C++程序设计 课程设计报告

**小组成员：**

刘文越（学号：2021902610）（班级号：2021240207）

胡云皓（学号：2020900210）（班级号：2020240208）

**课设主题：**

学生成绩管理系统

**主要实现功能：**

1. 实现学生记录的录入
2. 实现学生记录的删除
3. 实现学生记录的更新
4. 实现学生记录的查询
5. 显示全体学生记录
6. 实现将学生成绩分段呈现，并统计对应分数段人数
7. 实现按照学生成绩平均分的顺序逆序排序
8. 清空全部学生记录
9. 将学生记录以txt文件的形式保存在硬盘中，实现长期储存。

**实现方式：**

1. 学生记录的录入是链表节点的插入操作，在键盘上键入对应学生信息后，使用自己构造的链表类InsertList函数开辟对应的内存空间，并定义指针指向该内存空间，遍历到链表尾插入。
2. 学生记录的删除是链表节点的删除操作，在寻找到对应的学生记录节点后，将上个节点的下标改到该学生节点的下一个节点，最后释放该内存空间，实现删除操作。
3. 学生记录的更新操作是将学生信息重新录入，寻找到对应学生记录，重新赋值变量实现。
4. 学生记录的查询操作，是寻找到对应的学生记录后，直接将该学生记录信息输出并返回。
5. 显示全体学生记录操作，是从链表头一直遍历到链表尾，依次输出学生记录。
6. 分段呈现过程中，使用判断语句，依次将对应分数段的学生记录输出。
7. 按照学生成绩平均分排序，是通过冒泡排序实现的，但是冒泡排序中的两层指针用链表指针替换。
8. 清空学生记录，是将链表头的下标直接指向链表尾，并且释放原来链表中全部内存空间实现的。
9. 储存学生记录通过输入输出流函数实现。

**实现过程：**

在一开始先采用了数组的方式，在实现了部分功能后，意识到数组无法调整学生的上限，即数组只能开辟自己规定大小的内存空间，并且在删除和修改上，数组存在明显的时间消耗，在搜集资料后发现有链表这种数据结构，可以实现内存空间按需开辟，并且可以方便的进行删除修改操作。综上所述，链表相比于数组具有以下优点：

1. 插入和删除的速度快，保留原有的物理顺序，在插入或者删除一个元素的时候，只需要改变指针指向即可。
2. 没有空间限制，存储元素无上限，只与内存空间大小有关，数组的存储需要实现开辟内存，且数组有容量上限，操作不当或数据量过大可能引起内存溢出。
3. 动态分配内存，不用实现开辟内存，同点2所述。
4. 内存利用率变高，链表的大小可以在运行的时候增加或减少，因此不会浪费内存，相比数组的存储方式，如果我们声明一个大小为10的数组并在其中只存储6个元素，那么就会浪费4个元素的空间。
5. 使用链表可以轻松实现栈、堆、队列等数据结构，便于后期功能的维护和增加。

（时间复杂度：查找操作为O(n) ,插入和删除操作为O(1)。）

后来在实现链表的过程中，发现在链表节点中定义过多元素并不方便操作，于是定义学生结构体后，再单独定义链表节点结构体。

（学生结构体：

typedef struct \_STRU\_STU\_SCORE\_ {

    unsigned int nStuId;

    string cpName;

    string sex;

    double nScoreComputing;

    double nScoreMath;

    double nScoreEnglish;

    double nScoreAver;

} STRU\_STU\_SCORE;

链表节点结构体：

struct NODE {

    \_STRU\_STU\_SCORE\_ data;

    NODE\* next;

};

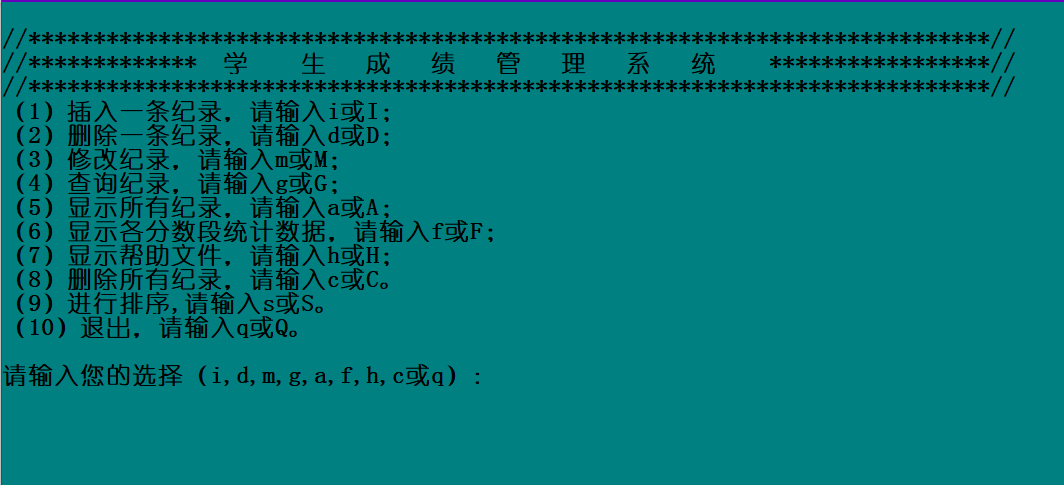
）

在此基础上可以方便构建链表类的函数，同时方便的按结构体形式存储学生数据，且对应节点的内存空间大小一致。

在实现全部功能后开始着手优化设计，通过搜集资料，寻找到getche函数，该函数用于输入后立即从控制台取字符，以实现用户在面板键入对应操作按键时立即得到反馈，并且保证只能输入一个字符，增加了设计的稳健性。

至此，该程序设计已经以链表形式完成所有功能的实现。

最终面板效果示意图：



**改进空间：**

在排序模块中，冒泡排序算法效率过低，可以改用快速排序或者归并排序的方式。希望在以后学习后可以进一步提高该程序的算法效率。

**收获及心得：**

在该程序设计课题的实现过程中，通过编写程序，小组讨论的形式提高了自己的编程能力和解决问题的能力，在学习了有关面向对象和数据结构的有关知识后能够熟练掌握并且运用，收获颇丰。