**前言**

本次课程设计的内容共有三道大题。第一、二是为应用类题目，第三题为综合类题目，涉及循环、数组、函教、链表等主要知识点。本次选题为学院根据专业学生学号尾号数字相对应分配题目。本次小编的选题是学生籍贯信息管理，针对选题，实现三道大题，并进行综合设计与实现，此次的课程设计说明书主要分为封面，前言，目录，课程设计任务书和正文五大部分。 其中正文部分有六章主要内容和参考文献，附录。第一章为概述,其内容为包括课题设计的目的与要求，选题的内容、设计的过程,开发环境等。第二章为需求分析,主要内容为描述完成的功能，以及每一个功能所完成的任务。第三章为系统设计，其分为总体设计，详细设计和界面设计。总体设计是系统的功能结构图；详细设计是主要算法的流程图，以及确定程序中所用的主要数据的数据类型，选择的数据结构等。第四章为编码，列出程序中结构体类型的定义，列出程序中全局变量，程序中通用的符号常量。列出程序中自己设计的所有函数的原型，并说明功能，参数的意义。给出主要功能的代码并有适当的注释。第五章运行与调试，主要内容是运行、调试程序并截图，进行说明调试的目的与结果。初步学会软件测试。第六章为总结部分，其主要内容是对本次课程设计过程中的收获，体会以及不足或存在的问题进行总结。

由于学生的能力范围有限，本次课程设计说明书中还存在很多不理想的地方，也难免有不当和欠妥之处，敬请各位指导老师多提宝贵意见。

日期：2022年6月13日

**目录**

[**课程设计任务书** 4](#_Toc105173287)

[**第1章 概述** 7](#_Toc105173288)

[1.1课程设计的目的与要求 7](#_Toc105173289)

[1.2课程设计选题的内容 8](#_Toc105173290)

[1.3课程设计的内容以及开发环境 8](#_Toc105173291)

[**第2章 需求分析** 9](#_Toc105173292)

[2.1系统需要包含的功能 9](#_Toc105173293)

[2.2每个功能所要完成的任务 9](#_Toc105173294)

[**第3章 系统设计** 10](#_Toc105173295)

[3.1总体设计 10](#_Toc105173296)

[3.2详细设计 10](#_Toc105173297)

[**第4章 编码** 14](#_Toc105173298)

[4.1程序中结构体类型的定义 14](#_Toc105173299)

[4.2程序中全局变量、程序中通用的符号常量。 14](#_Toc105173300)

[4.3自己设计的所有函数的原型，并说明功能、参数的意义。 14](#_Toc105173301)

[4.4主要功能的代码与注释。 14](#_Toc105173302)

[**第5章 运行与调试** 21](#_Toc105173303)

[5.1打印菜单 21](#_Toc105173304)

[5.2增加学籍信息 21](#_Toc105173305)

[5.3修改学籍信息 22](#_Toc105173306)

[5.4查找学籍信息 22](#_Toc105173307)

[5.5删除学籍信息 23](#_Toc105173308)

[5.6打印学籍信息 23](#_Toc105173309)

[**第6章 总结** 24](#_Toc105173310)

[**参考文献** 25](#_Toc105173311)

[**附录：** 26](#_Toc105173312)

# 插图清单

[图3-1 系统总功能图 10](#_Toc105173472)

[图3-2 主函数流程图 11](#_Toc105173473)

[图3-3 增加学籍信息流程图 11](#_Toc105173474)

[图3-4 修改学籍信息流程图 12](#_Toc105173475)

[图3-5 查找学籍信息流程图 12](#_Toc105173476)

[图3-6 删除学籍信息流程图 13](#_Toc105173477)

[图3-7 打印学籍信息流程图 13](#_Toc105173478)

[图5-1 打印菜单 21](#_Toc105173479)

[图5-2 增加学籍信息 21](#_Toc105173480)

[图5-3 修改学籍信息 22](#_Toc105173481)

[图5-4 查找学籍信息 22](#_Toc105173482)

[图5-5 删除学籍信息 23](#_Toc105173483)

[图5-6 打印学籍信息 23](#_Toc105173484)

# 表格清单

[表4-1 主要函数功能 14](#_Toc105173509)

# 课程设计任务书

一、课程设计题目： 校运会项目管理系统

**二、设计任务和内容**

本课程设计的内容共有3道大题。第1、2题多为应用类题目，第3题为综合类题目，涉及循环、数组、函数、链表等主要知识点。针对选题，实现三道大题，并进行综合设计与实现。具体任务包括：

1、熟悉实训教材中的课程设计的目的、要求与过程。

2、掌握实训教材中的课程设计的内容。

3、针对选题进行需求分析，特别是功能分析，划出功能图。

4、算法分析与设计，写出每个模块的设计思想，画出每个算法的流程图或N-S图。

5、编写源程序。

6、静态走查程序和上机调试程序，写出调试报告。

7、综合程序。如果三道题目实现时，是以三个程序独立完成的，则将三道题目，组合成一个程序（以命令菜单的方式，供用户选择每题实现的功能），并进行调试。也可以在开始设计时就以一个程序实现，通过菜单的方式实现功能的选择。

8、书写上述文档和撰写课程设计说明书。

**三、设计步骤和要求**

**1、设计步骤**

首先进行相关资料查阅和学习、调查研究，了解基本的业务流程和系统功能、数据需求。然后结合软件工程的理论完成设计任务，主要包括需求分析、总体设计以及详细设计、编码与调试部分。

**2、课程设计说明书要求与格式**

**(1) 基本要求：**

①能反映完成了上述设计内容要求。

② 要求撰写不少于5000个文字（20页）的文档。

③ 文档中至少要包括：主要功能分析、系统功能结构图、算法流程图、系统运行与调试截图、总结、参考文献。

④ 课程设计说明书可以打印或手写，其中用户界面设计、程序运行与调试需要截图打印。

**(2) 文档格式要求**

① 封面

② 前言

③ 目录

④ 课程设计任务书

⑤ 正文（分章、层次等，每一章从新一页开始）**（每章内容参考如下：）**

**第1章 概述**

　 包括课题的课程设计的目的与要求、选题的内容、设计的过程、开发环境等内容。

**第2章 需求分析**

描述完成哪些功能，以及每一个功能所完成的任务。

**第3章 系统设计**

3.1总体设计

**画出系统功能结构图。**

． 3.2详细设计

**画出主要算法的流程图。**

确定程序中所用的主要数据的数据类型，选择所用的数据结构，如：结构体、数组、链表等。

列出程序中用到的所有函数的名称、功能，调用关系。

　 界面设计

**第4章 编码**

　　列出程序中结构体类型的定义。

列出程序中全局变量、程序中通用的符号常量。

列出自己设计的所有函数的原型，并说明功能、参数的意义。

给出主要功能的代码并有适当的注释。

**第5章 运行与调试**

运行、调试程序并截图，进行说明调试的目的与结果。初步学会软件测试。

**第6章 总结**

对本次课程设计过程中的收获、体会以及不足或存在的问题进行总结。

**参考文献**

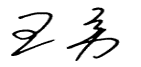
**附录：第4章没有列出的代码**

**(3) 装订要求**

课程设计报告必须按要求和顺序装订，封面的指导老师必须和课表中课程设计指导老师一致，不一定是理论课老师。

说明：该门课程的任务提前布置给学生，在学生放寒假前布置。

教师签名： C:\Users\YunyunYi\Desktop\无标题.png

审核人： 

日 期：2022年 6月 1日

# 第1章 概述

## 1.1课程设计的目的与要求

1.1.1课程设计的目的

(1)综合使用所学过的C语言程序设计知识，掌握结构化程序设计的基本思路和 方法，加深对C语言基本要素和控制结构的理解。

(2)通过查阅手册和文献资料，培养系统理解、综合运用c语言知识分析问题和利用计算机求解问题的能力。

(3)利用所学的基本知识和技能，开发小型应用程序，培养数据处理的综合能力。

(4)结合课程的教学内容循序渐进地进行设计方面的实践训练，以获得软件开发经验，并加强与其他相关课程的横向联系及其应用，为后续课程的学习和毕业设计奠定基础。

(5)及时巩固已学知识，对于给定的设计题目，考虑如何进行系统分析，建立相应的数学模型，从而提高自身的专业综合素质。

(6)熟练应用自顶而下的计算思维方法，将大问题模块化，领会结构化程序设计的 精髓；培养独立思考、阅读和编写程序的能力，积紧编程经验，打下良好的计算机应用基础。

(7)学习软件设计的思维方法。培养从全局考虑问题的模块化思维方法，培养设计 单个模块算法的逻辑化思维方法，培养产生多种设计方案的发散性思维方法。

(8)掌握程序的局部测试与系统测法的逻辑化思维方法，掌握在集成环境下如何设置 断点进行单步调试或分块调试的方法。

1.1.2课程设计的要求：

（1）利用单链表结构实现学生籍贯管理，深入了解数据库管理的基本功能，掌握c语言中的结构体、指针、函数（系统函数、自定义函数）、文件操作等知识。

（2）对系统进行功能模块、控制模块分析，可以增加其他功能修饰，使系统更加完善合理，并给出功能模块图。  
（3）采用模块化程序设计方法，程序尽量局部化和模块化。

（4）利用c语言面向过程的程序思维来完成系统的设计，突出c语言的函数特征，采用多个函数实现各个子功能。  
（5）程序流程图清晰，数据类型和数据结构定义详细。  
（5）算法简洁明了，少使用技巧，尽量避免使用多种循环嵌套或条件嵌套结构。

（6）充分利用库函数，系统设计实用，程序简练，输入/输出操作能给用户必要的提示，功能全面可用，能实现友好的交互。

（7）源程序采用锯齿形书写格式，注释详细采用较好的缩进格式。

## 1.2课程设计选题的内容

【题目】建立学生籍贯管理系统，对学生籍贯相关信息进行管理。学生的相关信息包括：学生学号、姓名、电话号码、身份证号码、籍贯、通讯地址等。具体要求：

（1）建立该存储结构。

（2）录入学生的相关信息。

（3）查找某个学生的籍贯。

（4）修改某个学生的籍贯。

（5）删除某个学生及相关信息。

（6）插入某个学生的相关信息。

【编程提示】将每个学生记录当作一个结点，类型为结构体，结构体中包含学生的相关信息和结点指针。定义联系人结点的结构为

struct node

{

char no[12]; /\*学号\*/

char name[40]; /\*姓名\*/

char tele[20]; /\*电话号码\*/

int D[18]; /\*身份证号码\*/

char jg[60]; /\*籍贯\*/

char addr[80]; /\*通讯地址\*/

struct node \*link; /\*指向下一结点的指针\*/

};

编写查找函数，首先输入要查找的学号或身份证号，然后顺序查找结点，如果找到则显示找到的记录，之后继续向下查找直至链表结尾。删除某个结点时，可调用查找函数，如果没找到则输出没找到信息；如果找到则分三种情况讨论：该结点为头结点时，修改原头结点所指向的结点为新的头结点；该结点为中间结点时，修改该结点的前一结点的指针使其指向该结点的后一结点；该结点为尾结点时，修改原尾结点的前一结点为新的尾结点；最后释放该结点。

## 1.3课程设计的内容以及开发环境

课程设计的内容主要包括5个步骤，即程序的功能分析，数据结构设计，算法分析和设计，函数设计编码及调试等各个步骤都有特定的任务。课程设计的开发环境是Visual Stdio 2022。

# 第2章 需求分析

## 2.1系统需要包含的功能

1. 建立信息的存储结构
2. 登陆输入密码的功能
3. 输入功能：进行基本数据信息的输入。
4. 存储功能：对文件的存储管理。
5. 查找功能：在面临比较大的数据时，便于用户快捷找到所需要的信息。
6. 修改功能：修改学生的籍贯信息。
7. 删除功能：删除因某些原因退出的学生籍贯信息。

## 2.2每个功能所要完成的任务

1. 使用单链表建立数据储存结构
2. 建立两个全局变量password,和passwd,用于存储正确密码和输入的密码，然后进行比对，判断是否正确，决定是否进入系统。
3. 首先输入要录入的信息个数，用于实现一次录入多个数据，然后依次输入学生的学号、姓名、电话号码、身份证号码、籍贯。
4. 所有的数据在完成一项功能后与都会自动保存在data.txt文件中，使数据更加安全。
5. 输入要查找的学生学号后，遍历单链表，如果找到该学号，则输出该学号学生的所有信息，否则输出找不到该学号。
6. 输入要修改的学号后，遍历单链表，如果找到该学号，依次输入修改后的学号、姓名、电话号码、身份证号码、籍贯，否则输出找不到该学号无法修改。
7. 按下对应的数字，输入要删除的学号以后，遍历单链表，找到后删除，否则输出找不到该数据，退出:>。

# 第3章 系统设计

## 3.1总体设计

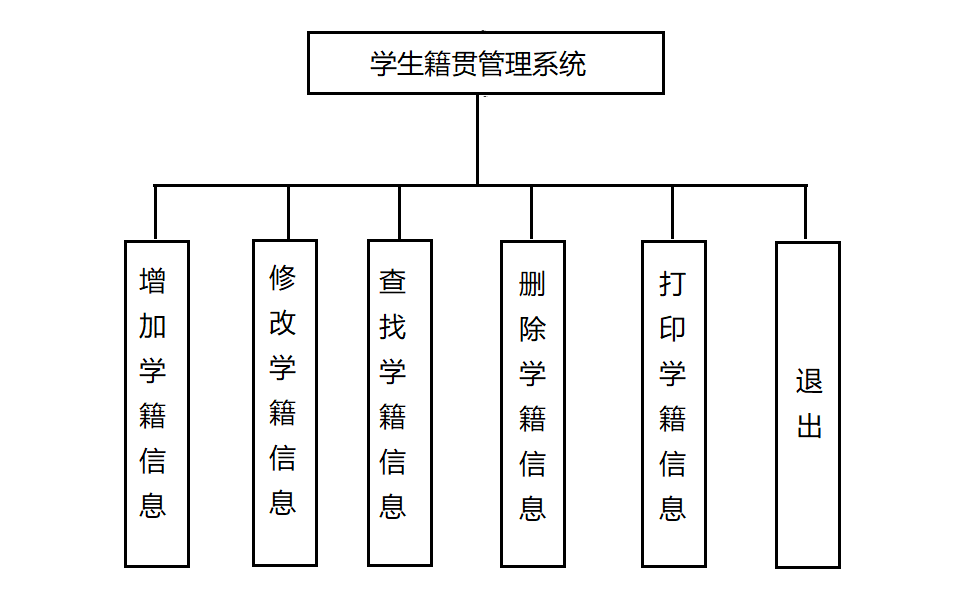


图3-1 系统总功能图

## 3.2详细设计

1. 程序中所用的主要数据的数据类型

程序中所用的主要数据类型为：单链表

1. 程序中用到的所有函数的名称、功能、与调用关系

本系统中使用的自定义函数有readfile函数，用于读文件，savefile函数用于保存文件，menu函数，用于打印菜单，add函数，用于录入学生学籍信息，revise函数用于修改学生学籍信息，find函数用于查找学生学籍信息，dele函数用于删除指定学生信息print函数用于打印所有的学生信息。以上函数均被主函数调用。Print函数打印指定结点信息，search函数查找指定结点信息，以上函数均被static修饰，被自定义函数调用。

1. 主要算法的流程图

1.主函数流程图

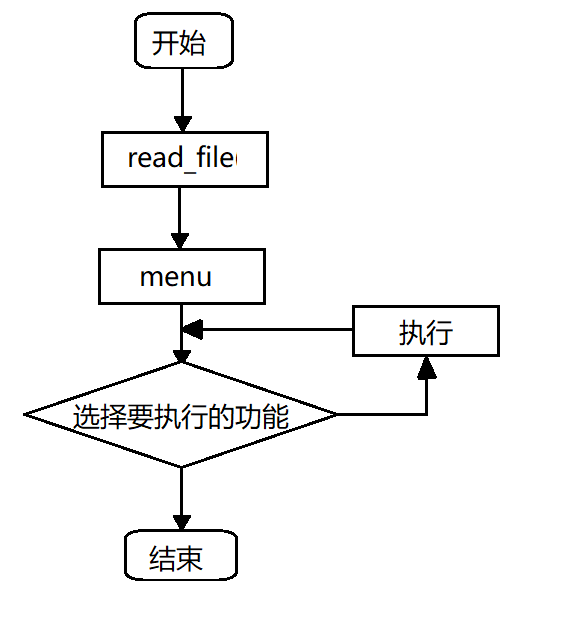


图3-2 主函数流程图

2.增加学籍信息流程图

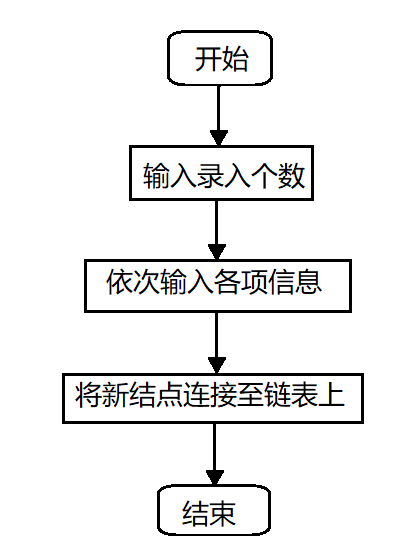


图3-3 增加学籍信息流程图

3.修改学籍信息流程图

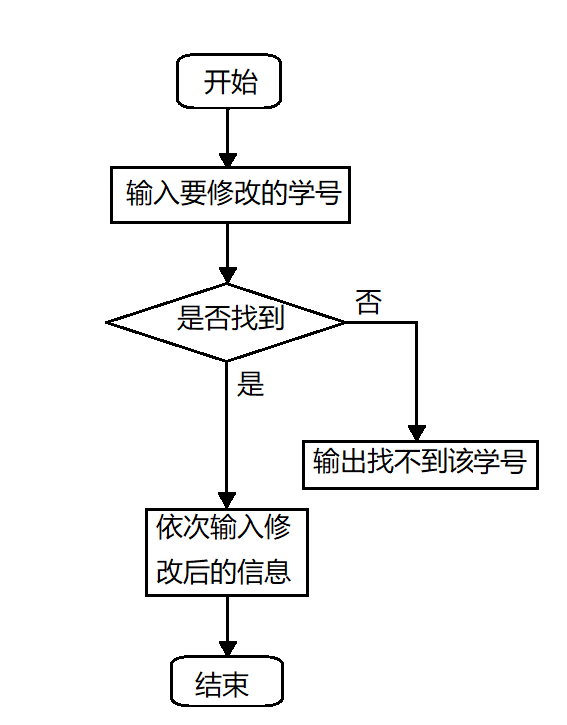


图3-4 修改学籍信息流程图

4.查找学籍信息流程图

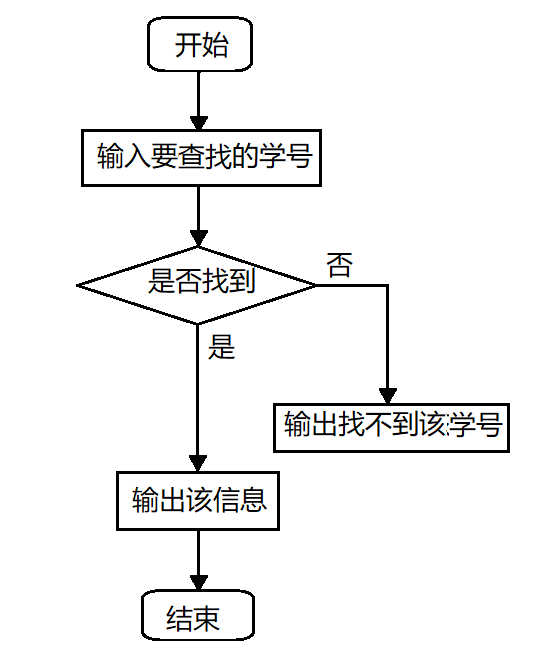


图3-5 查找学籍信息流程图

5.删除学籍信息流程图

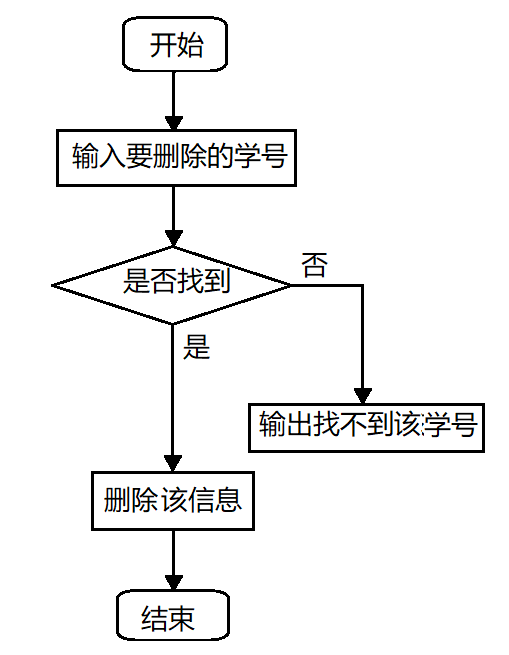


图3-6 删除学籍信息流程图

6.打印学籍信息流程图

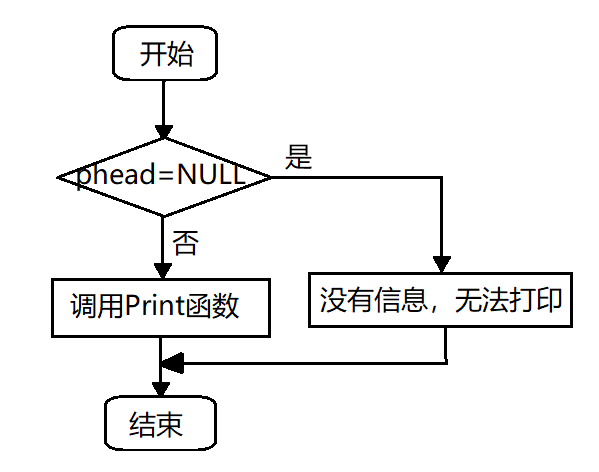


图3-7 打印学籍信息流程图

# 第4章 编码

## 4.1程序中结构体类型的定义

typedef struct student

{

char no[12]; //学号

char name[40]; //姓名

char tele[20]; //电话号码

char ID[20]; //身份证号码

char jg[60]; //籍贯

struct student\* next; //指向下一结点的指针

}student;

## 4.2程序中全局变量、程序中通用的符号常量。

全局变量有：student\* phead = NULL;//定义头节点指针

## 4.3自己设计的所有函数的原型，并说明功能、参数的意义。

|  |  |  |
| --- | --- | --- |
| 函数原型 | 参数的意义 | 功能 |
| void read\_file(student\*\* pphead) | 头节点的二级指针 | 读文件 |
| void save\_file(student\*\* pphead) | 头节点的二级指针 | 保存文件 |
| void menu(); | 无 | 打印菜单 |
| void add(student\*\* pphead); | 头节点的二级指针 | 增加学籍信息 |
| void revise(student\*\* pphead); | 头节点的二级指针 | 修改学籍信息 |
| void find(student\* phead); | 头节点的指针 | 查找学籍信息 |
| void dele(student\*\* pphead); | 头节点的二级指针 | 删除学籍信息 |
| void print(student\* phead); | 头节点的指针 | 打印学籍信息 |
| static Print(student\* tmp) | 结点指针 | 输出指定内容 |
| static student\* search(student\* phead, char no[]) | 头结点指针，no数组指针 | 通过学号找到指定结点 |

1表4-1 主要函数功能

## 4.4主要功能的代码与注释。

4.4.1主函数代码与注释

printf("欢迎使用学生籍贯信息管理系统!\n");

read\_file(&phead);//读文件

int input = 0;

while (1)

{

menu();//打印菜单

printf("请选择:>\n");

scanf("%d", &input);

switch (input)

{

case 1:

add(&phead);//增加信息

break;

case 2:

revise(&phead);//修改信息

break;

case 3:

find(phead);//查找信息

break;

case 4:

dele(&phead);//删除信息

break;

case 5:

print(phead);//打印信息

break;

case 0:

printf("退出...\n");//退出

save\_file(&phead);

return 0;

default:

printf("输入错误，请重新输入\n");

}

}

4.4.2增加学籍信息函数代码与注释

int n = 0;

printf("请输入要录入的个数\n");

scanf("%d", &n);

for (int i = 0; i < n; i++)//录入信息

{

student\* newnode = (student\*)malloc(sizeof(student));

printf("请输入要录入的学号\n");

scanf("%s", newnode->no);

printf("请输入要录入的姓名\n");

scanf("%s", newnode->name);

printf("请输入要录入的电话号码\n");

scanf("%s", newnode->tele);

printf("请输入要录入的身份证号\n");

scanf("%s", newnode->ID);

printf("请输入要录入的籍贯\n");

scanf("%s", newnode->jg);

newnode->next = NULL;

//连接入链表

if (\*pphead == NULL)//判断是否为空表

\*pphead = newnode;

else

{

student\* tmp = \*pphead;

while (tmp->next != NULL)

{

tmp = tmp->next;

}

tmp->next = newnode;

}

}

printf("录入完成...\n");

save\_file(pphead);

system("pause");

system("cls");

4.4.3修改学籍信息函数代码与注释

assert(pphead);//判断指针是否有效

char no[12] = { 0 };

//学号查询：输出该学号学生的信息

printf("请输入要修改的学号:>\n");

scanf("%s", no);

student\* tmp = search(\*pphead, no);

if (tmp == NULL)

{

printf("找不到该学号\n");

}

else

{

printf("请输入修改后的学号:>\n");

scanf("%s", tmp->no);

printf("请输入修改后的姓名：\n");

scanf("%s", tmp->name);

printf("请输入修改后的电话号码\n");

scanf("%s", tmp->tele);

printf("请输入修改后的身份证号\n");

scanf("%s", tmp->ID);

printf("请输入修改后的籍贯\n");

scanf("%s", tmp->jg);

printf("修改完成\n");

}

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

4.4.4查找学籍信息函数代码与注释

int i = 0;//循环变量

char no[12] = { 0 };

//学号查询：输出该学号学生的信息

printf("请输入要查询的学号:>\n");

scanf("%s", no);

student\* tmp = search(phead, no);

if (tmp == NULL)

{

printf("找不到该学号\n");

}

else

{

//输出指定内容

printf("学号\t\t姓名\t\t电话号码\t\t身份证号码\t\t籍贯\n");

Print(tmp);

}

system("pause");

system("cls");

4.4.5删除学籍信息函数代码与注释

if (\*pphead == NULL)//判断是否为空表

{

printf("没有数据，无法删除\n");

system("pause");

system("cls");

return;

}

else

{

char no[12] = { 0 };

printf("请输入需要删除的学号\n");

scanf("%s", no);

student\* tmp = \*pphead;

student\* temp = NULL;

if (strcmp(tmp->no, no) == 0)

{

\*pphead = tmp->next;

tmp->next = NULL;

free(tmp);

printf("删除完成\n");

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

return;

}

while (tmp->next != NULL)

{

if (strcmp(tmp->next->no, no) == 0)

{

temp = tmp->next;

tmp->next = tmp->next->next;

temp->next = NULL;

free(temp);

printf("删除完成\n");

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

return;

}

tmp = tmp->next;

}

printf("找不到该学号，退出:>\n");

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

}

4.4.6打印学籍信息函数代码与注释

student\* tmp = phead;

if (phead == NULL)

{

printf("没有信息，无法打印\n");

}

else

{

printf("学号\t\t姓名\t\t电话号码\t\t身份证号码\t\t籍贯\n");

while (tmp != NULL)

{

Print(tmp);

tmp = tmp->next;

}

printf("打印完成...\n");

}

system("pause");

system("cls");

# 第5章 运行与调试

## 5.1打印菜单

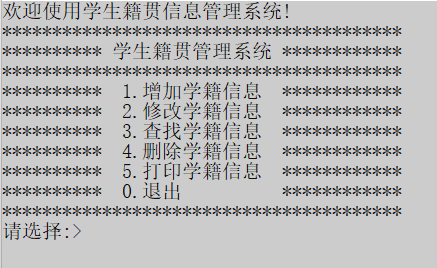


图5-1 打印菜单

## 5.2增加学籍信息

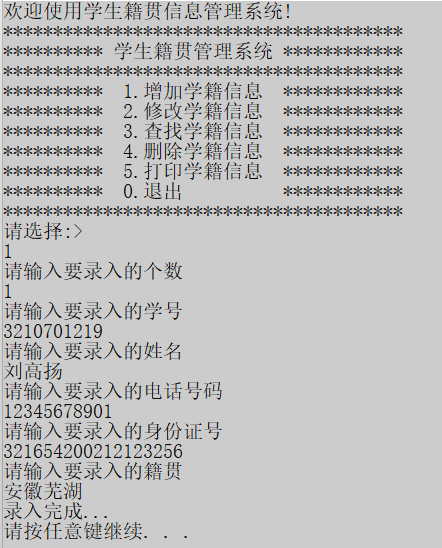


图5-2 增加学籍信息

## 5.3修改学籍信息

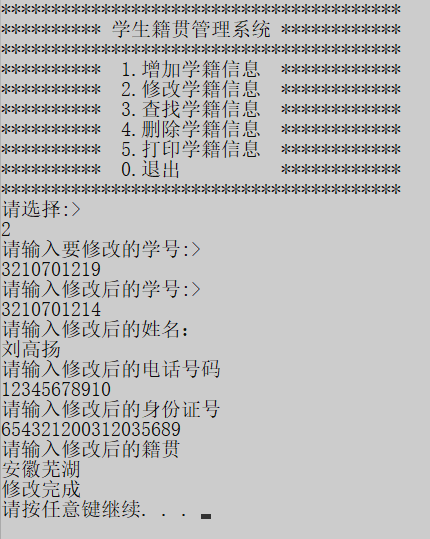


图5-3 修改学籍信息

## 5.4查找学籍信息

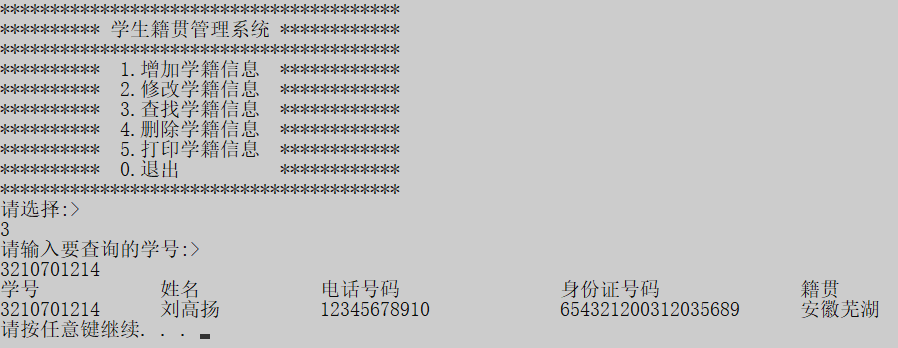


图5-4 查找学籍信息

## 5.5删除学籍信息

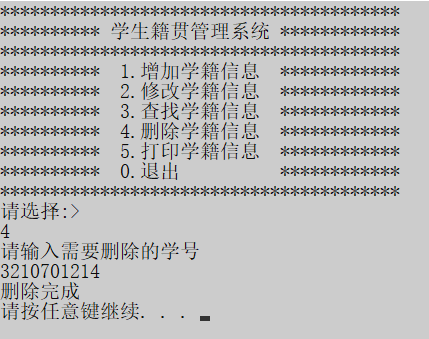


图5-5 删除学籍信息

## 5.6打印学籍信息

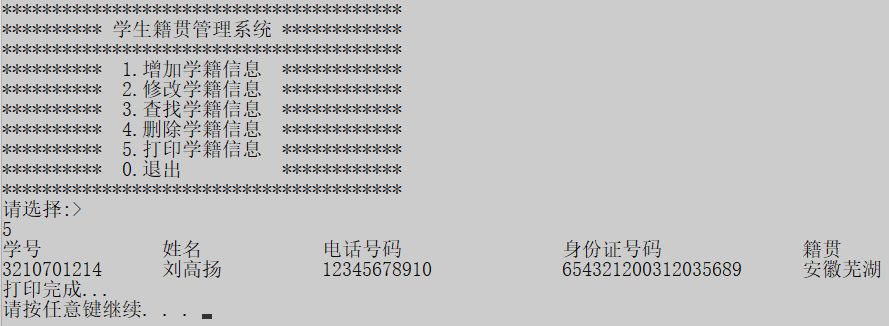


图5-6 打印学籍信息

# 第6章 总结

本次的课程设计是笔者第一次的综合实践，也是第一次接触编写系统的过程。在此期间有收获，也有不足。

第一，认识到了学好C语言的最重要的地方就是要对各个概念都要透彻了解。在这次课设里面笔者才发现自己对概念性问题有很多不了解的地方，导致使用的时候出现很多错误。笔者也非常感谢有这么一个机会，可以发现在基础概念上的不足，同时也让笔者对C语言有更深入的认识。

第二，让笔者认识到了实践的重要性。通过这次的课设，认识到了要通过实践上机操作，才能发现对c语言好多的不足之处，知道了编程时的错误，同时也加深了对程序的格式的重要性。只有上机操作，才能验证程序正确与否，并进而改进，所以在以后的学习过程中，会更加重视实践操作，更好地学习计算机。

第三，做程序时要有模块化思想，要学会分开分模块地实现不同的功能，这样也有利于增大对程序的可读性和条理性。同时程序的规范性也有较重要的作用。还有要尽可能地用多种方法去解同一题，可以锻炼编程能力。

第四，要有自学的能力。在这次的课程设计中，特别是命令菜单程序的编写和单链表的创建，是通过查阅书籍，上网搜索，观看视频等方式解决的，这也是笔者这次最大的收获。

# 参考文献

[1]钟志水, 周鸣争.C语言程序设计[M].成都:电子科技大学出版社, 2015.5.

[2]刘涛,叶明全.c语言程序设计学习指导与实践教程[M].上海交通大学出版社:上海,2017.8.

[3]钱晓捷.汇编语言程序设计（第二版）［M］，电子工业出版社，2003.6. [4]谭浩强.C程序设计（第二版）.清华大学出版社出版，2005.6.

[5]钱晓捷.汇编语言程序设计.电子工业出版社，2003.

[6]普拉塔.CPrimerPlus（第五版）中文版[M].人民邮电出版社，2005.

[7]郭晓.汇编语言与C语言的混合编程及应用［J］.南京农专学报，2002.

[8]常玉龙.Turbo C 2.0实用大全，北京航空航天大学出版社，1994.9。

[9]袁志祥,秦锋. 数据结构（c语言版）例题详解与程序设计基础训练指导(第二版)，中国科学技术大学出版社，2014.7.

[10]程成,陈霞.软件工程[M].机械工业出版社,2003.

# 附录：

1. **梅森素数**

【题目】形如2^n-1的素数成为梅森素数，这里指数n是一个素数。例如，2^2-1=3,2^3-1=7, 使用循环设计程序，求出指数n<30的所有梅森素数。

【编程提示】设置指数n（2≤n<30），t=2；通过t\*2求t=2n；然后用试商法判断素数。

#include <math.h>

#include <stdio.h>

int prime(int n)

{

int i;

long k;

k = sqrt(n) + 1;

for (i = 2; i <= k; i++)

if (n % i == 0)

return 0;

return 1;

}

int main()

{

int mp, n = 0, i;

printf("Mersenne Prime:\n");

for (i = 2; i <= 20; i++)

{

mp = pow(2, i) - 1;

if (prime(mp))

{

n++;

printf("M(%d)=%d", i, mp);

printf("\n");

}

}

printf("the number of Mersenne Prime less than 20 is:%d\n", n);

return 0;

}

1. **选美比赛**

【题目】在选美大奖赛的半决赛现场，有一批选手参加比赛，比赛的规则是最后得分越高，名次越低。当半决赛结束时，要在现场按照选手的出场顺序宣布最后得分和最后名次，获得相同分数的选手具有相同的名次，名次连续编号，不用考虑同名次的选手人数。例如：

选手序号：1，2，3，4，5，6，7

选手得分：5，3，4，7，3，5，6

则输出名次为：3，1，2，5，1，3，4

请采用数组方法编程帮助大奖赛组委会完成半决赛的评分和排名工作。

【编程提示】题目换种说法为：将数组A中的整数从小到大进行连续编号，要求不改变数组中元素的顺序，且相同的整数要具有相同的编号。普通的排序方法均要改变数组元素原来的顺序，显然不能满足要求。为此，引入一个专门存放名次的数组，再采用通常的算法：在尚未排出名次的元素中找出最小值，并对具有相同值的元素进行处理，重复这一过程，直到全部元素排好为止。

#include <stdio.h>

struct player {

int num;

int score;

int rand;

};

void sortScore(struct player psn[], int n)

{

int i, j;

struct player tmp;

for (i = 0; i < n - 1; i++)

for (j = 0; j < n - 1 - i; j++)

{

if (psn[j].score > psn[j + 1].score)

{

tmp = psn[j];

psn[j] = psn[j + 1];

psn[j + 1] = tmp;

}

}

}

void setRand(struct player psn[], int n)

{

int i, j = 2;

psn[0].rand = 1;

for (i = 1; i < n; i++)

{

if (psn[i].score != psn[i - 1].score)

{

psn[i].rand = j;

j++;

}

else

psn[i].rand = psn[i - 1].rand;

}

}

void sortNum(struct player psn[], int n)

{

int i, j;

struct player tmp;

for (i = 0; i < n - 1; i++)

for (j = 0; j < n - 1 - i; j++)

{

if (psn[j].num > psn[j + 1].num)

{

tmp = psn[j];

psn[j] = psn[j + 1];

psn[j + 1] = tmp;

}

}

}

void sortRand(struct player psn[], int n)

{

sortScore(psn, n); /\*以分数为关键字排序\*/

setRand(psn, n); /\*按照分数排名次\*/

sortNum(psn, n); /\*按照序号重新排序\*/

}

int main()

{

struct player psn[7] = { {1, 5, 0}, {2, 3, 0}, {3, 4, 0}, {4, 7, 0}, {5, 3, 0}, {6, 5, 0}, {7, 6, 0} };

int i;

sortRand(psn, 7);

printf("num score rand \n");

for (i = 0; i < 7; i++)

{

printf("%d%6d%6d\n", psn[i].num, psn[i].score, psn[i].rand);

}

getchar();

return 0;

}

1. **学生籍贯管理系统**
2. 头文件

#pragma once

#include"stdio.h"

#include"stdlib.h"

#include<assert.h>

#include<Windows.h>

typedef struct student

{

char no[12]; //学号

char name[40]; //姓名

char tele[20]; //电话号码

char ID[20]; //身份证号码

char jg[60]; //籍贯

struct student\* next; //指向下一结点的指针

}student;

void read\_file(student\*\* pphead);//读文件

void save\_file(student\*\* pphead);//保存文件

void menu();//打印菜单

void add(student\*\* pphead);//增加学籍信息

void revise(student\*\* pphead);//修改学籍信息

void find(student\* phead);//查询学籍信息

void dele(student\*\* pphead);//删除学籍信息

void print(student\* phead);//打印学籍信息

1. 源文件

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include"system.h"

void read\_file(student\*\* pphead)//读文件

{

assert(pphead);

FILE\* fp = fopen("data.txt", "r");//定义文件指针

if (!fp)判断是否打开成功

return;

while (1)

{

student\* newnode = (student\*)malloc(sizeof(student));

if (newnode == NULL)判断是否读取成功

{

printf("文件读取失败！");

exit(0);

}

else

{

if (fread(newnode, sizeof(student), 1, fp) < 1)

break;

newnode->next = \*pphead;

\*pphead = newnode;

}

}

fclose(fp);//关闭文件

}

void save\_file(student\*\* pphead)//保存文件

{

assert(pphead);

FILE\* fp = fopen("data.txt", "w+");//定义文件指针

student\* tmp = \*pphead;

if (!fp)//判断文件是否打开成功

return;

while (tmp)

{

fwrite(tmp, sizeof(student), 1, fp);

tmp = tmp->next;

}

fclose(fp);

}

static Print(student\* tmp)//输出指定内容

{

printf("%s\t%s\t\t%s\t\t%s\t%s\n", tmp->no, tmp->name, tmp->tele, tmp->ID, tmp->jg);

}

static student\* search(student\* phead, char no[])//通过学号找到指定结点

{

assert(phead);

student\* tmp = phead;

while (tmp != NULL)

{

if (strcmp(tmp->no, no) == 0)

{

return tmp;

}

tmp = tmp->next;

}

return NULL;

}

void menu()//打印菜单

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\* 学生籍贯管理系统 \*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\* 1.增加学籍信息 \*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\* 2.修改学籍信息 \*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\* 3.查找学籍信息 \*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\* 4.删除学籍信息 \*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\* 5.打印学籍信息 \*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\* 0.退出 \*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

void add(student\*\* pphead)

{

int n = 0;

printf("请输入要录入的个数\n");

scanf("%d", &n);

for (int i = 0; i < n; i++)//录入信息

{

student\* newnode = (student\*)malloc(sizeof(student));

printf("请输入要录入的学号\n");

scanf("%s", newnode->no);

printf("请输入要录入的姓名\n");

scanf("%s", newnode->name);

printf("请输入要录入的电话号码\n");

scanf("%s", newnode->tele);

printf("请输入要录入的身份证号\n");

scanf("%s", newnode->ID);

printf("请输入要录入的籍贯\n");

scanf("%s", newnode->jg);

newnode->next = NULL;

//连接入链表

if (\*pphead == NULL)//判断是否为空表

\*pphead = newnode;

else

{

student\* tmp = \*pphead;

while (tmp->next != NULL)

{

tmp = tmp->next;

}

tmp->next = newnode;

}

}

printf("录入完成...\n");

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

}

void revise(student\*\* pphead)

{

assert(pphead);//判断指针是否有效

char no[12] = { 0 };

//学号查询：输出该学号学生的信息

printf("请输入要修改的学号:>\n");

scanf("%s", no);

student\* tmp = search(\*pphead, no);

if (tmp == NULL)

{

printf("找不到该学号\n");

}

else

{

printf("请输入修改后的学号:>\n");

scanf("%s", tmp->no);

printf("请输入修改后的姓名：\n");

scanf("%s", tmp->name);

printf("请输入修改后的电话号码\n");

scanf("%s", tmp->tele);

printf("请输入修改后的身份证号\n");

scanf("%s", tmp->ID);

printf("请输入修改后的籍贯\n");

scanf("%s", tmp->jg);

printf("修改完成\n");

}

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

}

void find(student\* phead)

{

int i = 0;//循环变量

char no[12] = { 0 };

//学号查询：输出该学号学生的信息

printf("请输入要查询的学号:>\n");

scanf("%s", no);

student\* tmp = search(phead, no);

if (tmp == NULL)//判断是否查询到该学号

{

printf("找不到该学号\n");//没有查询到

}

else//查询到

{

//输出指定内容

printf("学号\t\t姓名\t\t电话号码\t\t身份证号码\t\t籍贯\n");

Print(tmp);

}

system("pause");

system("cls");

}

void dele(student\*\* pphead)

{

if (\*pphead == NULL)//判断是否为空表

{

printf("没有数据，无法删除\n");

system("pause");

system("cls");

return;

}

else

{

char no[12] = { 0 };

printf("请输入需要删除的学号\n");

scanf("%s", no);

student\* tmp = \*pphead;

student\* temp = NULL;

if (strcmp(tmp->no, no) == 0)

{

\*pphead = tmp->next;

tmp->next = NULL;

free(tmp);

printf("删除完成\n");

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

return;

}

while (tmp->next != NULL)

{

if (strcmp(tmp->next->no, no) == 0)

{

temp = tmp->next;

tmp->next = tmp->next->next;

temp->next = NULL;

free(temp);

printf("删除完成\n");

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

return;

}

tmp = tmp->next;

}

printf("找不到该学号，退出:>\n");

save\_file(pphead);//同步文件信息

system("pause");

system("cls");

}

}

void print(student\* phead)

{

student\* tmp = phead;

if (phead == NULL)//判断是否为空表

{

printf("没有信息，无法打印\n");//该表为空表

}

Else//不为空表

{

printf("学号\t\t姓名\t\t电话号码\t\t身份证号码\t\t籍贯\n");

while (tmp != NULL)

{

Print(tmp);

tmp = tmp->next;

}

printf("打印完成...\n");

}

system("pause");

system("cls");

}

student\* phead = NULL;//定义头节点指针

int main()

{

printf("欢迎使用学生籍贯信息管理系统!\n");

read\_file(&phead);//读文件

int input = 0;

while (1)

{

menu();//打印菜单

printf("请选择:>\n");

scanf("%d", &input);

switch (input)

{

case 1:

add(&phead);//增加信息

break;

case 2:

revise(&phead);//修改信息

break;

case 3:

find(phead);//查找信息

break;

case 4:

dele(&phead);//删除信息

break;

case 5:

print(phead);//打印信息

break;

case 0:

printf("退出...\n");//退出

save\_file(&phead);

return 0;

default:

printf("输入错误，请重新输入\n");

}

}

}