高级程序设计训练实验4

—实验报告

学期：2016-2017-2学期

报告人：杨宇

学号：16281237

班级：计算机1608

编制日期：2017年3月29日

目录

[一、 概述 3](#_Toc478680351)

[1. 概述 3](#_Toc478680352)

[二、 链表和链表的操作 3](#_Toc478680353)

[1. 链表组织数据的形式 3](#_Toc478680354)

[2. 链表的实现方式 3](#_Toc478680355)

[三、 排序和链表的排序 3](#_Toc478680356)

[1. 排序算法的选择 3](#_Toc478680357)

[四、 用户界面的设计 3](#_Toc478680358)

[1. 用户界面的实现 3](#_Toc478680359)

[2. 用户界面的包含关系 3](#_Toc478680360)

[五、 整体的设计思路 3](#_Toc478680361)

[六、 一些细节处理和反思 4](#_Toc478680362)

[1.程序的鲁棒性 4](#_Toc478680363)

[七、 实验结果展示 4](#_Toc478680364)

1. 概述
2. 概述

本次实验在实验3代码的基础上添加了链表的数据结构和简单的命令行用户界面。通过实现链表的插入，排序来组织学生的信息。同时通过命令行来实现简单的人机交互。

1. 链表和链表的操作
2. 链表组织数据的形式

链表是一种物理存储单元上非连续、非顺序的存储结构，数据元素通过链表结点中的指针依次链接形成一整个数据元素链。链表组织数据的特点，使得链表具有了便于插入和删除的特点。

1. 链表的实现方式

链表可以通过结构体和内嵌的结构体指针实现，也可以通过数据数组和指向数组实现。但是第二种实现方式有很多的局限性，本次实现没有采用。

1. 排序和链表的排序
2. 排序算法的选择

很多的排序算法对于链表来说并不是特别的方便。因为链表交换两个变量的难度比较大。根据链表便于插入删除的特点，链表最适合的排序方法是插入排序。通过比较数据中某一个数值的大小作为比较的依据，然后将相对较小的一个排在前面。通过链表的插入和删除操作可以方便的执行。

1. 用户界面的设计
2. 用户界面的实现

本次实验中采取了命令行用户界面，无须使用其他的图形库和框架。将整个程序放在一个死循环中，除了输入特定的值（0）进行返回上一级菜单和退出程序外，其他的操作通过（1-4）个选项来提供给用户选择。在这里，我把每一个界面用一个print\_screen()函数来写，print\_creen()函数在被调用的时候，会在屏幕中打印相关的选项，供用户选择，同时读入一个用户的选项作为返回值，在interface()函数中，直接通过用户的返回值判断下一步执行的程序。

1. 用户界面的包含关系

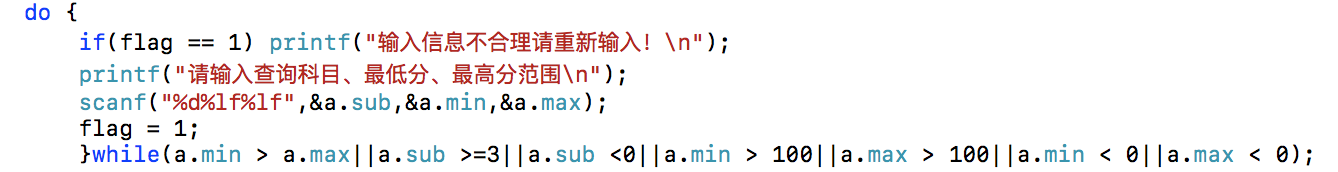
整个用户界面的实现，我放在了UI.h头文件中，在main.c中直接包含UI.h即可轻松的完成用户界面的设计。通过interface()函数组织调用相关的函数。

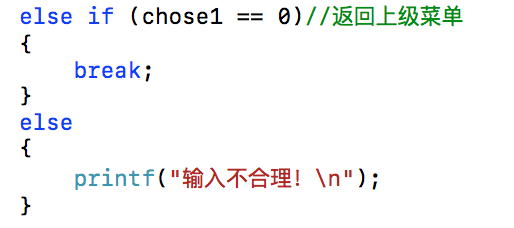
1. 整体的设计思路

分别在ClassInfo.h和UI.h头文件中实现整个程序所需要的函数功能，再通过main.c调用相依模块实现真个程序的功能。其中ClassInfo.h头文件中实现了整个程序中需要完成的数据组织和处理，UI.h中实现了用户界面。数据的组织，通过结构体CLASSINFO、CLASSINFOFILE、STUDENTINFO、CLASSSCORE CHAININFO、SEEK NODE、NODEHEAD等结构体组织学生信息。再通过相应的ReadText()、ReWriteText()、ReadTextForStructNarr()、ReadTextForChainNarr()、ReadTextForList()函数实现了读入。再通过PrintStructNarr()、ReWriteTextStruct()、ReWriteText()实现了输出，FindStuInfoStruct()、FindStuInfoPoint()、FindStuInfoList()实现了信息的查找。

1. 一些细节处理和反思

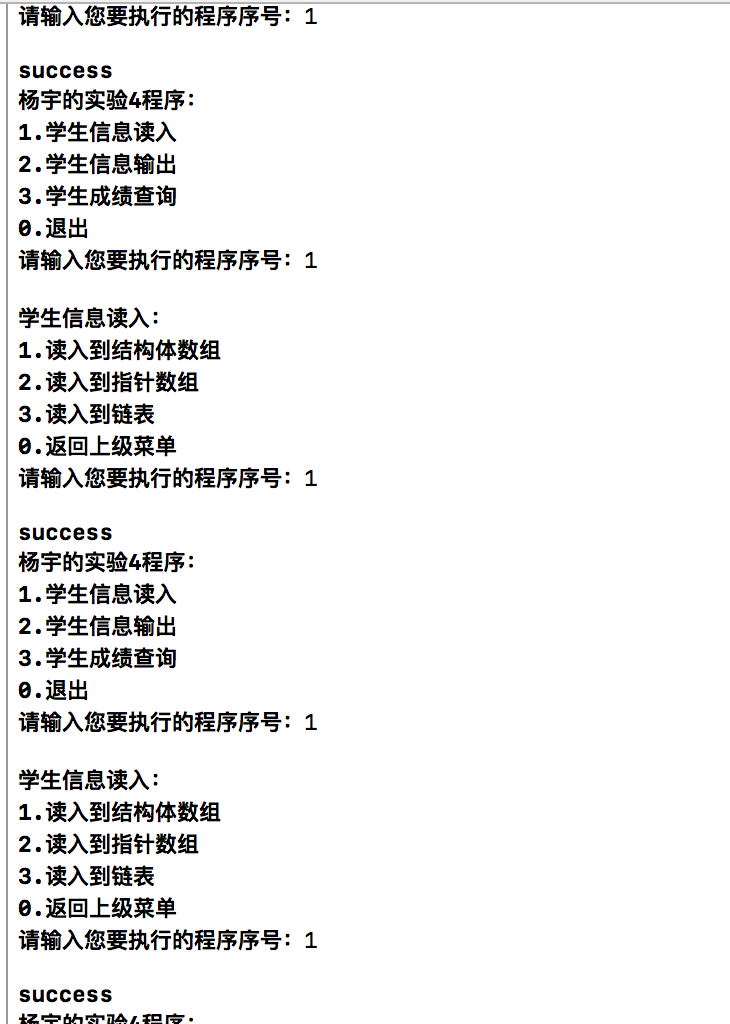
## 1.程序的鲁棒性

在实际的编程过程中，代码的容错能力不可小觑。为了提高代码的容错水平，要全面考虑每个变量的值会造成怎样的结果。

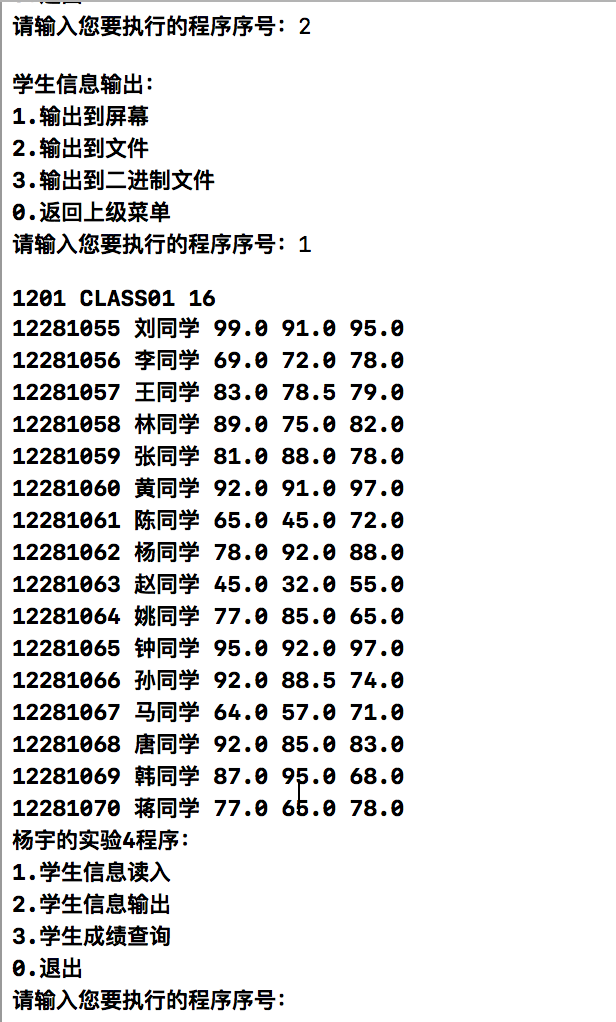


在这两处的代码设计中，我对输入不合理的情况进行了检查。避免程序因为输入不合理出现BUG。

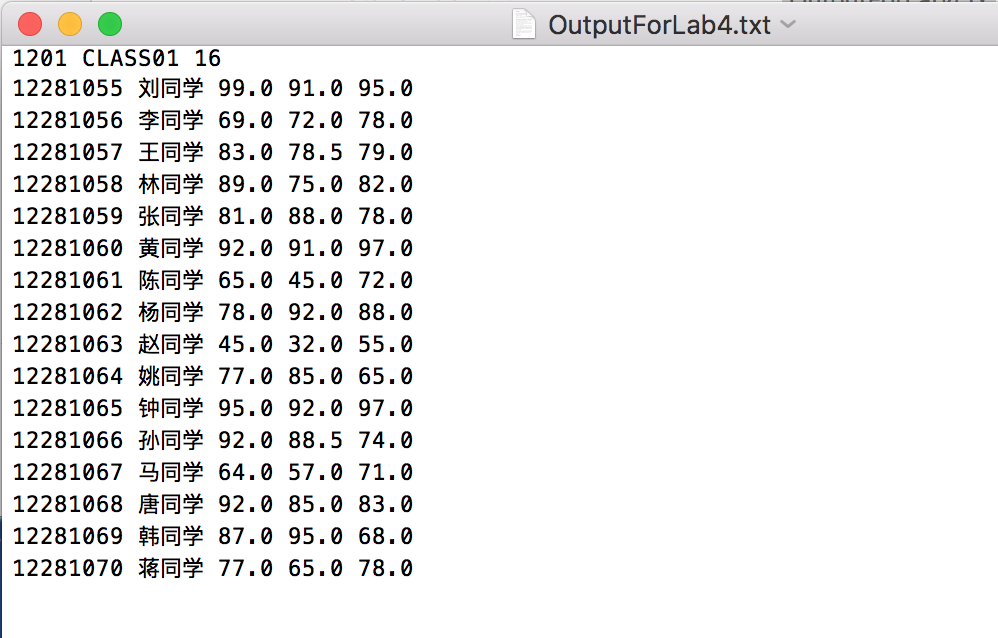
1. 实验结果展示



从文件中读取到三种数据结构。



输出到屏幕



输出到文本文件