电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2018091602012

姓 名 杨贺然

（实验） 课程名称 程序设计与算法基础 II

理论教师 郝宗波

实验教师 郝宗波

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：杨贺然 学号：2018091602012 指导教师：郝宗波**

**实验地点：基础实验大楼 A508 实验时间：2019.04.19**

**一、实验室名称：学校实验中心软件实验室**

**二、实验名称：学生课程成绩查询程序**

**三、实验学时：8 课时**

**四、实验原理：**

链表是一种物理存储单元上非连续、非顺序的存储结构，数据元素的逻辑顺序是通过链表中的指针链接次序实现的。链表由一系列结点（链表中每一个元素称为结点）组成，结点可以在运行时动态生成。每个结点包括两个部分：一个是存储数据元素的数据域，另一个是存储下一个结点地址的指针域。由于其可扩展性高，适用于这样需要动态添加元素的操作，因此采用链表作为数据存储方式。

栈是一种先进先出的数据结构，队列是一种先进后出的数据结构，利用栈可以实现链表反向，利用队列可以动态管理链表中元素。

**五、实验目的**

通过本实验练习，掌握需要包括:磁盘文件的读写方法；掌握链表创建方法，掌握链表节点的数据结构定义、链表的排序方法、两个链表内容链接查询的方法；掌握创建栈、进栈和出栈的实现方法。

**六、实验内容：**

设有学生信息文件 student.dat，每个学生记录包括：学号 sno、姓名 sname、性别 sex、专业 major；课程信息文件 course.dat，课程记录包括：课程号 cno、课程名称 cname、课时数 classHours；课程成绩信息文件 courseGrade.dat，成绩记录包括学号 sno、课程号 cno、考试成绩 score；自学教材 61-64 页线性表的应用，设计应用程序完成如下功能，同时需要设计程序运行的功能菜单：

1. 输入 10 条学生记录，其中软件技术专业 5 人，人工智能专业 5 人，并存入文件 student.dat 中；
2. 输入 3 门课程（数据库、数据结构、程序设计）信息记录，并存入文件 course.dat 中；
3. 输入上述 10 位同学分别选修上述三门课程的考试成绩到文件 courseGrade.dat 中；
4. 从文件 student.dat 中读出学生信息，生成按照学号升序排列的单向链表，并在屏幕上显示输出；
5. 从文件 course.dat 中读出课程信息，生成按照课程号升序排列的单向链表，并在屏幕上显示输出；
6. 从文件 courseGrade.dat 中读出成绩信息，生成按照学号和课程号升序排列的单链表，并在屏幕上显示输出；
7. 查询所有学生所有课程的考试成绩，生成该课程的成绩单链表，要求包括学号、学生姓名、专业、课程名、考试成绩等信息，按照考试成绩降序排列，并将学生的该成绩信息输出到文件 studentGrade.dat 中，同时在屏幕上显示输出；
8. 在（7）的链表中，查询指定课程号的所有学生的考试成绩，生成该课程的成绩单链表，要求包括学号、学生姓名、专业、课程名、考试成绩等信息，按照考试成绩降序排列输出到屏幕上显示；
9. 在（7）的链表中，查询指定课程号的考试成绩小于 60 分的学生成绩信息，生成该课程的成绩链表，要求包括学号、学生姓名、专业、课程名、考试成绩等信息，并按照考试成绩降序排列在屏幕上显示输出；
10. 使用栈实现将（4）的单链表中的学生信息逆序生存新的链表；
11. 设计使用链式队列完成问题（7）的要求。

**七、实验器材（设备、元器件）：**

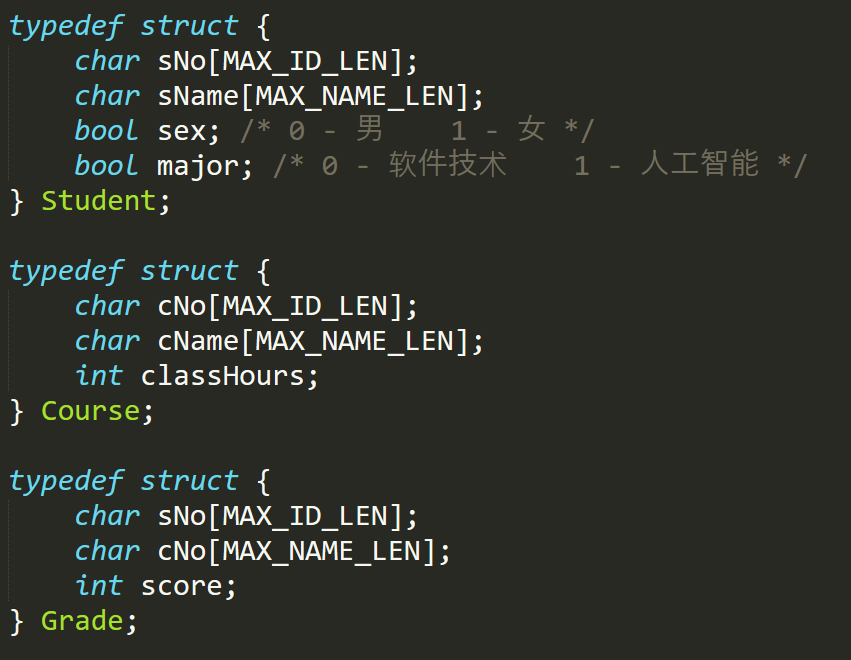
个人电脑一台。

**八、数据结构与程序：**

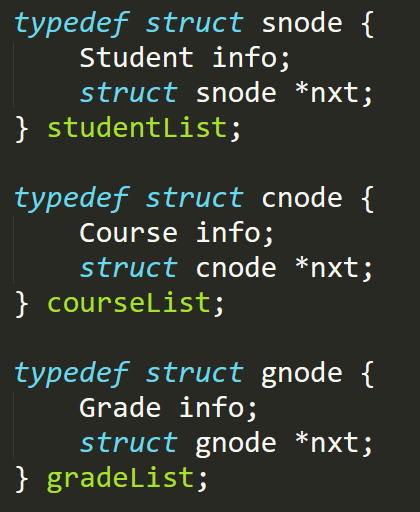
1. **明确学生与课程属性，定义结构与链表结构**

学生信息包括学生姓名，学号，性别，专业；课程信息包括课程号，课程名称和学时；成绩信息包括学号，课程号，成绩。其中用字符数组保存学生姓名，学号，课程号，课程名称，用布尔型保存学生性别和专业，用整数型保存学时数与成绩。链表结构包括学生信息，课程信息或成绩信息与下一节点指针。

学生信息，课程信息与成绩信息结构如下：

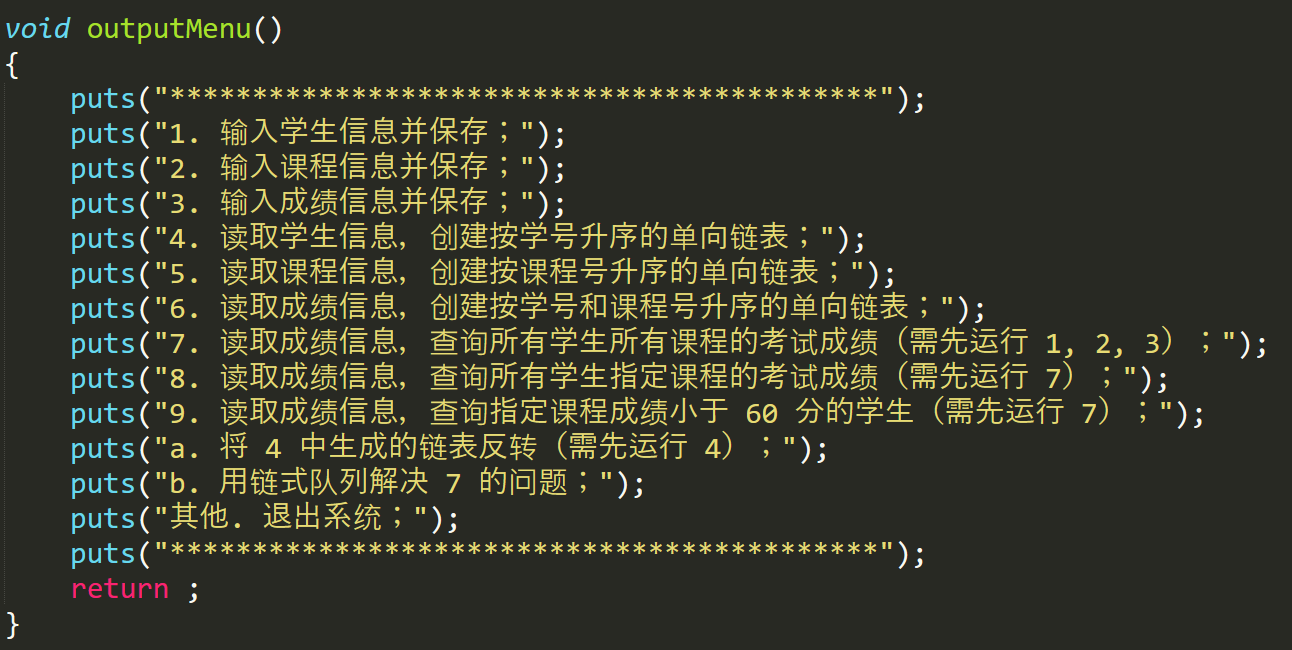


链表结构如下：



1. **实现主程序部分**

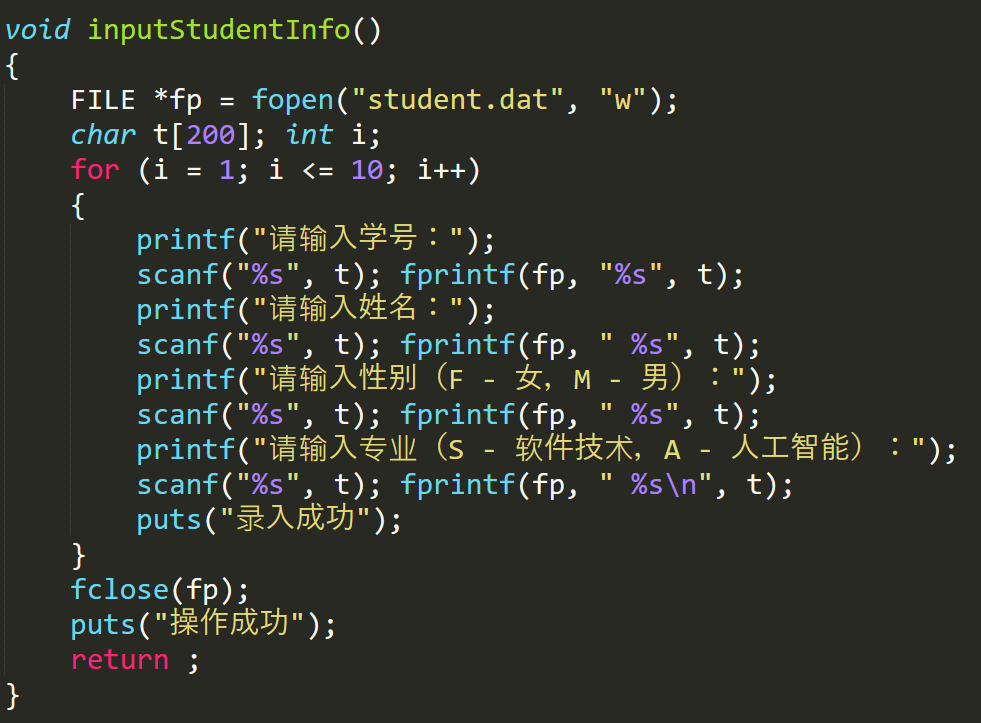
主程序部分需要输出菜单，接收用户选择。这里用标准输入输出与用户交互。实现如下：



1. **实现创建学生信息链表功能**

该函数需要从标准输入中读入学生的所有信息，然后创建链表节点存储，并插入学生信息的链表中。最后需要把学生的信息全部输出到文件中。

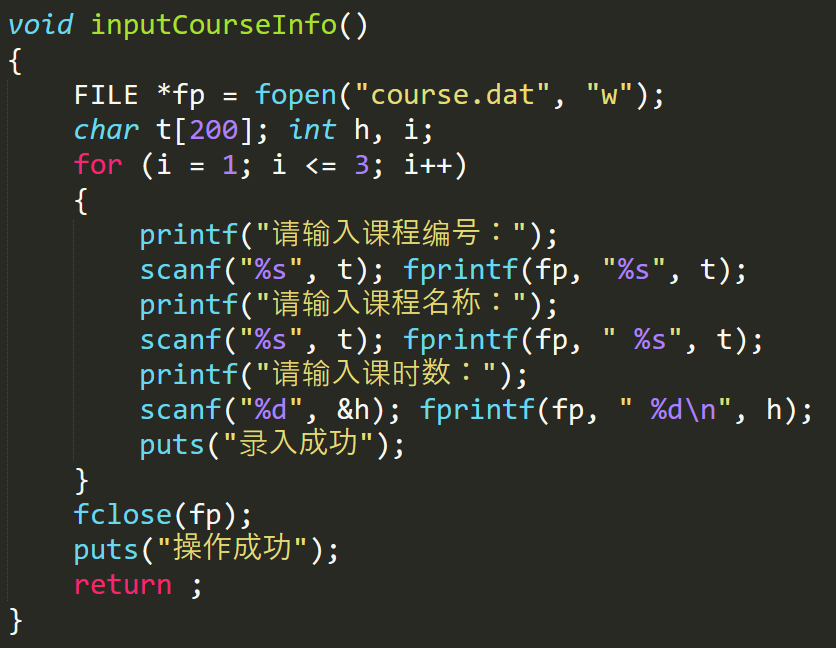
详细实现如下：



1. **实现创建课程信息链表功能**

该函数需要从标准输入中读入课程的所有信息，然后创建链表节点存储，并插入课程信息的链表中。最后需要把课程的信息全部输出到文件中。

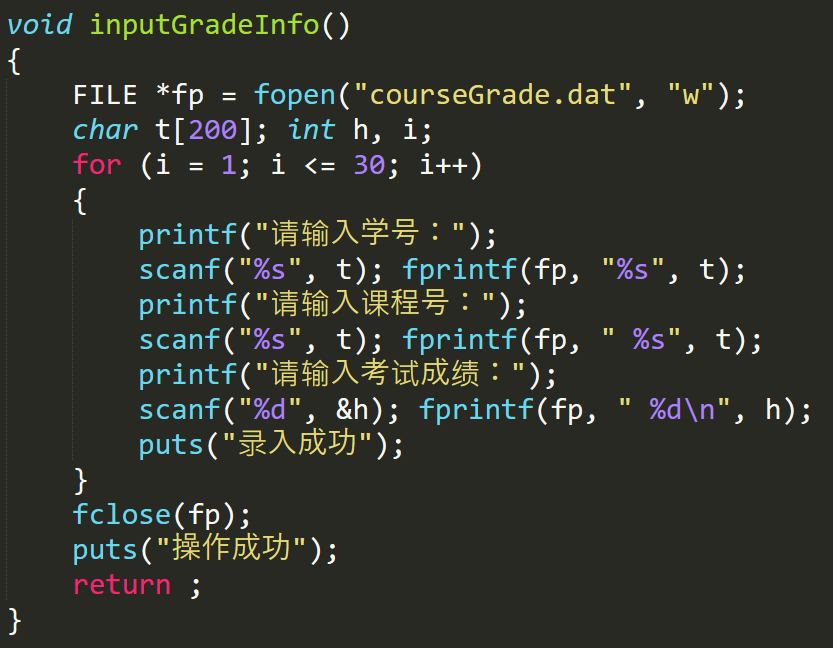
详细实现如下：



1. **实现创建成绩信息链表功能**

该函数需要从标准输入中读入成绩的所有信息，然后创建链表节点存储，并插入成绩信息的链表中。最后需要把成绩的信息全部输出到文件中。

详细实现如下：

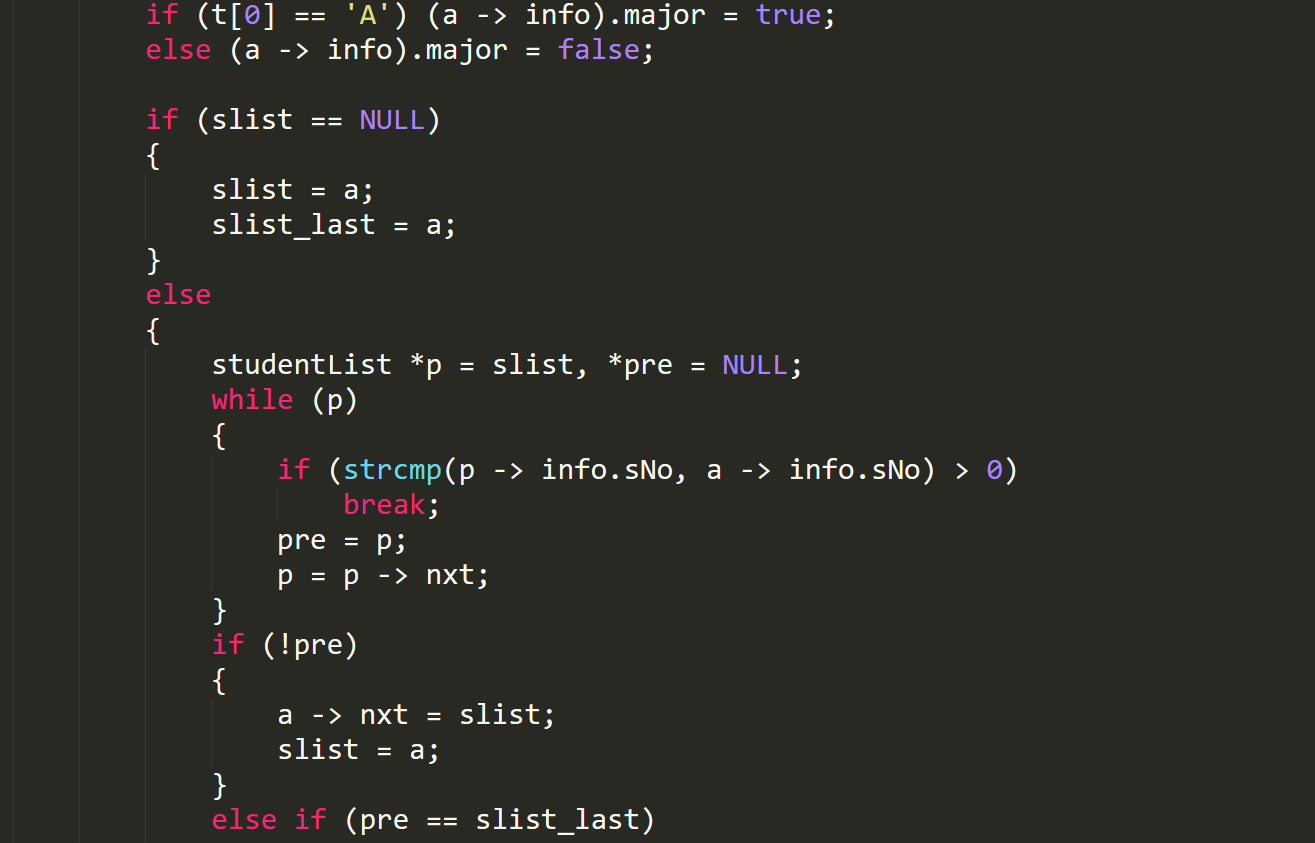


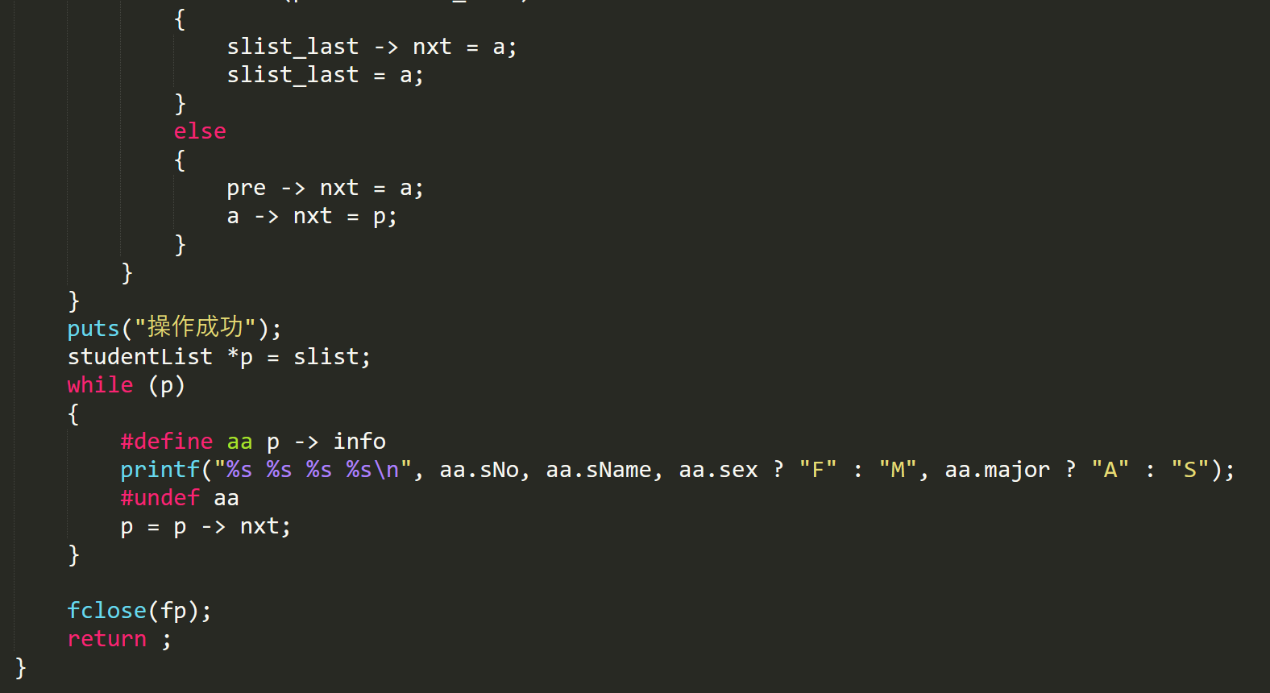
1. **实现读入学生信息并按学号排序的功能**

该函数需要从文件读入学生信息，并将学生信息按学号排序后输出到屏幕。每次读取一条学生信息记录，都将它保存到一个链表节点中。从头遍历链表，直到找到第一个学号比这条记录大位置，并将其插入到这个位置前面。最后从头开始输出链表中元素。

详细实现如下：





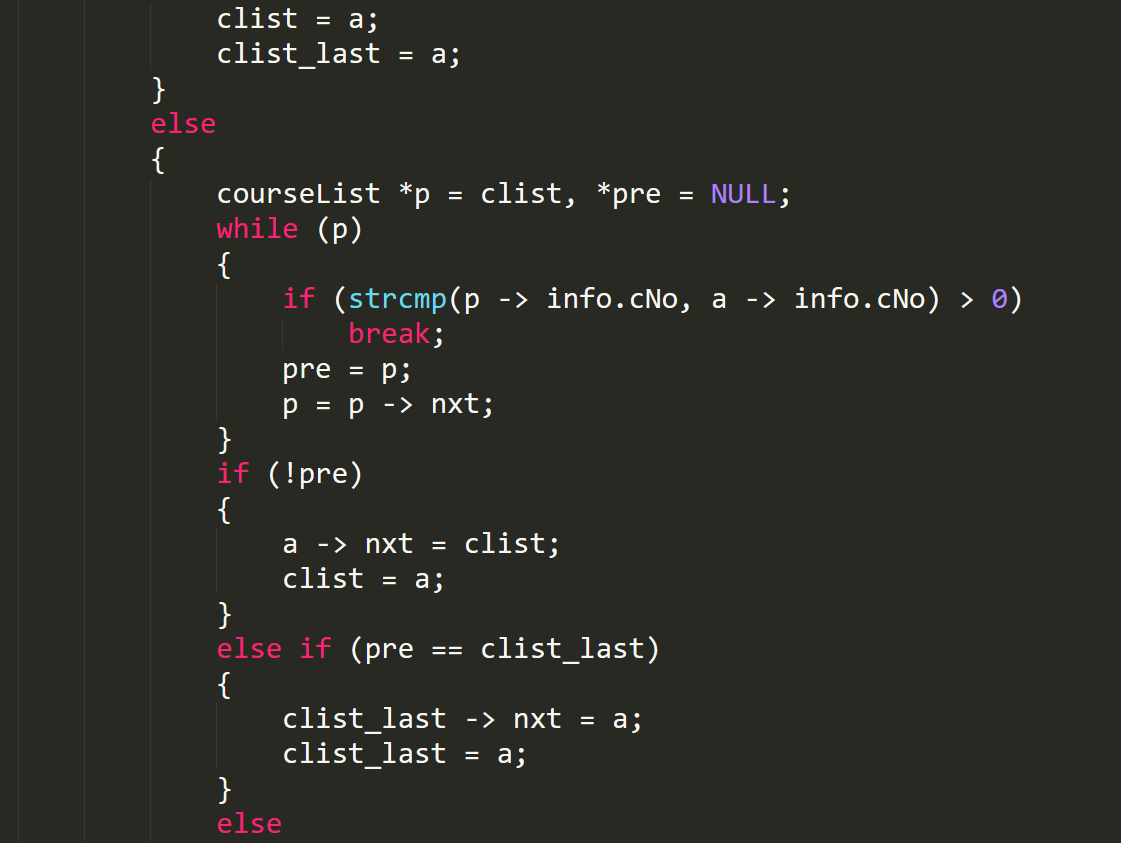


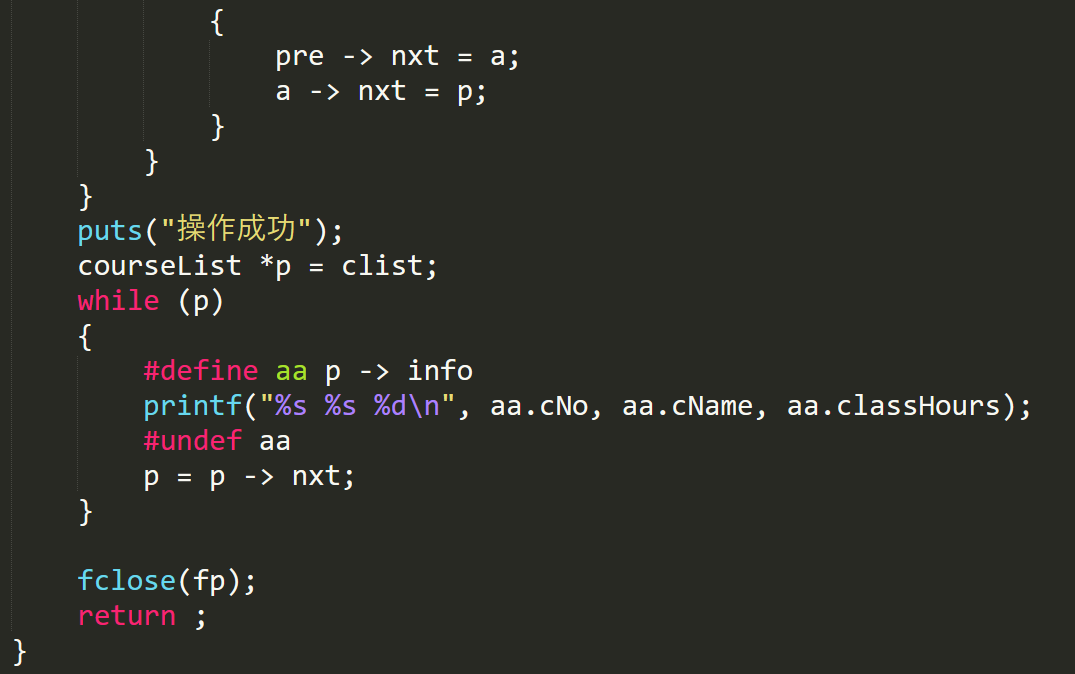
1. **实现读入课程信息并按课程号排序功能**

该函数需要从文件读入课程信息，并将课程信息按课程号排序后输出到屏幕。每次读取一条课程信息记录，都将它保存到一个链表节点中。从头遍历链表，直到找到第一个课程号比这条记录大位置，并将其插入到这个位置前面。最后从头开始输出链表中元素。

详细实现详见附件 studentsInfo.c 中 void rwGenerateCourseLink()。





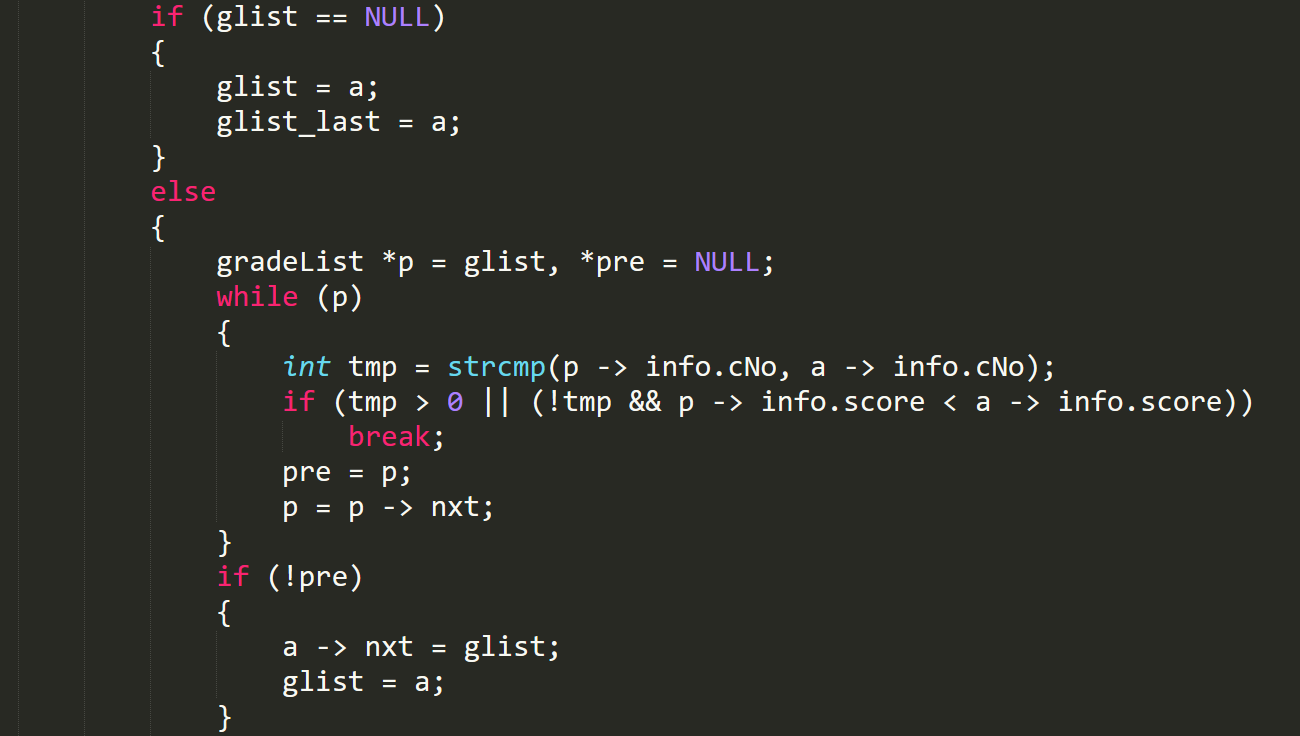


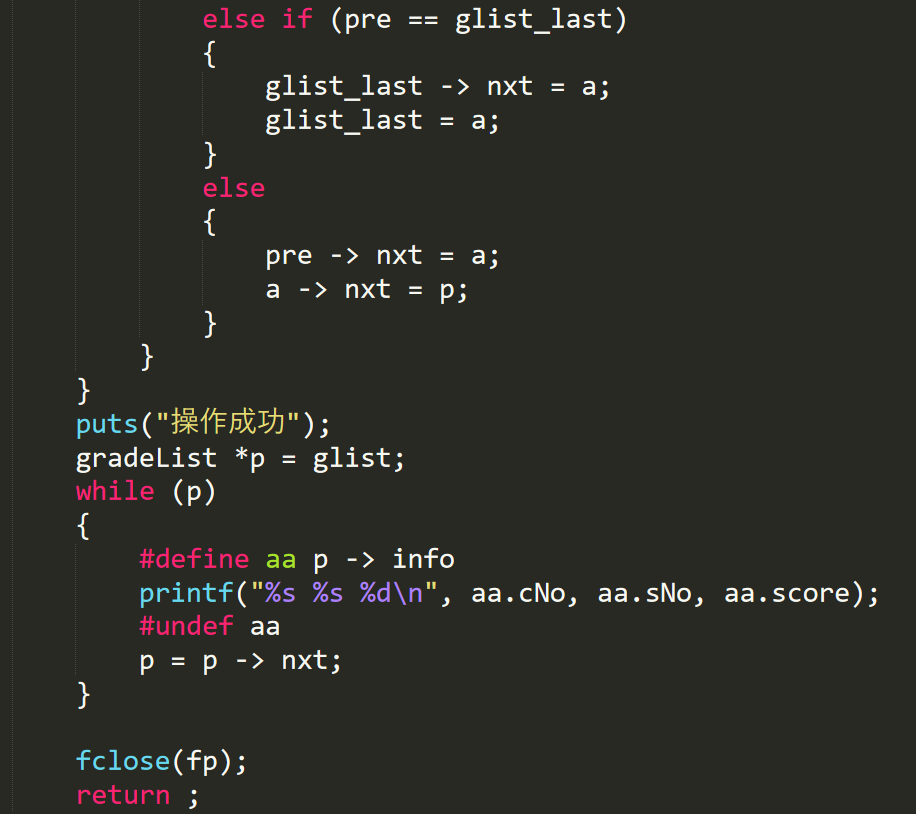
1. **实现读入成绩信息并按学号与课程号排序功能**

该函数需要从文件读入成绩信息，并将成绩信息按学号与课程号排序后输出到屏幕。每次读取一条成绩信息记录，都将它保存到一个链表节点中。以课程号为第一关键字，学号为第二关键字，从头遍历链表，直到找到第一个课程号比这条记录大，或课程号相同且第一个学号比这条记录大的位置，将这条记录插入至该位置前面。最后从头开始输出链表中元素。

详细实现详见附件 studentsInfo.c 中 void rwGenerateGradeLink()。







1. **实现生成成绩单链表功能**

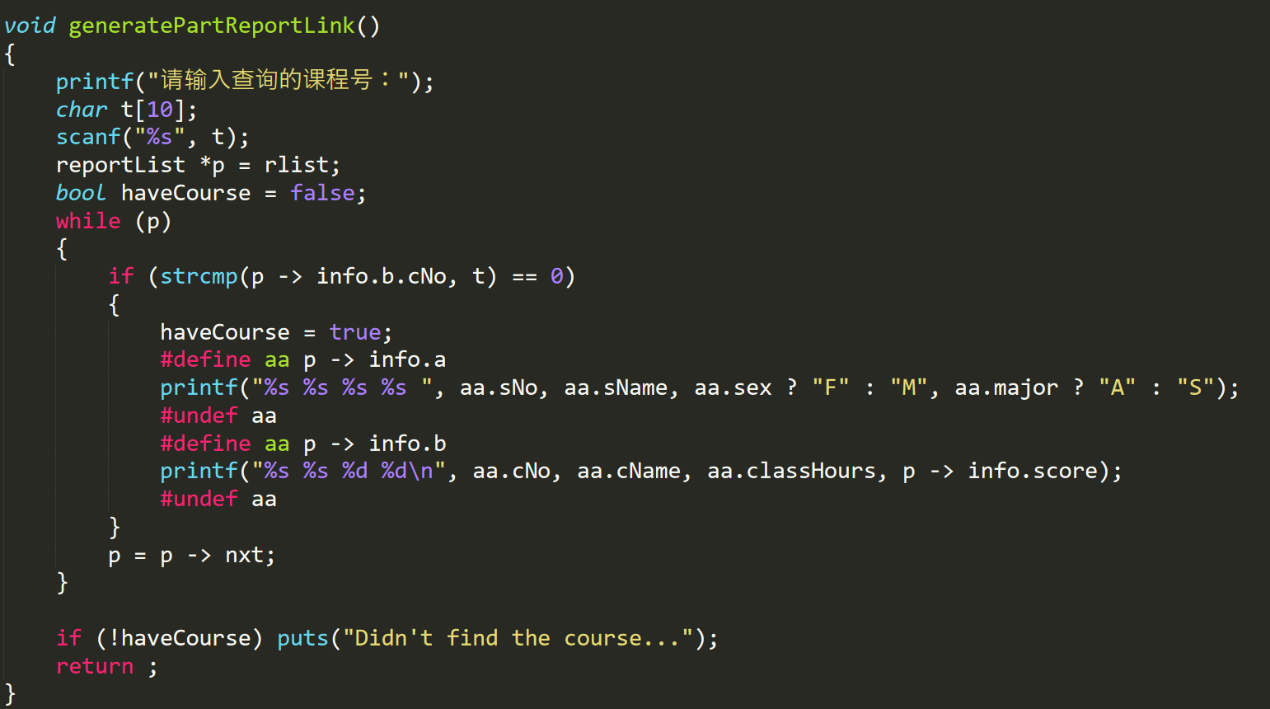
该函数利用（4），（5），（6）中生成的链表生成成绩单链表。先从（6）中生成的链表获取学号，课程号和成绩信息，之后从（4），（5）中链表把信息补全。最后以课程号为第一关键字，成绩为第二关键字，从头遍历链表，直到找到第一个课程号比这条记录大，或课程号相同且第一个成绩比这条记录小的位置，将这条记录插入至该位置前面。最后从头开始输出链表中元素，注意要输出到屏幕和文件。

详细实现详见附件 studentsInfo.c 中 void generateReportLink()。

1. **实现查询指定课程成绩成绩单功能**

该函数利用（7）中生成的链表生成成绩单链表。遍历链表中的每一个元素，将课程号与查询的课程号相同的记录输出。

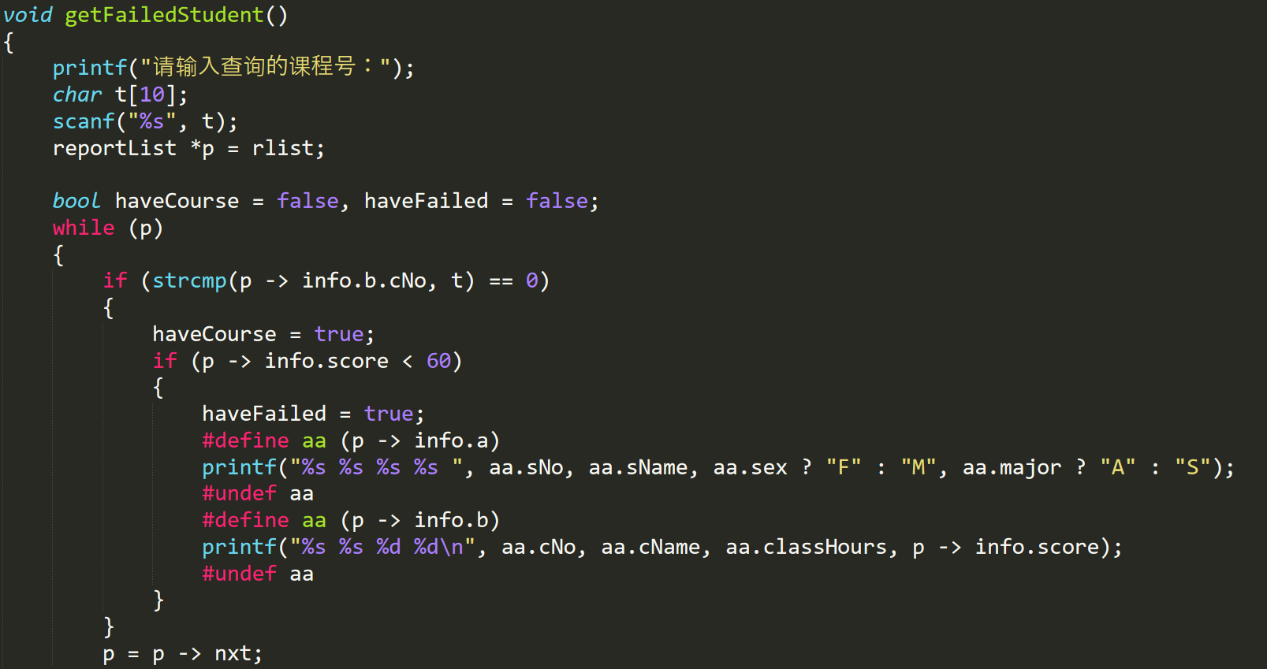
详细实现如下：

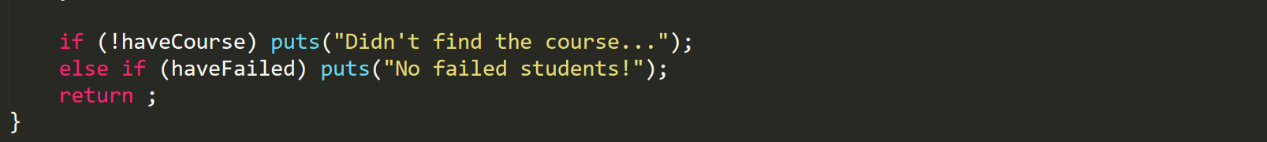


1. **实现查询不及格学生记录功能**

该函数利用（7）中生成的链表生成成绩单链表。遍历链表中的每一个元素，如果这条记录的成绩小于 60，则输出这条记录。

详细实现如下：





1. **实现学生信息逆序存储功能**

该函数利用学生信息链表，将链表中元素依次全部压入栈中，然后依次弹出元素，将元素按弹出先后次序插入链表尾部。

详细实现详见附件 studentsInfo.c 中 void reverseStudentLink()。

1. **实现用队列管理成绩信息功能**

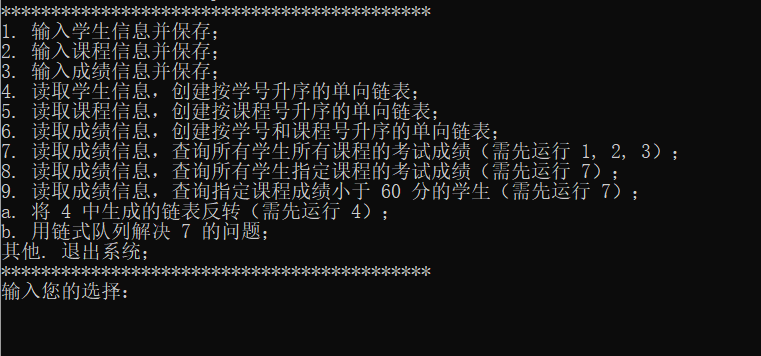
该函数利用成绩信息链表，将链表中元素依次插入队列。再从队列中弹出。

详细实现详见附件 studentsInfo.c 中 void generateReportLink\_queue()。

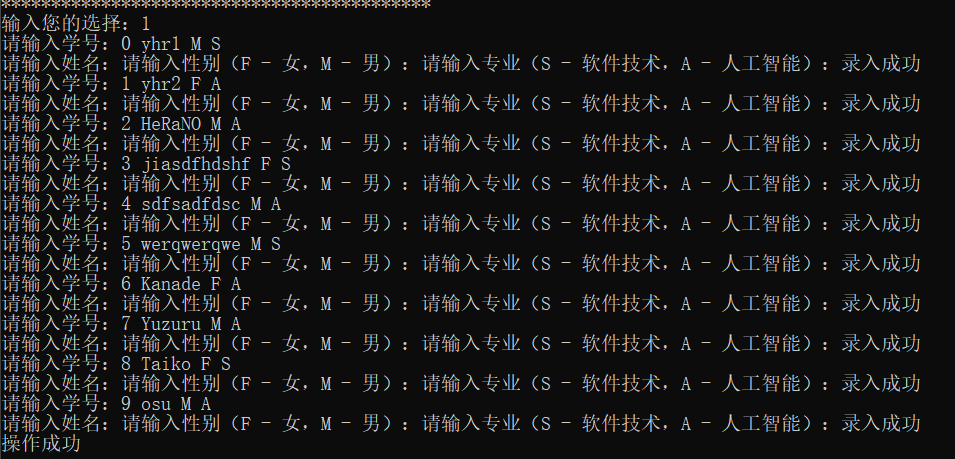
1. **反复调试直到程序可以正常实现所有功能**

**九、程序运行结果**

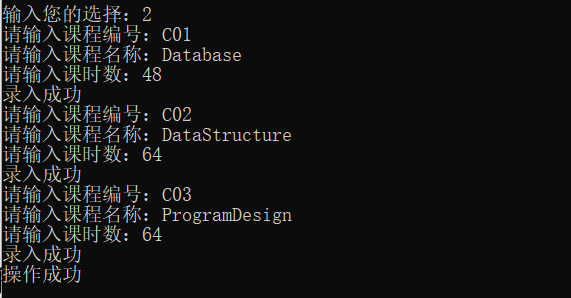
1. **主界面实现**



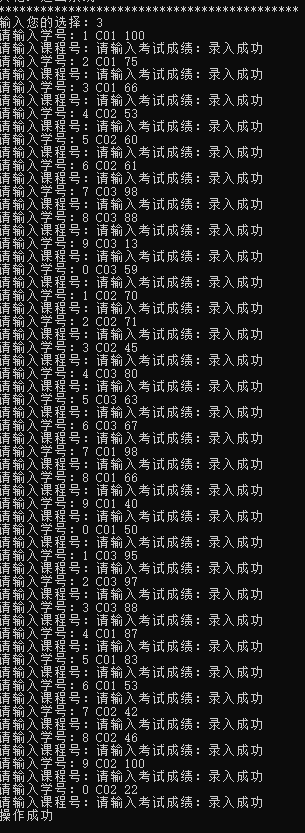
1. **输入学生信息并保存**



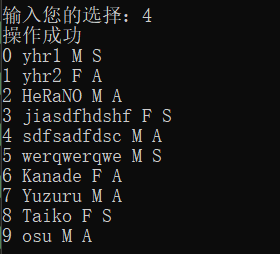
1. **输入课程信息并保存**



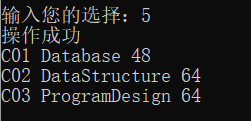
1. **输入成绩信息并保存**



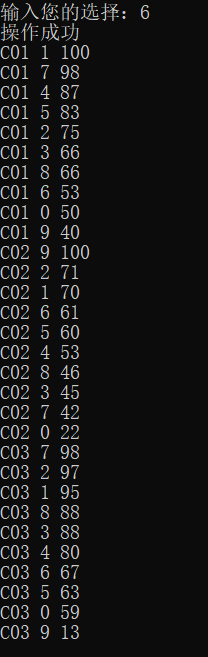
1. **读取学生信息，创建按学号升序的单向链表**



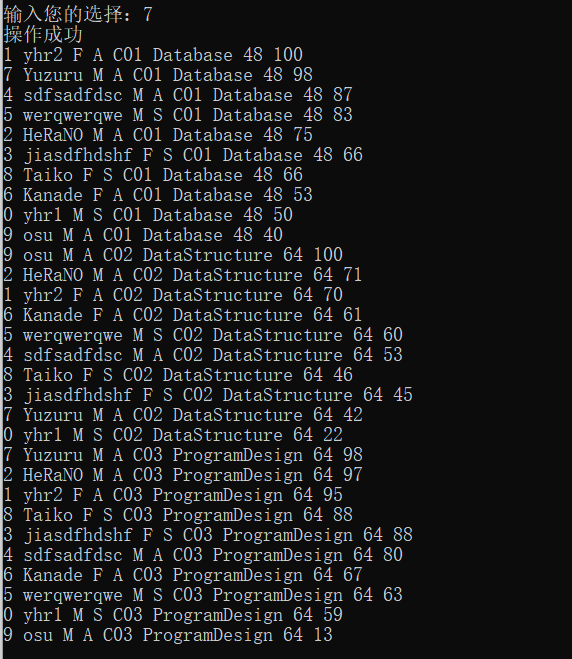
1. **读取课程信息，创建按课程号升序的单向链表**



1. **读取成绩信息，创建按学号和课程号升序的单向链表**

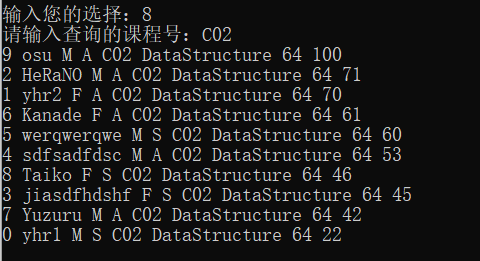


1. **读取成绩信息，查询所有学生所有课程的考试成绩**

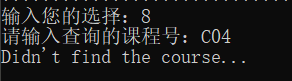


1. **读取成绩信息，查询所有学生指定课程的考试成绩**

当课程存在时：

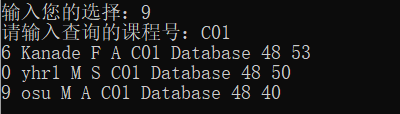


当课程不存在时：

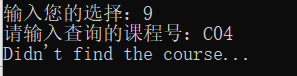


1. **读取成绩信息，查询指定课程成绩小于 60 分的学生**

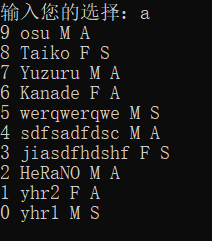
当课程存在时：



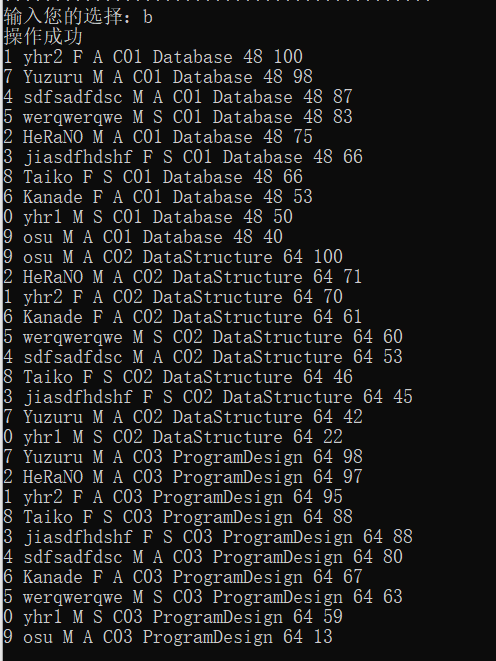
当课程不存在时：



1. **将 4 中生成的链表反转**



1. **用链式队列解决 7 的问题**



**十、实验结论**

对于这种学生信息管理需求，可以用链表实现，对于将链表反向，可以利用栈先进后出的性质实现。

**十一、总结及心得体会**

经过本实验，对学生成绩信息管理系统，链表，栈和队列有了更深刻的了解，锻炼了代码编写与调试能力。将功能写成函数既能简化程序，又能增加程序可读性，便于调试与日后维护。