XXXX大学

XXXXXX学院实验报告

实验题目： 课程设计报告设计与撰写

课程名称： 学生成绩管理程序

实验类型：□演示性 □验证性 □操作性 ■设计性 □综合性

实验日期： XXXX 实验地点：XXXXX实验学时：XXXX

指导教师：XXXXXX

目录

[1.问题描述 1](#_Toc35719578)

[1.1题目内容与要求 1](#_Toc35719579)

[1.2需求分析 1](#_Toc35719580)

[2. 概要设计 3](#_Toc35719581)

[2.1模块划分 3](#_Toc35719582)

[2.2文件结构 4](#_Toc35719583)

[2.3系统主要执行逻辑设计 5](#_Toc35719584)

[2.4函数原型及关系 5](#_Toc35719585)

[2.4.1函数调用关系 6](#_Toc35719586)

[2.4.2函数原型设计 6](#_Toc35719587)

[3.详细设计 10](#_Toc35719588)

[3.1用户交互界面设计 10](#_Toc35719589)

[3.2核心数据结构设计 12](#_Toc35719590)

[3.3核心算法设计 12](#_Toc35719591)

[4.测试 17](#_Toc35719592)

[5.总结 22](#_Toc35719593)

[5.1遇到的问题及解决方法 22](#_Toc35719594)

[5.2感言 22](#_Toc35719595)

# 1.问题描述

## 1.1题目内容与要求

## 本程序用于教学单位（院/系）的学生成绩管理。要求程序能够实现学生信息录入（可以实现增加、删除、修改学生的基本信息）、单科学习成绩的录入；班级内单科成绩排名；成绩查询：查询某个学生的各科成绩、统计所有不及格科目超过2科的学生名单；

## 1.2需求分析

（1）添加学生信息，用户需要在控制台输入学生的学号，若学号格式输入错误或已存在该学号学生信息给出重新输入的提示信息，然后依次输入学生的姓名、性别、各科成绩，结束后，给出显示录入结束提示信息；（2）删除某个学生信息，用户许要在控制台输入要删除学生的学号，若学号格式输入错误或不存在该学号学生信息给出重新输入的提示信息，然后给出用户二次决定的机会，若依然决定删除，删除后给出成功删除的提示信息；（3）修改学生信息，用户需要的在控制台输入要修改信息学生的学号，若学号格式输入错误或不存在该学号学生信息给出重新输入的提示信息，依次给出学生姓名、性别 、各科成绩等信息，若修改，则给出修改成功的提示信息；（4）显示所有学生信息，选中该菜单后，在控制台打印出所有学生的所有信息；（5）单科成绩录入，用户需要在控制台选择要录入的科目代号，接下来依次根据显示的学号班级姓名信息输入对应的单科成绩，结束后，给出显示录入结束提示信息。若输入超过正常成绩数值范围应报错；（6）查看单科成绩排名，用户需要在控制台选择要排序科目的代号，接下来依次显示按该科成绩排名的学生信息，查看完毕后按任意键返回主菜单；（7）成绩查询，用户需要在控制台输入要查询成绩的学生学号，若学号格式输入错误或不存在该学号学生信息给出重新输入的提示信息，若学号输入正确，接下来依次显示该学生的所有信息；（8）统计所有挂科超过两个的学生名单，选择该功能后，依次显示挂科超过两科的所有学生信息，查看完毕后，按任意键返回主菜单。

# 2. 概要设计

## 2.1模块划分

本系统共划分为11个模块。具体介绍如表2.1所示：

表2.1模块划分说明

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 模块名称 | 模块作用 | 包含函数代码 | 包含函数名 | 源文件名称 |
| 1 | 主菜单 | 显示菜单，提供选项输入接口，以显示子菜单 | 1-1 | Showmenu | menu.c |
| 1-2 | GetMenuChoose | menu.c |
| 2 | 信息录入 | 依次录入1-n个学生基本信息 | 2-1 | CreateStudent | Add.c |
| 2-2 | AddStudent | Add.c |
| 2-3 | GetPrevAddr | Add.c |
| 3 | 信息显示 | 显示学生基本信息及成绩信息 | 3-1 | ShowAll | Showall.c |
| 3-2 | ShowStudentCount | Showall.c |
| 4 | 删除信息 | 删除学生基本信息及成绩信息 | 4-1 | DelStudent | delete.c |
| 4-2 | DelAll | delete.c |
| 5 | 修改信息 | 修改学生的基本信息及成绩信息 | 5-1 | ModityStudent | Modity.c |
| 6 | 录入单科学习成绩 | 录入学生的单科成绩信息 | 6-1 | InputScore | Addone.c |
| 7 | 成绩查询 | 查询某个学生基本信息及成绩信息 | 7-1 | Search | Search.c |
| 7-2 | GetAddr | Search.c |
| 8 | 单科成绩排序 | 按各科成绩排名 | 8-1 | SerchResuls | Sort.c |
| 8-2 | Sort1 | Sort.c |
| 8-3 | Sort2 | Sort.c |
| 8-4 | Sort3 | Sort.c |
| 8-5 | Sort4 | Sort.c |
| 9 | 统计不符合要求的学生名单 | 显示挂科超过两科的学生的所有信息 | 9-1 | Statistic | Statistic.c |
| 10 | 从文件中读取学生信息 | 从文件中读取学生信息 | 10-1 | LoadFromFile | Read.c |
| 11 | 把学生信息保存到文件当中 | 把学生信息保存到文件当中 | 11-1 | SaveToFile | Save.c |

## 2.2文件结构

本系统有14个文件。本系统包括主程序文件main.c, 用来显示主菜单以及存储各个分菜单的选项；源程序文件Add.c,用来录入学生信息，Showall.c，主要负责显示所有学生信息；源程序文件delete.c用来删除学生信息；源程序文件Modity.c，用来修改学生信息；源程序文件Addone.c，用来录入学生单科成绩；源程序文件Search.c，用来查询学生成绩；源程序文件Sort.c，用来以单科成绩对学生进行排名；源程序文件Statistic.c，用来统计不符合要求的学生名单；源程序文件Read.c，用来从文件中读取学生信息；源程序文件Save.c，用来把学生信息保存到文件当中；以及头文件sms.h,用来进行函数原型的声明和数据结构声明。

## 2.3系统主要执行逻辑设计

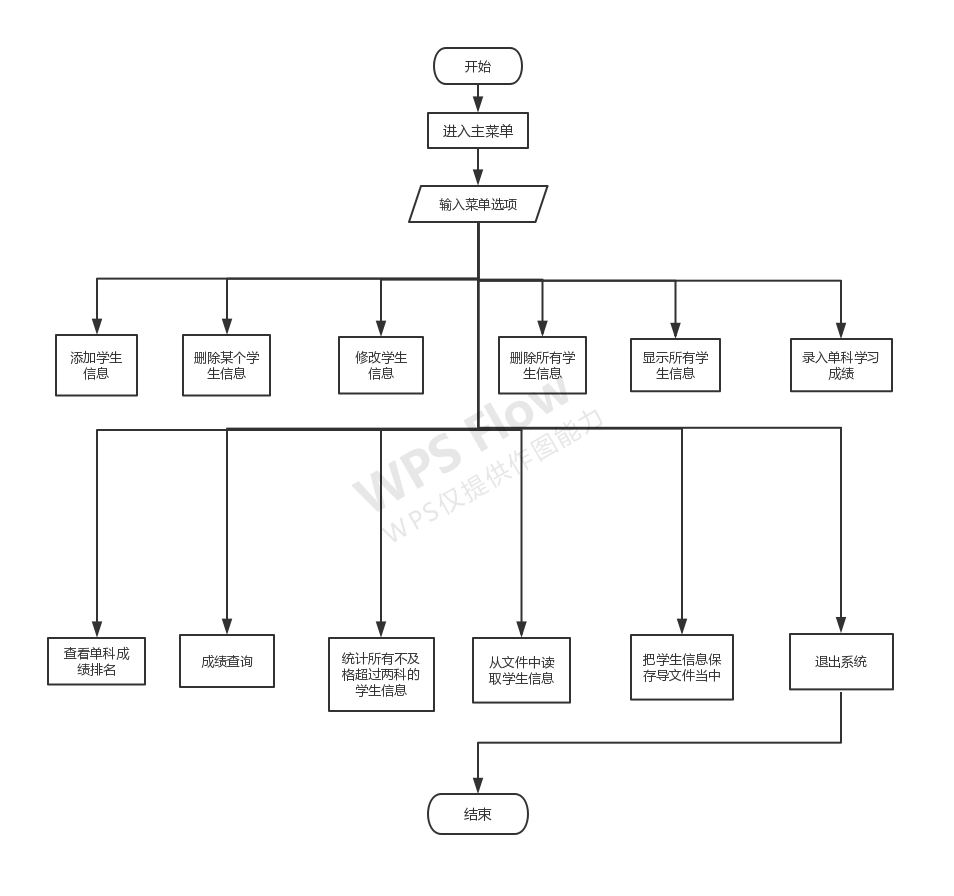
本系统主要执行逻辑，也即主函数流程图如图2.1所示：

图2.1系统主要执行逻辑

## 2.4函数原型及关系

····巴拉巴拉

### 2.4.1函数调用关系

本系统函数调用关系如图2.2所示：

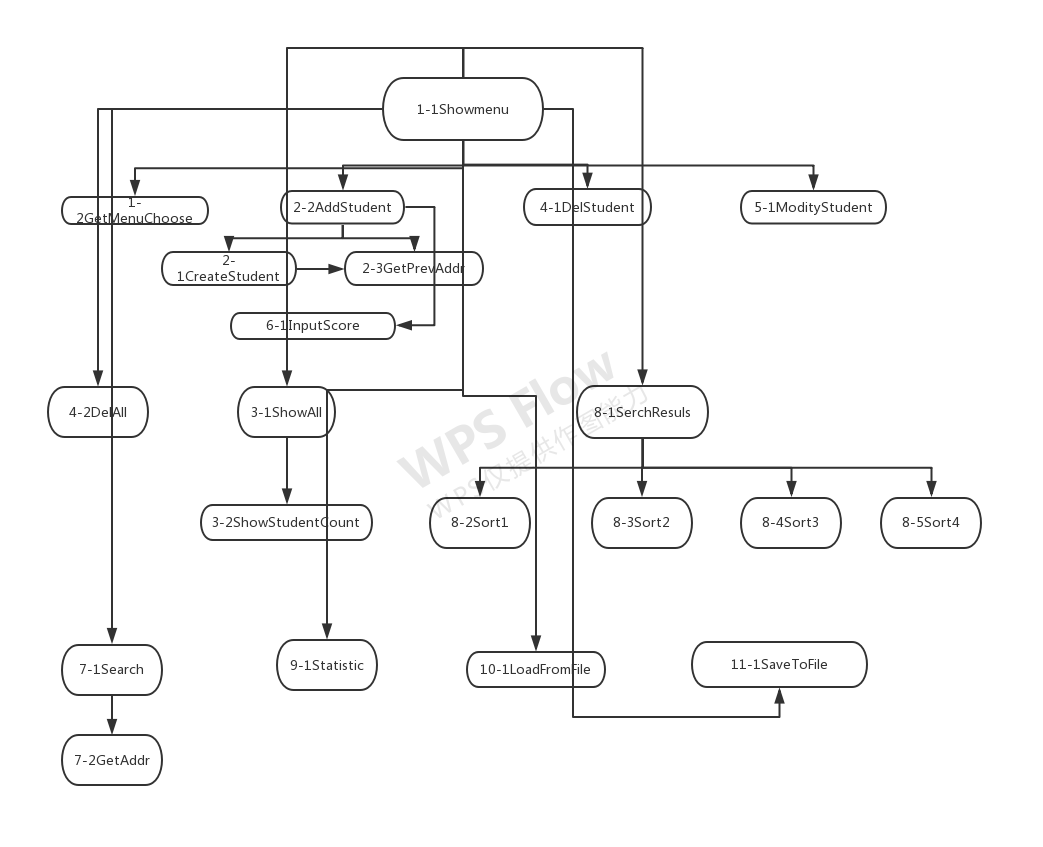


图2.2 函数调用关系图

### 2.4.2函数原型设计

现将本系统的11个模块包含的函数原型描述如下：

1.主菜单模块函数

（1）函数1-1Showmenu

void Showmenu()；

函数作用：显示主菜单。

1. 函数1-2GetMenuChoose

int GetMenuChoose()；

函数作用：获取用户选择的菜单编号

1. 信息录入模块函数
2. 函数CreateStudent

PSTUDENT CreateStudent()；

函数作用：创建一个节点，它会返回一个新创建的学生信息节点的指针 。

1. 函数AddStudent

int AddStudent(PSTUDENT pstu)；

函数作用：把学生信息节点加入到链表中。

参数解释：pstu为指针型结构体变量，表示新创建的学生信息节点的指针。

1. 函数GetPrevAddr

PSTUDENT GetPrevAddr(int ID)；

函数作用：返回指定编号学生节点的上一个节点的指针 。

参数解释：ID为整型变量，表示学生的学号；通常从CreateStudent函数接受数据。

1. 信息显示模块函数
2. 函数ShowAll

void ShowAll()；

函数作用：显示所有学生信息

1. 函数ShowStudentCount

int ShowStudentCount()；

函数作用：显示已有学生信息数量。

1. 删除信息模块函数
2. 函数DelStudent

void DelStudent(int ID)；

函数作用：删除学号为ID的学生信息。

参数解释：ID为整型变量，表示学生的学号；通常从Getinput函数接受数据。

1. 函数DelAll

void DelAll()；

函数作用：删除所有学生信息。

1. 修改信息模块函数
2. 函数ModityStudent

void ModityStudent(int ID)；

函数作用：修改学生信息。

参数解释：ID为整型变量，表示学生的学号；通常从Getinput函数接受数据。

1. 录入单科学习成绩模块函数
2. 函数InputScore

PSTUDENT InputScore()；

函数作用：录入单科学习成绩。。

1. 成绩查询模块函数
2. 函数Search()

void Search()；

函数作用：成绩查询。

1. 函数GetAddr

PSTUDENT GetAddr(int ID)；

函数作用：返回指定编号学生节点的指针。

参数解释：ID为整型变量，表示学生的学号；通常从Getinput函数接受数据。

1. 单科成绩排序模块函数
2. 函数SerchResuls

void SerchResuls()；

函数作用：浏览学生成绩。

1. 函数Sort1

void Sort1()；

函数作用：对学生科目一成绩进行排序。

1. 函数Sort2

void Sort2()；

函数作用：对学生科目二成绩进行排序。

1. 函数Sort3

void Sort3()；

函数作用：对学生科目三成绩进行排序。

1. 函数Sort4

void Sort4()；

函数作用：对学生总成绩进行排序。

1. 统计不符合要求的学生名单 模块函数
2. 函数Statistic

void Statistic()；

函数作用：统计不符合要求的学生名单 。

1. 从文件中读取学生信息模块函数
2. 函数LoadFromFile

void LoadFromFile()；

函数作用：从文件中读取学生信息。

1. 把学生信息保存到文件当中模块函数
2. 函数SaveToFile

void SaveToFile()；

函数作用：把学生信息保存到文件当中。

# 3.详细设计

## 3.1用户交互界面设计

这一小节将用户的交互界面做一个细致的说明，每一个界面设计的过程，设计的亮点（优点，一些小细节对用户体验的提升，比如刷屏的时机，比如显示的美观性。不过界面要尽可能简洁明快，可以（但不是必须）在背景和字体颜色等方面有高对比度的设计，但是严重不推荐使用花里胡哨五颜六色的背景。）。设计的结果展示出来，同样以图的形式展示出来。

当然，不是每个界面都要罗列出来，列出主要的和有亮点的。主要是为了文档读者更快，更准确的抓住你的软件系统的用户交互设计。

读取学生信息界面完成不存在学生信息读取失败交互，如图3.1所示：

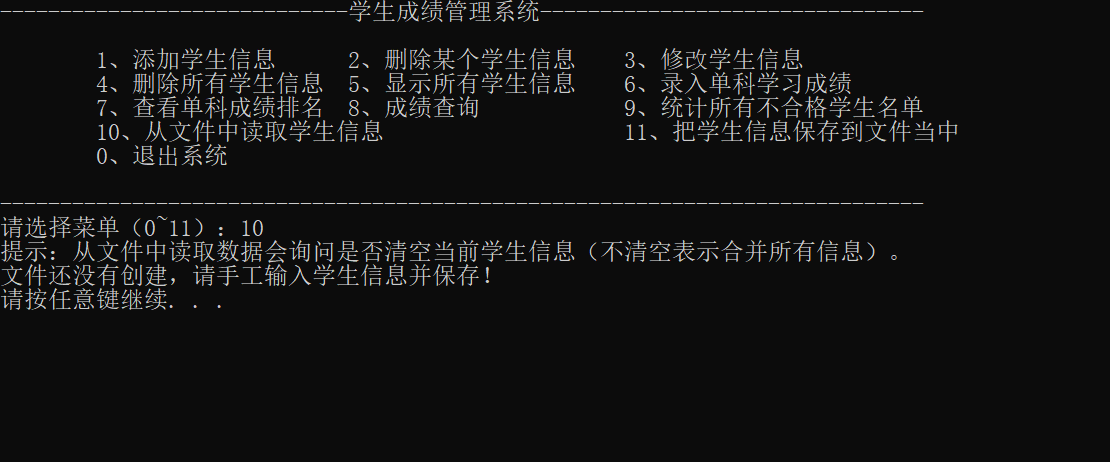
****

图3.1读取学生信息界面

修改学生信息界面完成给用户二次判断交互，如图3.2所示：

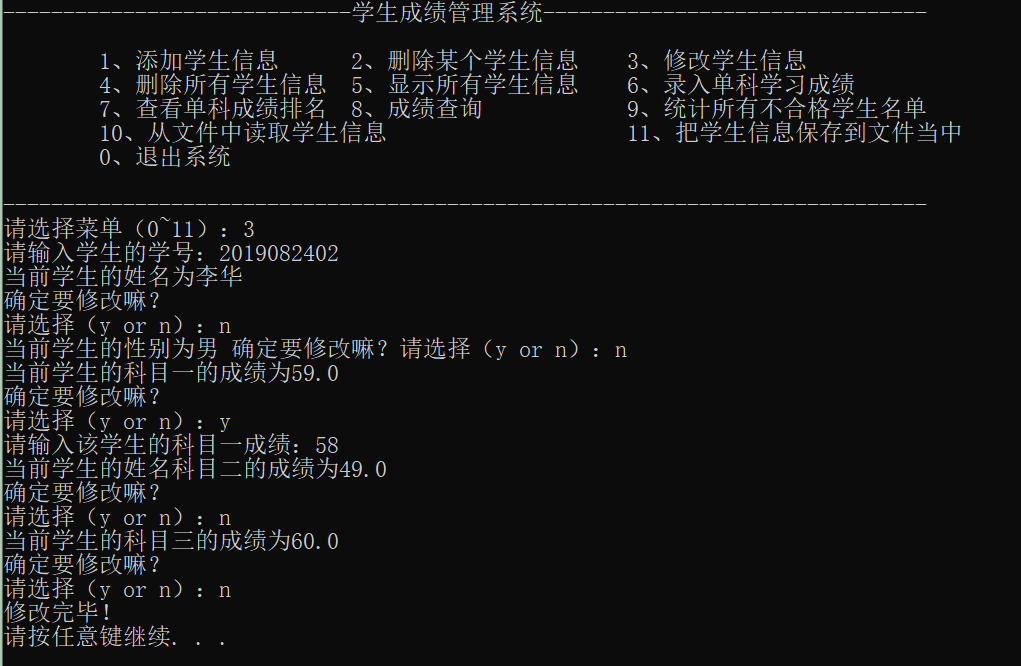
****

图3.2修改学生信息界面

保存学生信息界面完成给用户二次判断交互，如图3.3所示：

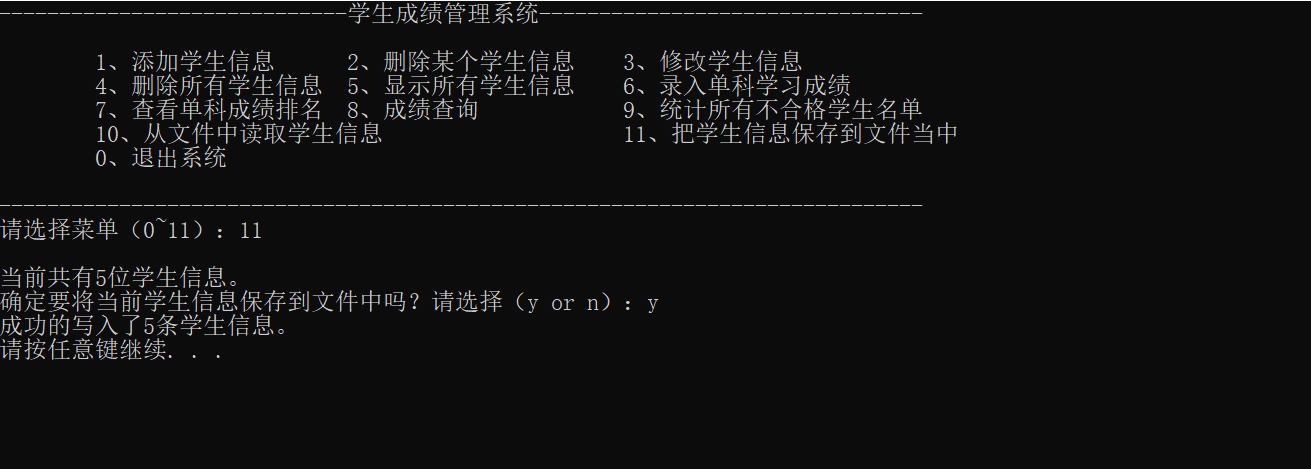
****

图3.3保存学生信息界面

## 3.2核心数据结构设计

学生信息属性结构体，定义ID为整形变量，用来存储学生的学号；定义Name为字符型数组，用于存储学生的姓名；定义SEX为枚举型，用于存储学生的性别；定义Mark1、Mark2、Mark3、Mark4为浮点型变量用于分别存储学生的科目一、科目二、科目三、以及总成绩；定义next为指针型结构体变量，用于指向下一结构体。

## 

## 3.3核心算法设计

1.成绩排序算法

（1）初始化该科成绩的最高分，首先遍历整个链表，若有该分的同学成绩则打印出该学生的信息，循环递减分数，再遍历整个链表，若有该分的同学成绩则打印出该学生的信息，最后达到成绩排序的效果。

（2）输入数据 无

（3）输出数据 按选定的科目成绩依次打印出学生排名信息

（4）算法流程图（尽量避免自然语言描述，应以流程图形式描述）

成绩排序算法流程图如图3.4所示：

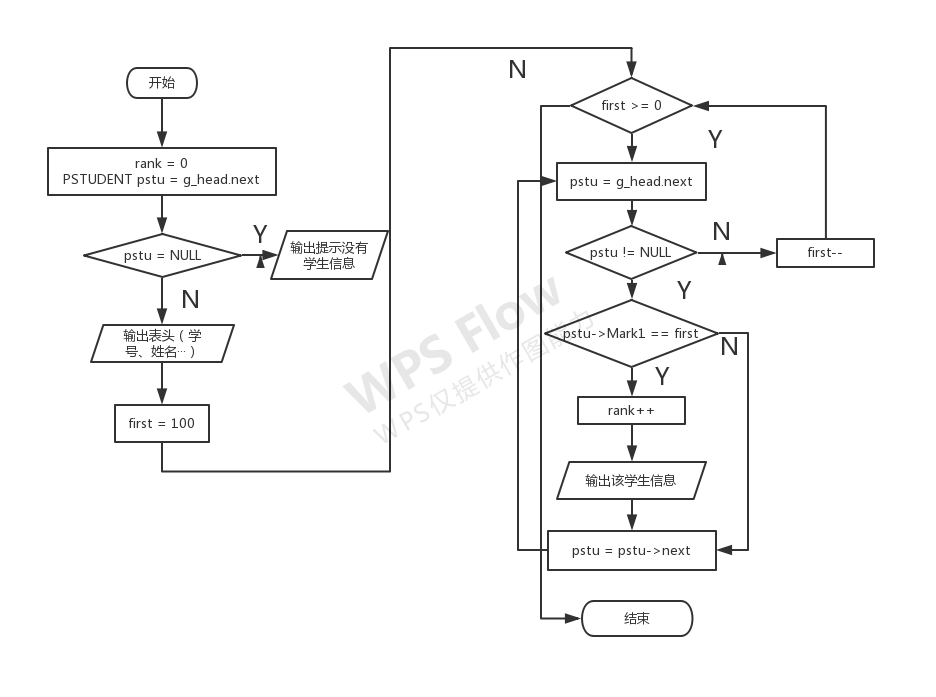


图3.4成绩排序算法流程图

2.统计所有不合格学生信息算法

1. 遍历所有节点，若有同学挂科数量超过两科，则打印出所有学生信息，遍历完毕后退出。
2. 输入数据 无
3. 输出数据 输出所有挂科数量超过两科的学生信息

（4）算法流程图 统计所有不合格学生信息算法流程图如图3.5所示：

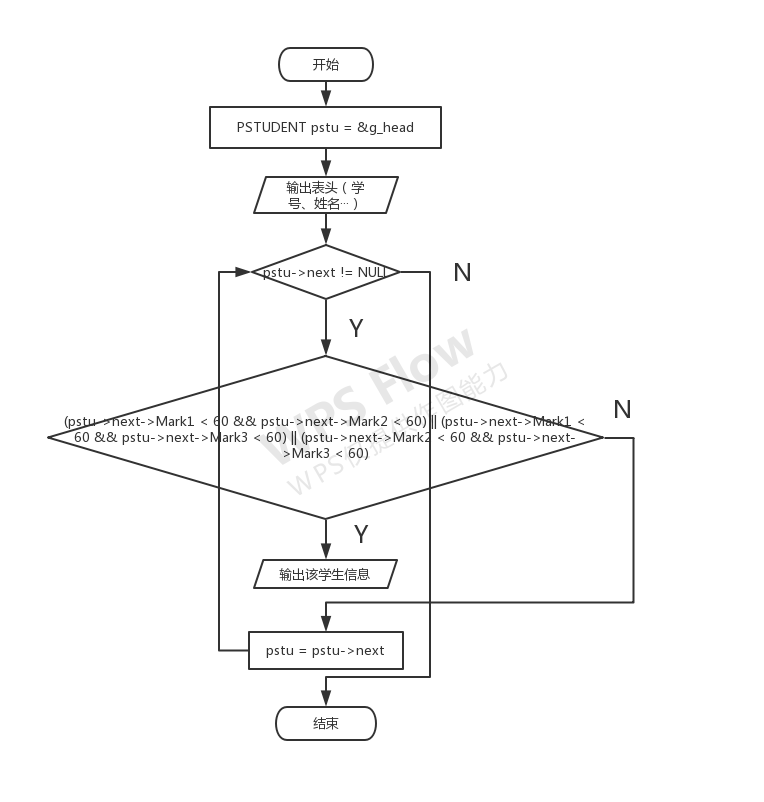


图3.5统计所有不合格学生信息算法流程图

1. 添加学生信息算法
2. 创建一个节点添加学生信息，然后把这个节点放到整个链表的最后
3. 输入数据 输入学生学号、姓名、性别、各科成绩等信息
4. 输出数据 无
5. 算法流程图 添加学生信息算法流程图如图3.6所示：

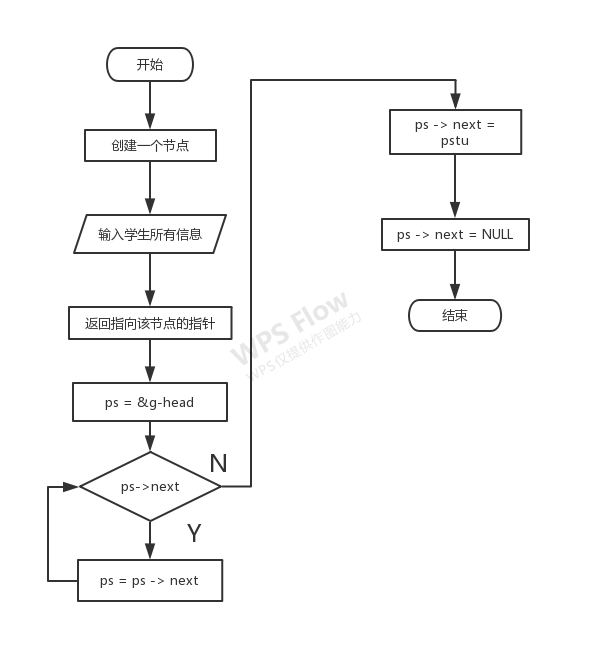


图3.6添加学生信息算法流程图

4.1删除学生信息算法

1. 输入要删除信息学生的学号，若学号输入错误或不存在该学号学生信息给出提示，确定删除后，返回该节点上一节点的指针，将该指针指向删除信息节点的下一节点。
2. 输入数据 输入要删除信息学生的学号
3. 输出数据 无
4. 算法流程图 删除某个学生信息的算法流程图如图3.7所示：

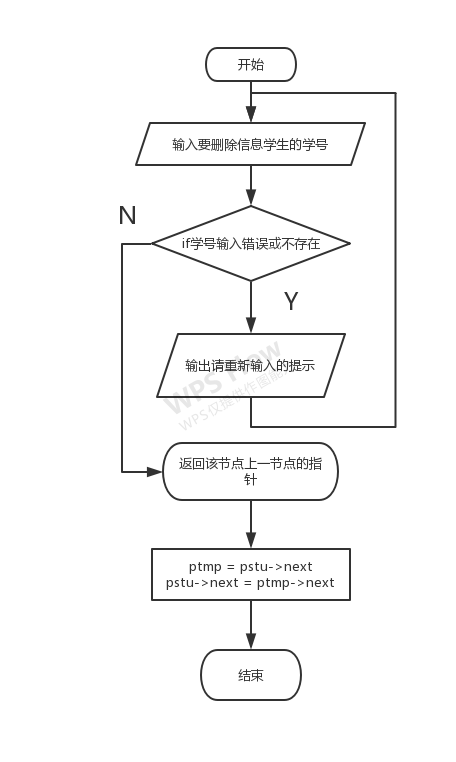


图3.7删除某个学生信息的算法流程图

# 

# 4.测试

这一章主要阐述测试的工作。

测试分为黑盒测试和白盒测试。白盒测试等到大三的时候，大家学过的软件工程的课程就会理解。本章主要描述黑盒测试。也就是运行程序并验证结果。

任何一个测试过程通常包括两个核心环节。第一，编写测试用例。第二，运行测试用例并验证。

测试用例包括如下两个方面：（1）测试数据，（2）预期结果。例如：对于某一个成绩排序模块（或函数）。测试用例可以写成：

测试用例编号1-1：（成绩数据（56，87，100，98），（100，98，87，56））

接下来给出运行的实际结果：（100，98，87，56），验证成功。

测试还应该包括一些边缘数据的测试。比如，一个成绩输入的模块（或函数），可以写出测试用例（成绩数据（101），（错误提示：输入有误））。接下来给出运行的实际结果：（错误提示：输入有误）。

当然，上述的实际结果都可以附以截图的形式增加表述的直观性。

测试还分为单体测试，集成测试和运行测试。考虑到大家是初学者。只需要完成系统运行测试就好。不硬性要求体现单体测试。系统运行测试可以以截图形式展现核心功能的运行结果。不必面面俱到。

测试编号1-1从文件中读取学生信息模块 读取学生信息失败测试如图4.1所示

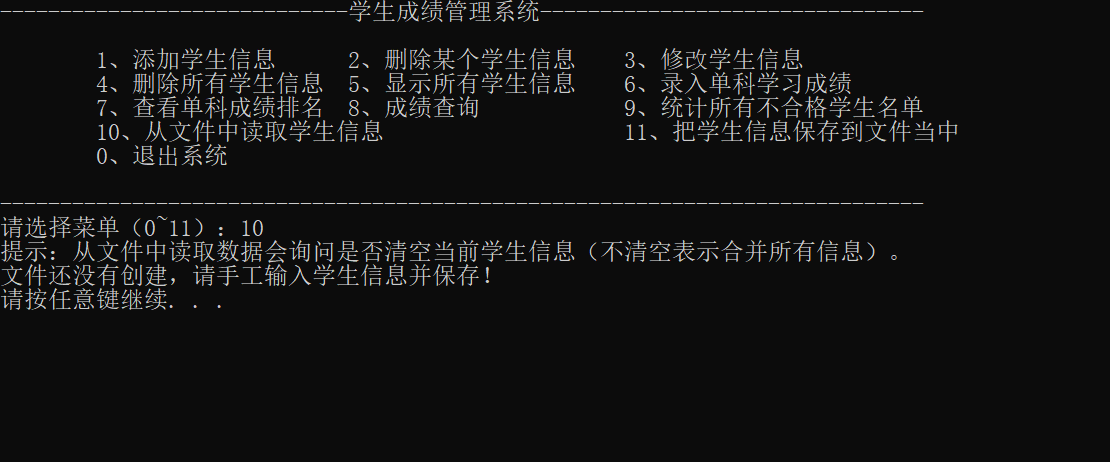
****

图4.1读取学生信息失败测试

测试编号1-2 信息录入模块 添加学生信息测试如图4.2所示

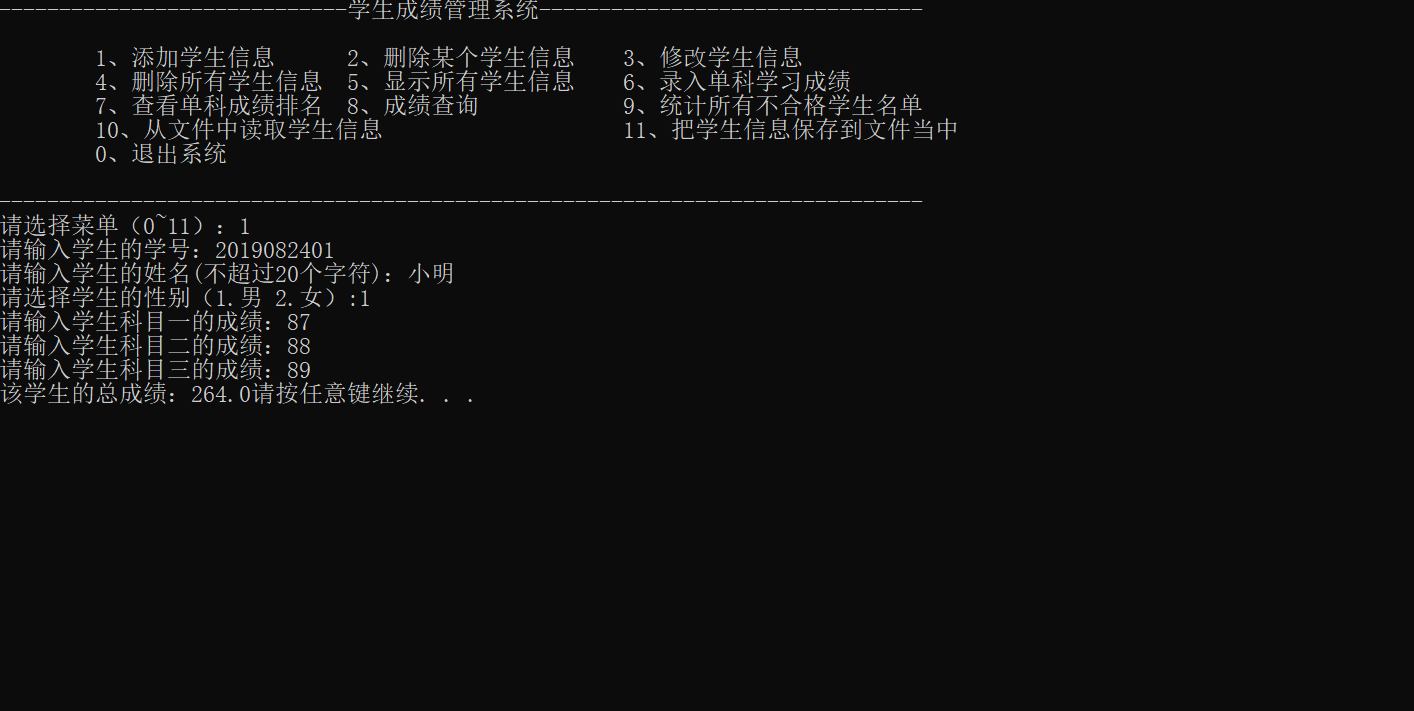
****

图4.2 添加学生信息测试

测试编号1-3删除信息模块，删除某个学生信息测试如图4.3所示

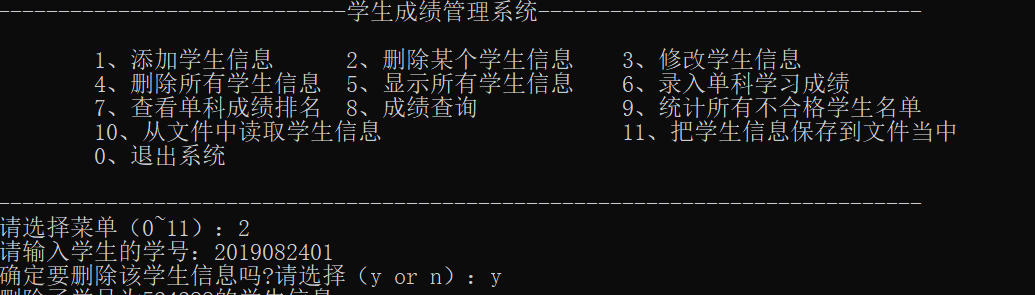
****

图4.3删除某个学生信息测试

测试编号1-4修改信息模块，修改某个学生信息测试如图4.4所示

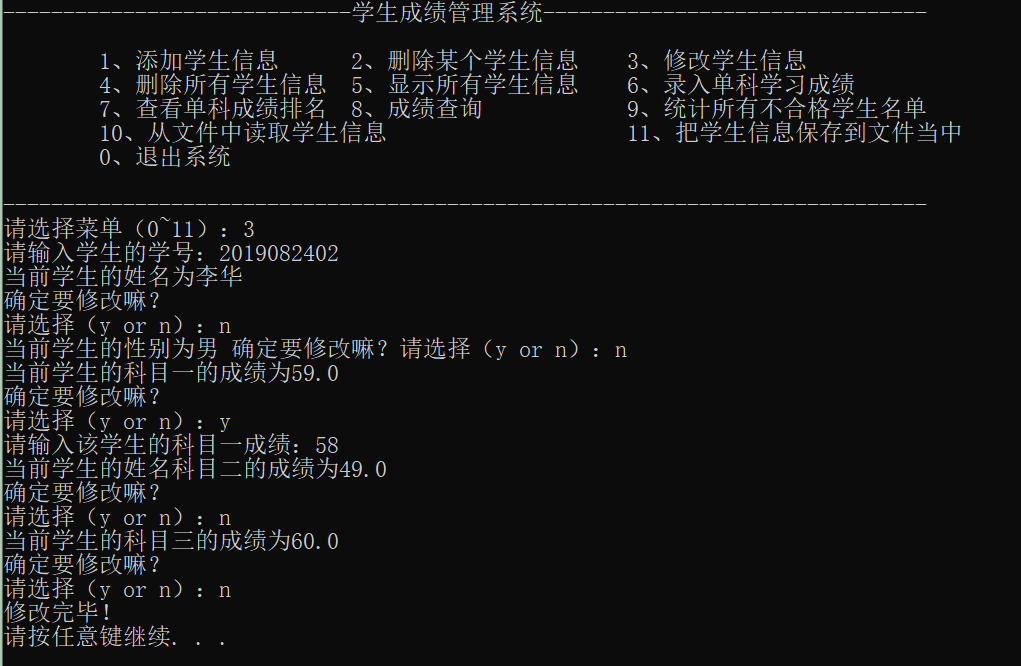
****

图4.4修改某个学生信息测试

测试编号1-5显示信息模块，显示所有学生信息测试如图4.5所示

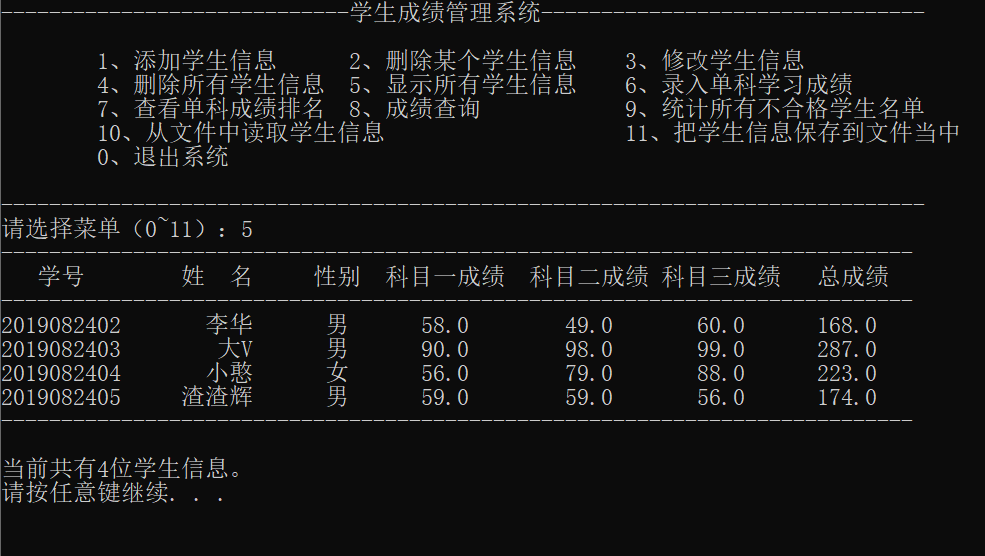
****

图4.5显示所有学生信息测试

测试编号1-6信息录入模块，录入单科学生成绩测试如图4.6所示

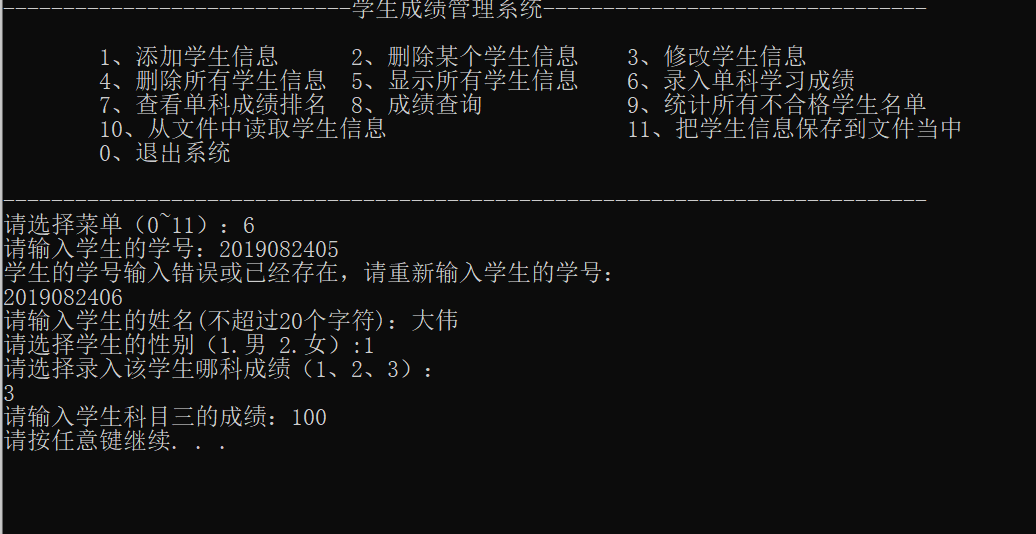
****

图4.6录入单科学生成绩测试

测试编号1-7查看单科成绩排名模块，查看单科成绩排名测试如图4.7所示

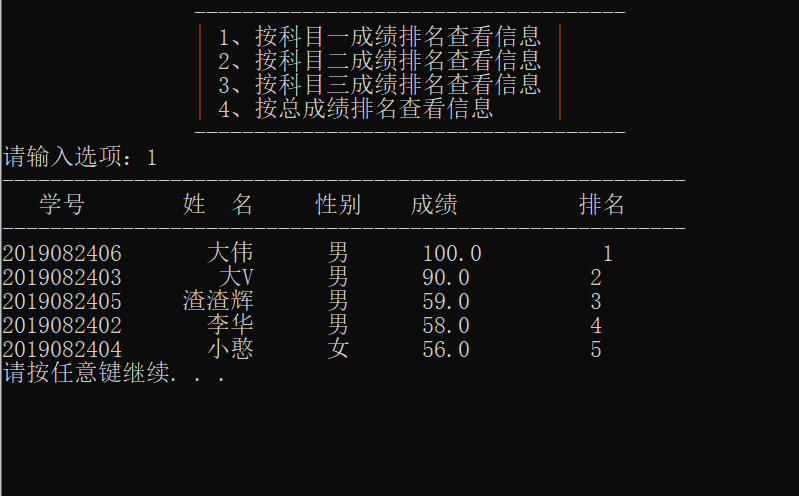
****

图4.7查看单科成绩排名测试

测试编号1-8统计不符合要求的学生名单模块，统计挂科超过两科的学生信息测试如图4.8所示

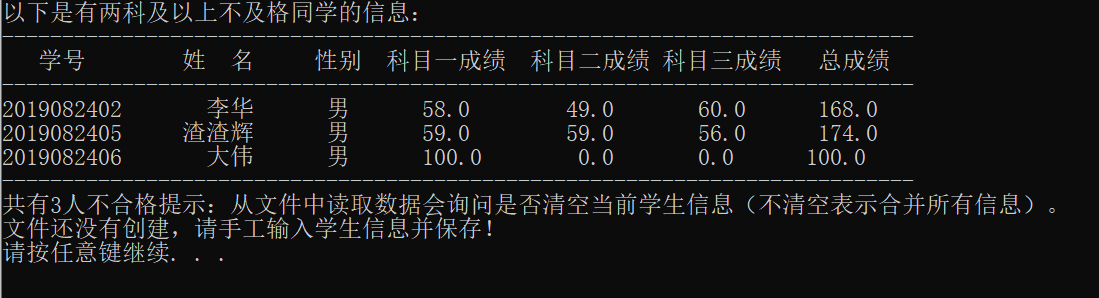
****

图4.8统计挂科超过两科的学生信息测试

测试编号1-9成绩查询模块，成绩查询测试如图4.9所示

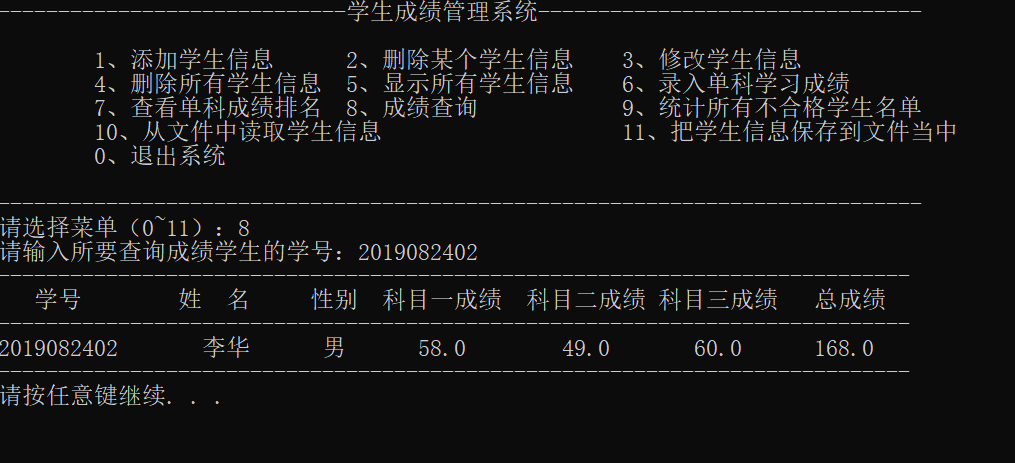
****

图4.9成绩查询测试

# 5.总结

## 5.1遇到的问题及解决方法

1. 在做添加学生信息模块时，总是把创建的节点加入不到链表当中，后来，通过上CSDN网查找大佬的帖子，明白了要写一个函数遍历整个链表，把新创建的节点加入到链表的末尾。
2. 在做完几个基本函数后，测试数据过程中控制台总是因为测试次数过多显的很乱，听X老师为我们讲课时提到了清屏函数，没太当回事，后来请学长帮忙查找bug时，学长叫我百度清屏函数然后运用，解决了控制台很乱的问题。
3. 在做单科成绩排序模块时，在网上看了好几节链表的冒泡排序法，不是很理解，卡了好几天没有动手，后来和同学交流的时候听同学讲了一种很容易理解但是有局限的方法，迫不得已就暂且用了这个简单的方法，顺利完成了课设。

## 5.2感言

巴拉巴拉``````