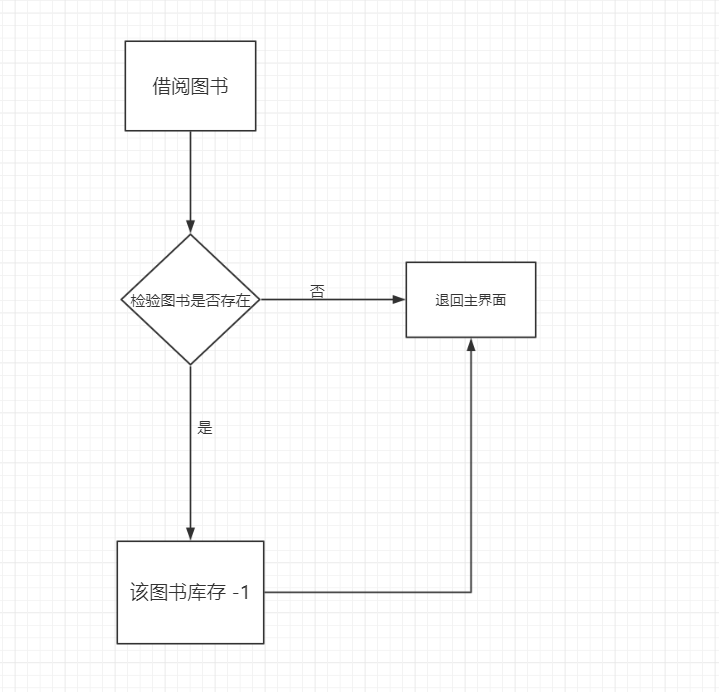
思路：把图书的全部信息输出到界面框



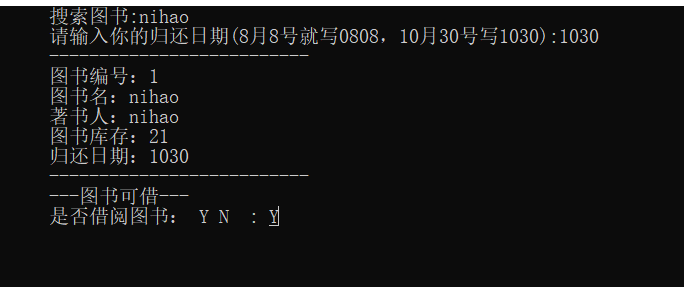
8查看图书

借阅图书

思路：通过查找书的名字来查找书，以实现借书



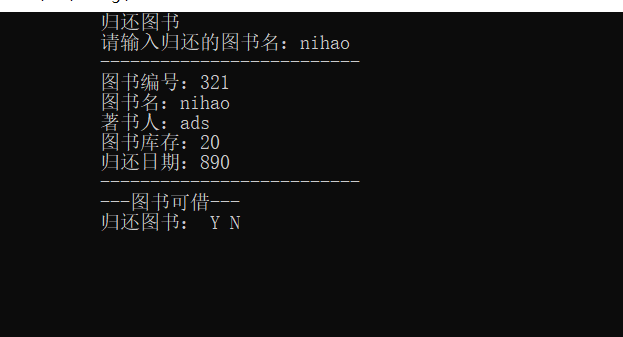
9借阅图书流程



10借阅图书流程

归还图书

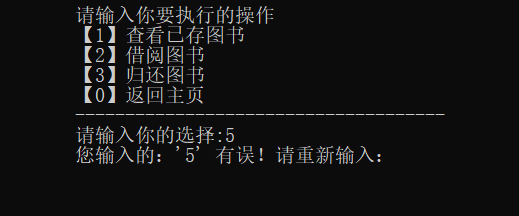
思路：通过查找书的名字来归还书籍



11归还图书

检查错误

思路：检查是否输入了其他非法字符和不对应的编号，错误则报出



12错误检查

# 五、结果分析

①很多主要函数如returnbook()；borrowbook()；在编写时出现很多错误，而对这些功能块的实现不很理解，导。

②传参和遍历检查的时候出现错误，让程序出现不少崩溃的情况，调试需要花费时间是巨大的。

③由于程序较长，在编译不过时虽然编译检查不出来语法错误，但程序的逻辑错误发生了，解决方法是在出错的地方进行分析，通过前后的联系，找出一些相关变量的值，看是否在我们的逻辑运算之内，这样就可以缩小范围，更容易找出错误的原因。

④一开始出现命名不规范导致检查代码十分困难，要对代码格式规范

# 六、总结

通过这一次的课程设计，我加深了对所学知识的理解，并学会了在时间极端的情况下耐心地寻找错误，对如何用所学知识解决实际问题有了一定的了解，体会到运用结构化的程序设计方法的优越性：在自顶而下、逐步细化的设计理念下理清思路，给出相应的数学模型解决当前问题将使程序更严谨合理，也将使操作更方便快捷.

代码规范。对于变量名称，长度并不是名称的价值所在，清晰的表达才是。不常用的全局变量可能会有一个很长的名称，像 maxphysaddr。在循环中每一行所使用的数组索引，并不需要取一个比 i 更详尽的名字。取 index 或者 elementnumber 会输入更多的字母（或调用文本编辑器），并且会遮盖住计算的细节。当变量名称很长时，很难明白发生了什么。

排版问题。程序是一种出版物。意味着程序员们会先阅读（也许是几天、几周或几年后的你自己阅读），最后才轮到机器。机器的快乐就是程序能编译，机器才不在乎程序写的有多么漂亮，可是人们应该保持程序的美观。有时人们会过度关心：用漂亮的打印机呆板地打印出漂亮的输出，而这些输出只是将所有介词用英文文本以粗体字体凸显出来，都是些与程序无关的细节。虽然有很多人认为程序就应该像 Algol.68 所描述的一样（有些系统甚至要求照搬该风格编写程序），可清晰的程序不会因为这样的呈现而变得更清晰，只会使糟糕的程序变得更可笑。

一个优秀的程序,不仅要让人一拿到就会用, 还应该具有人性化的界面，优秀的实现效率，并且在用户错误使用时能给予提示,拥有措施维护好用户的数据.

重视实践操作，这对学习任何一门语言来说都是不变的真理。我在这次的课程设计中深深地体会到这一点，认识到自己对程序的设计思路的组织以及对实际问题的解决方案的反映能力尚有待提高。C语言是门极具魅力的课程，这次的课程设计也提高了我对其的浓厚兴趣，在今后的学习中，我将更加重视实践操作，使自己对C语言的知识学习达到融会贯通。

# 七、附录

**参考文献：**

1.《c语言从入门到精通》

1. 百度文献，百度文库