|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | 2106410215 | **成绩** |  |





**课程设计说明书**

**设计名称 C++课程设计**

**设计题目 学生成绩管理系统**

**设计时间 2022.6.6—2022.6.10**

**学 院 信息与控制工程学院**

**专 业 计算机科学与技术**

**班 级 计算机2102**

**姓 名 李东志**

**指导教师 师金钢 宋晓宇**

**2022 年 6 月 6 日**

C++课程设计说明书

**目录**

**1.设计任务说明**

功能：学生成绩管理系统，每个学生是一个记录，包括学号、姓名、性别、3 门课程成绩。

系统要求实现以下功能：

1、信息录入：录入学生成绩信息（包括学生学号、姓名、各门课程的成绩等），并将信息写入文件中，要求有文件读写操作；

2、信息查询：输入学号，查询学生各门课程的成绩，并显示；

3、排序：按各门课程的成绩平均分进行排序，并显示；

4、信息删除与修改——输入学号，修改或删除该学生的成绩信息

**2.开发工具**

Visual Studio2022

**3.设计思路**

3.1输出，查找

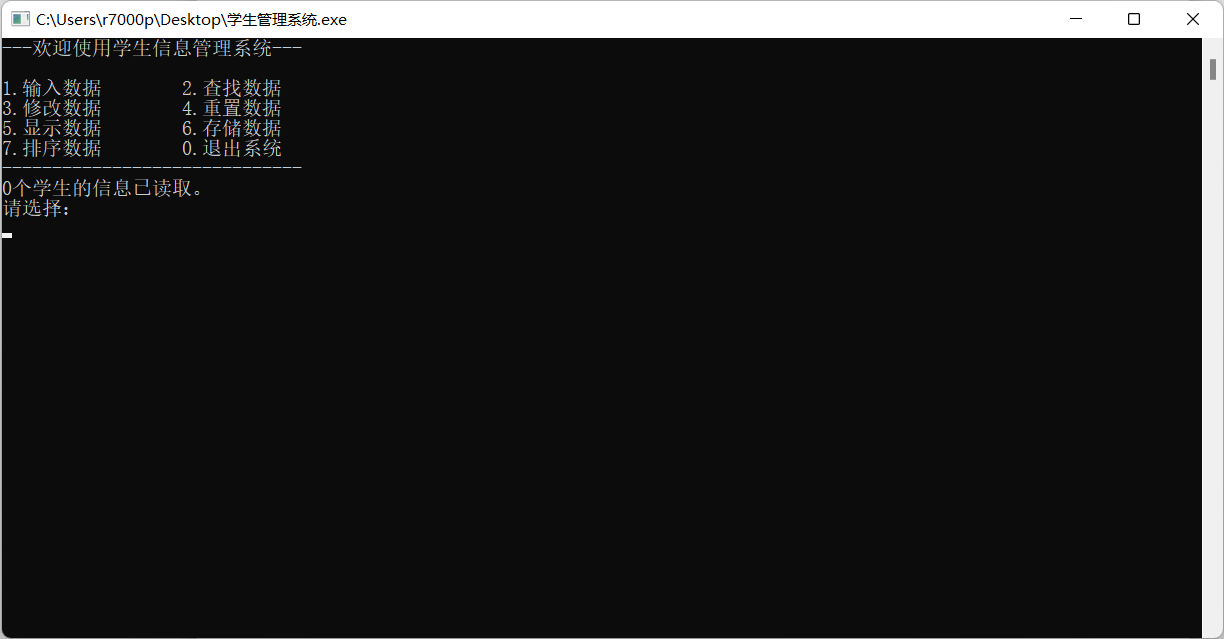
1. 按班级输出  
   2）按科目（数学，英语，c++的任意组合），按限制条件的输出  
   限制条件有：各科及平均分，总分的上下限。  
   3）输入学号查找学生，按设定的条件显示信息。  
   4）直接输出全部信息

3.2.输入  
 1）于指定位置插入任意个学生的数据  
 2）于结尾处添加数据  
 3.3.修改/删除  
 1）输入学号查找，找到之后输入要改的内容，不能手动改平均分和总分，其他的可以。  
 2）按学号查找后删除该学生的数据  
 3）清空所有数据

3.4. 排序数据  
 1）按照各科成绩，平均分或总分进行排序，从高到低或从低到高。  
 3.5. 存储/读取数据  
 会把信息存到当前目录一个叫student\_data的txt文件里，并且可以读取。

**4. 系统效果图**

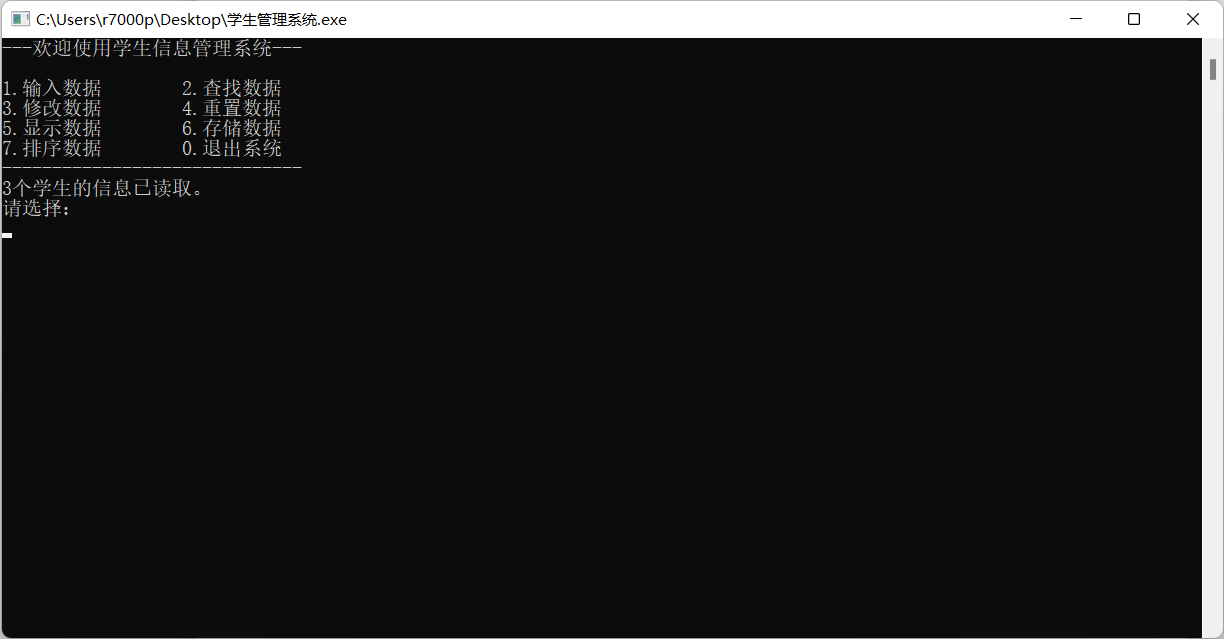
主页面：



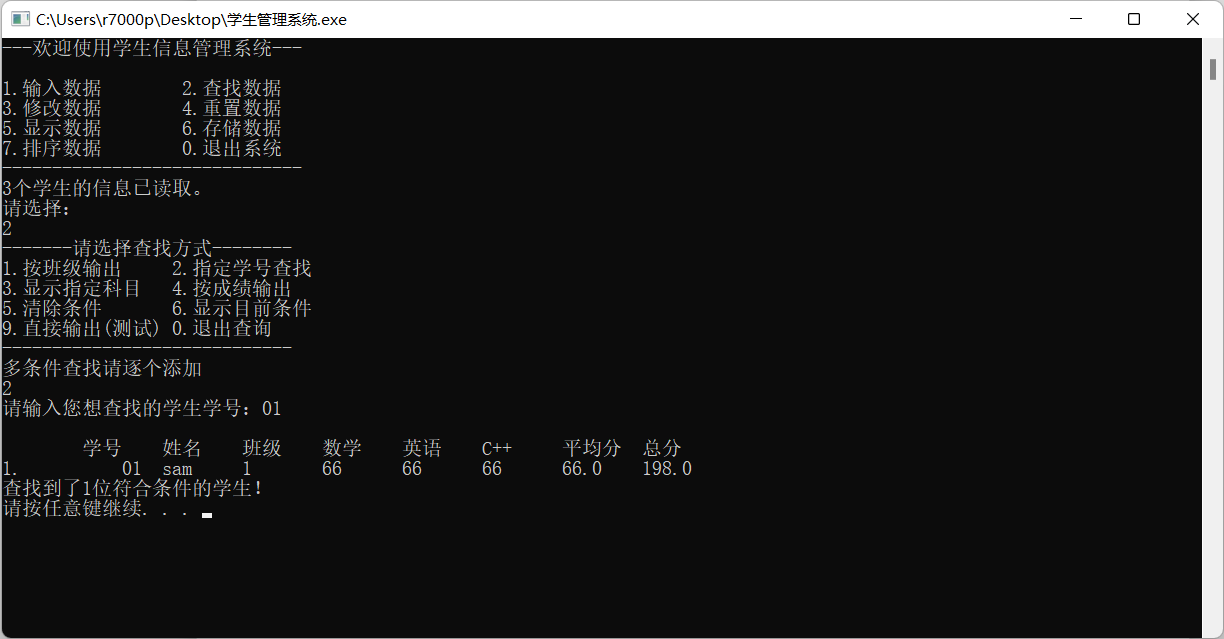
添加学生：



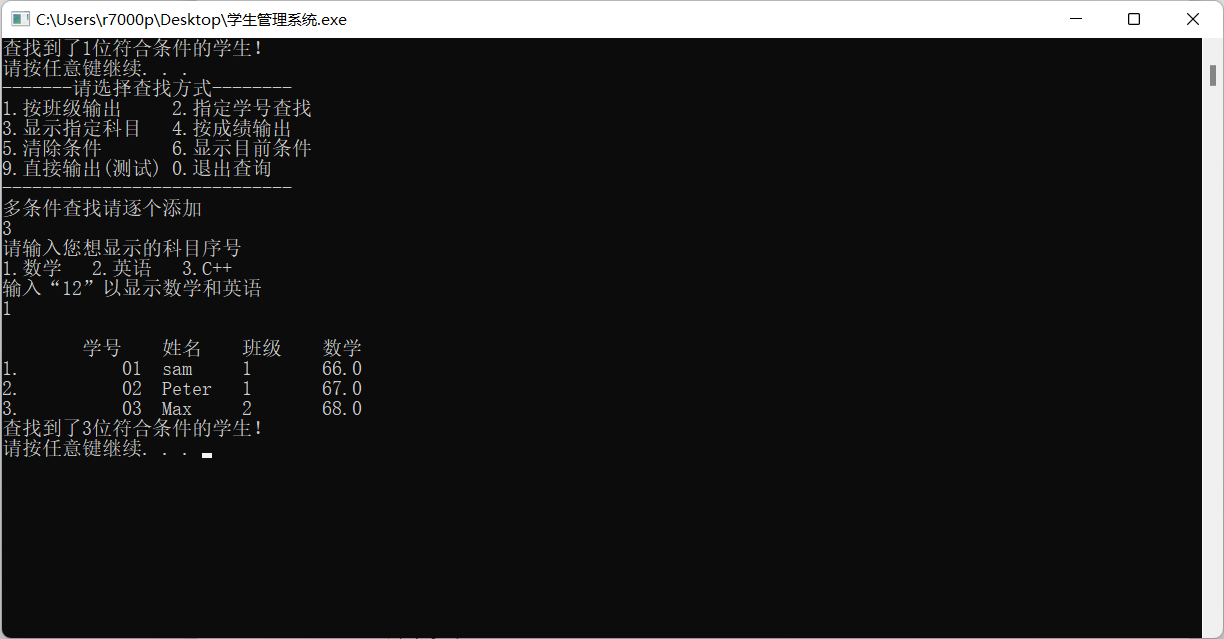
信息已读取：



输入学号查询：



按科目查询：



显示数据：



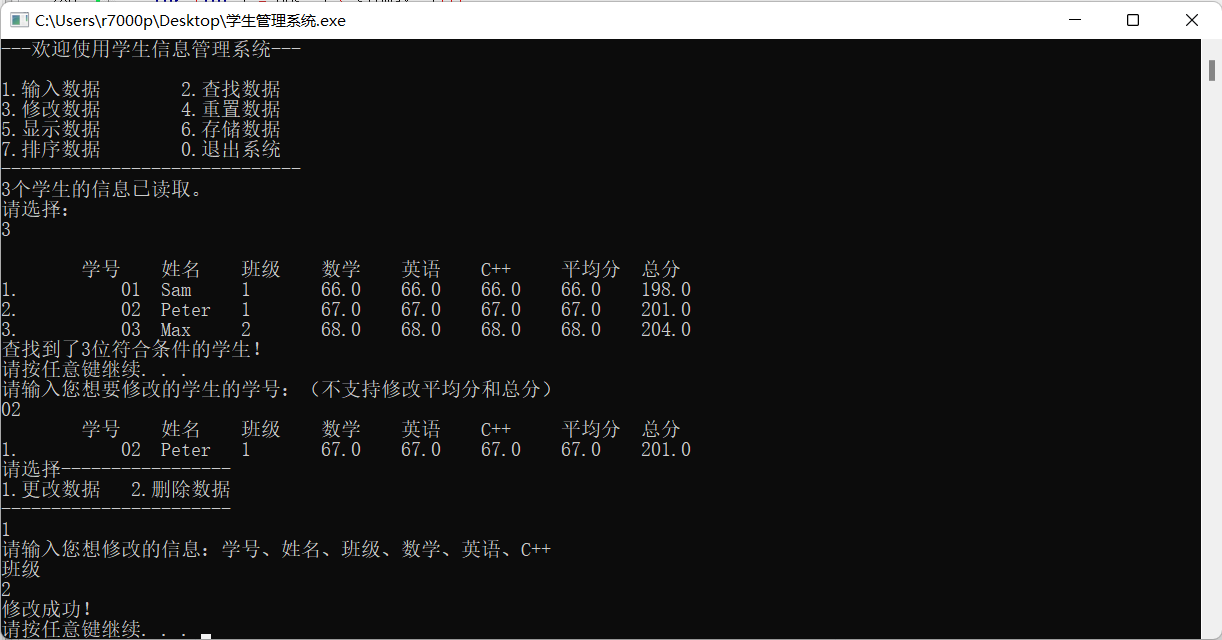
排序：



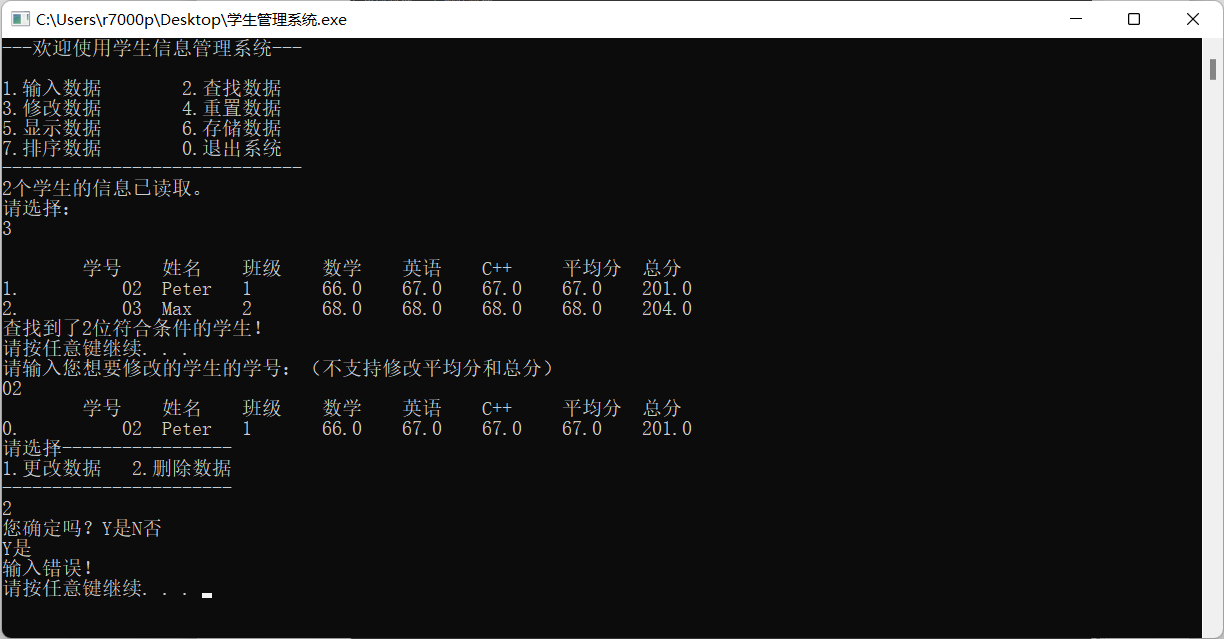
排序结果：



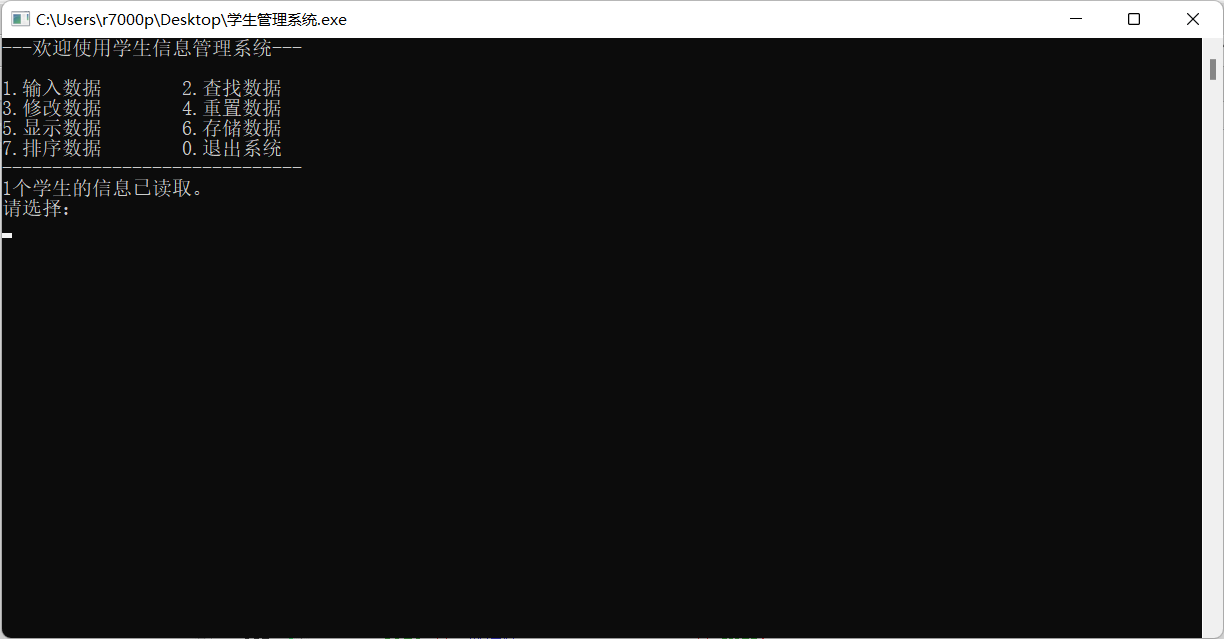
修改数据：



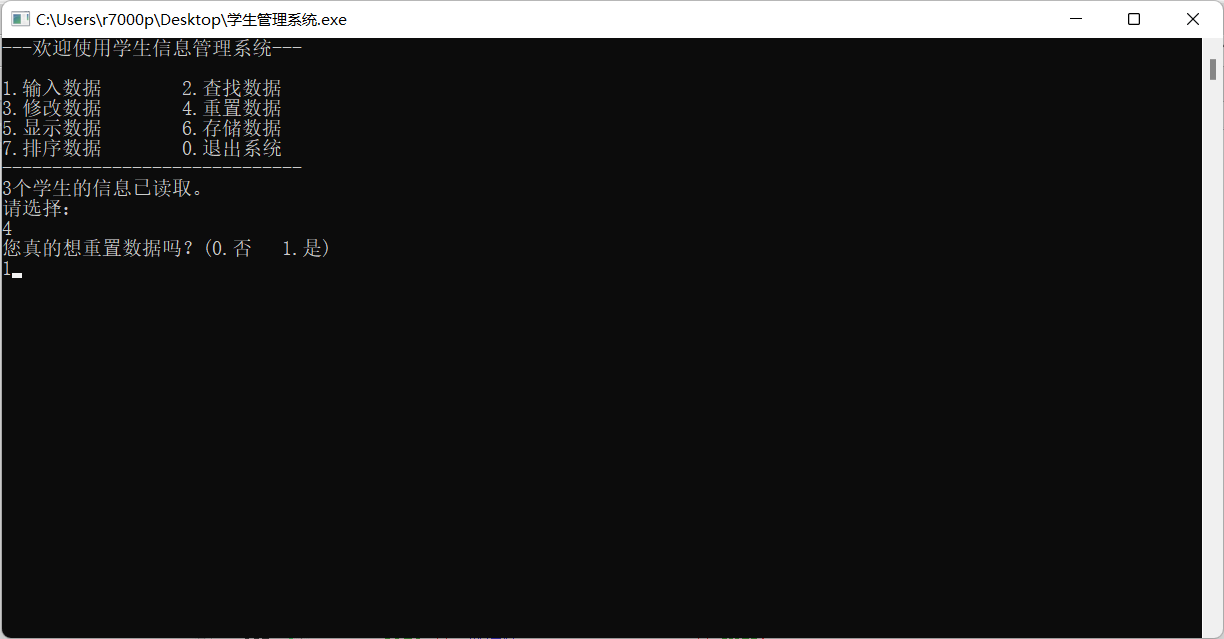
删除数据：



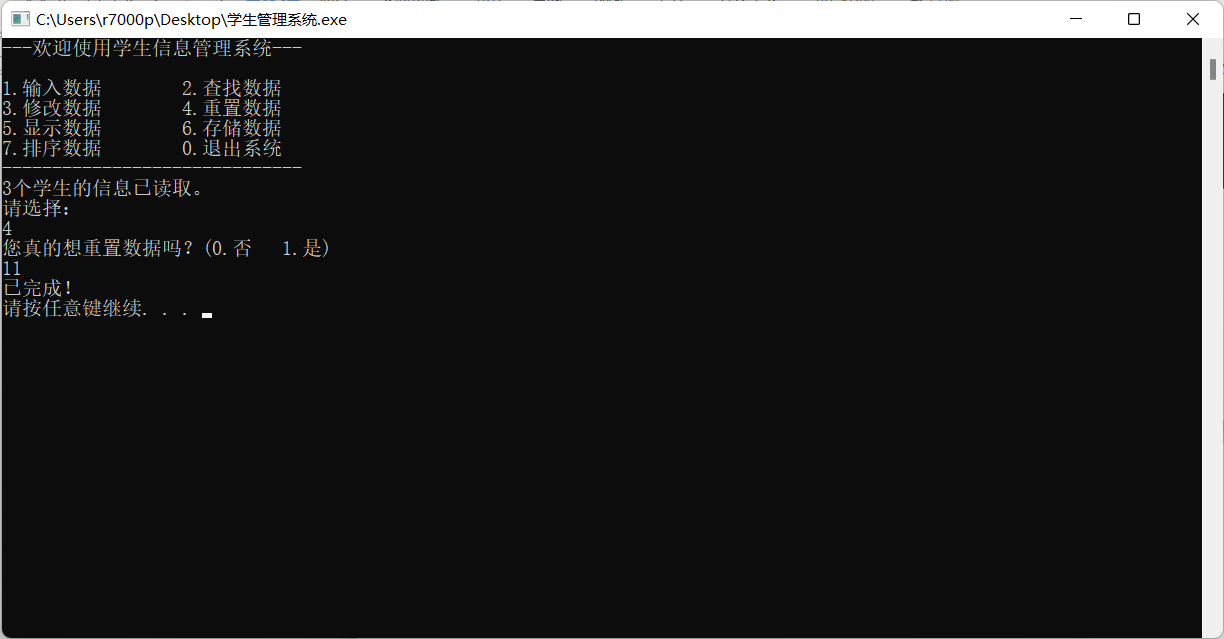
删除成功

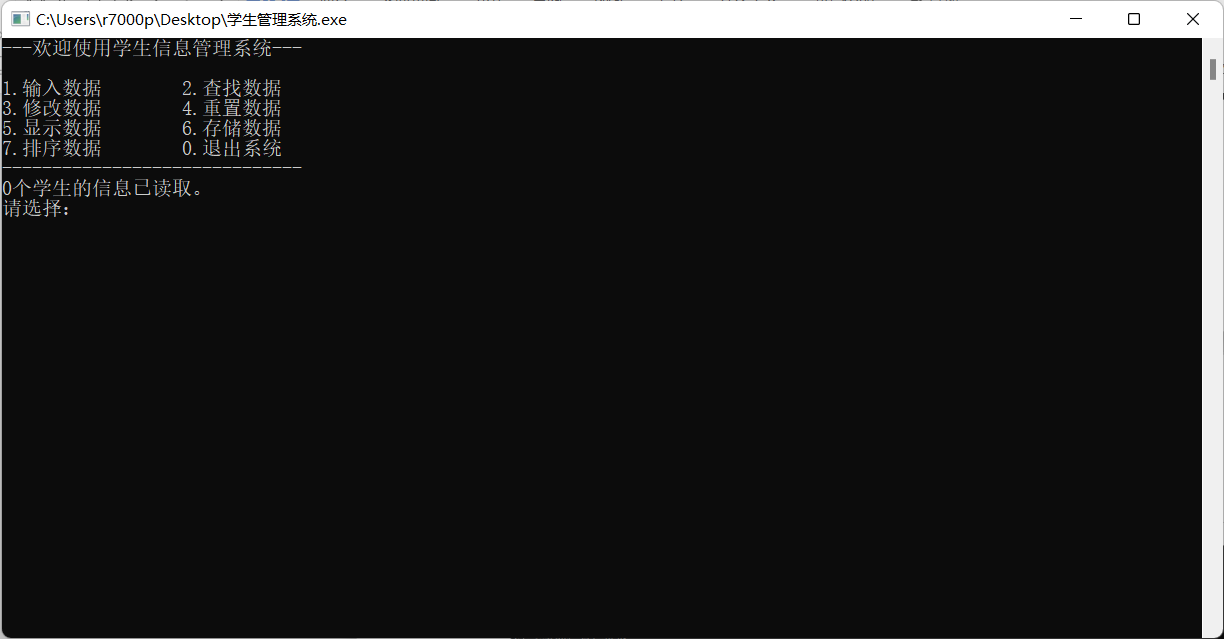


清除数据：



清除成功：





5. 源代码

#include<stdlib.h>

#include<stdio.h>

#include<direct.h>

#include<fstream>

#include<iostream>

#include<string.h>

#include<iomanip>

using namespace std;

//基本变量

int stumax = 0;

//读入的学生一共有多少

//输出选项

bool class\_limited(false), score\_limited(false), no\_limited(false);

char selected\_class(0), selected\_subjects[3] = { 1,1,1 }; //sub:[0]:数学 [1]:英语 [2]:C++

float min\_math\_score = 0, max\_math\_score = 99999, min\_english\_score = 0, max\_english\_score = 99999;

float min\_program\_score = 0, max\_program\_score = 99999;

float min\_aver\_score = 0, max\_aver\_score = 99999, min\_total\_score = 0, max\_total\_score = 9999999;

char selected\_no[25], selected\_subjects1[4]; //字符串多存一个\0

//修改用

char change\_no[25];

struct student

{

char no[25], name[25];//学生学号和名字，最多24字符

int Class;//班级

float math, english, program, ave, total;//分别是数学，英语，C++，平均分，总分，用float存，防止小数，平均分和总分不能手动修改

}a[100];//学生数组

void read\_data()

{//读取数据

stumax = 0;

ifstream test("student\_data.txt");

//ifstream输入流

if (!test.is\_open())

{//is\_open()这个函数能检测是不是打开了这个文件，如果没有，返回假，写的时候很方便，!is\_open就是没打开的意思。

ofstream create("student\_data.txt");

//注：如ifstream刚创建时参数写的那个文件，输入就用的哪个文件，ofstream参数写的哪个文件， ofstream<<就向哪个文件输出。

create << 0;//读取方式是先读取有多少人，再读取人数次数据，读到每个人的数据，输出0

cout << "无数据，已新建存档";

//提示没有找到数据，已经新建

create.close();

//关闭输出流

}

else

{

test.close();//用于测试，在上面is\_open测试下来的，如果打开了文件，证明已经存在，如果没有手动修改过，文件的内容

//应该没问题，验证完毕，就把测试用的流关掉。

}

ifstream fin("student\_data.txt");//重新开一个输入流，还打开student\_data.txt，当然可以用test，只是为了和上面分开额。

fin >> stumax;//和后面的写入顺序一样，先读入记录的学生总数，好知道要读入多少个学生数据

//注1：fin和上面的create，test一样，只是自己起名为这个，可以随便起名。

//注2：ifstream用法和cin一样，但是不需要键盘输入，而用文件输入，比如每空格/tab分隔输入，回车啥的。

for (int i = 0; i < stumax; i++)

{//循环stumax次，i=0，i<stumax正好是stumax次，比如i=0;i<3，取值是0，1，2三次。

fin >> a[i].no >> a[i].name >> a[i].Class >> a[i].math >> a[i].english >> a[i].program;

//和下面的输入顺序一样，是自己定义的存储方式，学号、名字、班级、成绩。

a[i].total = a[i].math + a[i].english + a[i].program;//总成绩是三科之和

a[i].ave = a[i].total / 3;//平均成绩是总成绩除以3

//总成绩和平均成绩不写入/输入，直接计算，防止出错。

}

cout << "已读取" << stumax << "位学生的记录。" << endl;

//提示读取了多少学生记录。

fin.close();

//关闭输入流

}

void write\_data() //写入数据

{//写入数据

ofstream fout("student\_data.txt", ios::out);

//打开输出流，前面参数是文件路径，后面参数是打开方式，因为这个out变量不在std命名空间里，所以要加ios::表示命名空间在ios

//ios指的是In/Out Stream,std是STanDard

if (!fout.is\_open())

{//如果打开失败，就退出，虽然一般不会出错吧。。

cerr << "出现错误@=-=@！" << endl;

exit(0);//调用这个系统函数直接退出

}

fout << stumax << endl;//先输出有多少人，这样读取的时候就知道读取多少次

for (int i = 0; i < stumax; i++)//stumax次输出，先输出endl换行和后输出没啥区别，每个数据间加个空格，读取时好识别是不同的数据

//就像cin输入时，一行输入多个数据要加空格。

fout << endl << a[i].no << ' ' << a[i].name << ' ' << a[i].Class << ' ' << a[i].math << ' ' << a[i].english << ' ' << a[i].program;

fout.close();//关闭

}

void enter\_data()

{//基础输入

fflush(stdin);

//清空cin缓冲区，没啥用，防止多余数据读入，调用后cin里没读取的数据会被清空。

for (int i = stumax; i < 100; i++)

{//最多读取100个，但是可以中途打断

//i = stumax的原因是，上次输入后有stumax个学生，所以追加时，下一个空位置的下标就是stumax，下标从0开始，所以很奇妙

cin >> a[i].no;//输入学号，也可能是结束符，在输入学号时输入0即可结束输入，或者等到输入满100个后自动结束

if (! (a[i].no[0] == '0' && a[i].no[1]=='\0') )

{//如果输入的是"0"，no里存的就是 0 \0 \0 \0 \0...如果输入别的

//字符数组里默认存的全是\0，如果输入一个字符串，读入会依次到数组里,

//但是输入最后会默认加个\0，所以假如输入“0”，读入的就是a[0] == 0

//a[1] == \0,所以没有结束的条件就是不符合这个，就是!(a[0]==0&&a[1]=='\0')

//如果要问这俩有什么区别，'0'是0这个字符的ascii码，\0的码值是0

cin >> a[i].name >> a[i].Class >> a[i].math >> a[i].english >> a[i].program;

//输入顺序

a[i].total = a[i].math + a[i].english + a[i].program;

a[i].ave = a[i].total / 3;

//计算总分和平均分

stumax++;//记录多了一个学生

}

else//假如打断或者装满了，就结束输入。

break;

}

//每次输入完成后调用这个函数，把数据写入文件，防止忘记保存。。当然也不能撤销了。

write\_data();

}

void show()

{//基础输出

if (stumax != 0)

{//stumax 是学生数，如果不等于0，就说明有记录，如果有记录，则输出

cout << " \t学号\t姓名\t班级\t数学\t英语\tC++\t平均分\t总分" << endl;//输出表头，这个是基本输出，测试用的

for (int i = 0; i < stumax; i++)//setw是对齐用的宽度为12个字符算空格和输出的字符

cout << i + 1 << "." << setw(12) << a[i].no << '\t' << a[i].name << '\t' << a[i].Class << '\t' << a[i].math << '\t' << a[i].english << '\t' << a[i].program << '\t' << fixed << setprecision(1) << a[i].ave << '\t' << a[i].total << endl;

}

}

void insstu(int ins\_start, int ins\_num)

{//插入数据

int ins\_index = ins\_start - 1;

for (int i = (stumax - 1) + ins\_num; i >= ins\_index + ins\_num; i--) //移位

{//stumax - 1是原本的最后一个数据，+ins\_num后就是它应该去的地方，直到移动到把ins\_index 所在位置也挪到ins\_num个之后。

a[i] = a[i - ins\_num];

}//每次将ins\_num个之前的数据移动到当前位置

cout << "请输入学生的学号，名字，班级，数学，英语，C++成绩\n";//移动完成，输出提示

for (int i = ins\_index; i < ins\_index + ins\_num; i++)

{//输入数据，计算，从ins\_index开始，输入ins\_num个

cin >> a[i].no >> a[i].name >> a[i].Class >> a[i].math >> a[i].english >> a[i].program;

a[i].total = a[i].math + a[i].english + a[i].program;

a[i].ave = a[i].total / 3;

stumax++;//加一个人

}

write\_data();//输入后，写入数据

}

void output\_reset()

{//重置查找选项

class\_limited = 0, score\_limited = 0, selected\_class = 0, memset(selected\_subjects, 1, sizeof(selected\_subjects)); //osub:1:数学 2:英语 3.C++

min\_math\_score = 0, max\_math\_score = 99999, min\_english\_score = 0, max\_english\_score = 99999, min\_program\_score = 0, max\_program\_score = 99999, min\_aver\_score = 0, max\_aver\_score = 99999, min\_total\_score = 0, max\_total\_score = 999999;

}

void output\_lim()

{//查找

cout << endl;

int j = 1; //记录输出数量

cout << " \t学号\t姓名\t班级";

if (selected\_subjects[0])cout << '\t' << "数学";

if (selected\_subjects[1])cout << '\t' << "英语";

if (selected\_subjects[2])cout << '\t' << "C++";

if (selected\_subjects[0] && selected\_subjects[1] && selected\_subjects[2])

cout << '\t' << "平均分" << '\t' << "总分"; //列表表头

cout << endl;

for (int i = 0; i < stumax; i++)

{

if ((!no\_limited || memcmp(a[i].no, selected\_no, 25) == 0) && (!class\_limited || a[i].Class == selected\_class) && (a[i].math >= min\_math\_score && a[i].math <= max\_math\_score) && (a[i].english >= min\_english\_score && a[i].english <= max\_english\_score) && (a[i].program >= min\_program\_score && a[i].program <= max\_program\_score) && (a[i].ave >= min\_aver\_score && a[i].ave <= max\_aver\_score) && (a[i].total >= min\_total\_score && a[i].total <= max\_total\_score))

{// 全部条件满足时进行输出

cout << j << '.' << setw(12) << a[i].no;

cout << '\t' << a[i].name << '\t' << a[i].Class << '\t';

if (selected\_subjects[0]) cout << a[i].math << '\t';

if (selected\_subjects[1]) cout << a[i].english << '\t';

if (selected\_subjects[2]) cout << a[i].program << '\t';

if (selected\_subjects[0] && selected\_subjects[1] && selected\_subjects[2]) //全部显示时显示平均分和总成绩

cout << fixed << setprecision(1) << a[i].ave << '\t' << a[i].total;

cout << endl;

j++;

}

}

if (j == 1) //显示数量判定

cout << "未查找到符合条件的学生QAQ 您可以修改条件。。" << endl;

else

cout << "查找到了" << j - 1 << "位符合条件的学生！" << endl;

system("pause");//暂停

}

void outPut\_menu()

{//查找菜单

int outputSelect(0);

while (1)

{

cout << "-------请选择查找方式--------" << endl;

cout << "1.按班级输出 2.指定学号查找" << endl;

cout << "3.显示指定科目 4.按成绩输出" << endl;

cout << "5.清除条件 6.显示目前条件" << endl;

cout << "9.直接输出(测试) 0.退出查询" << endl;

cout << "-----------------------------" << endl;

cout << "多条件查找请逐个添加 " << endl;

cin >> outputSelect;

switch (outputSelect)

{

case 1:

class\_limited = true;//设置标志

cout << "请输入您想查看的班级，输入0来显示全部班级：\n";

cin >> selected\_class;

if (selected\_class == 0) class\_limited = 0;

output\_lim();

break;

case 2:

no\_limited = 1; //激活条件变量

output\_reset(); // 重置变量以从全部学生中查找

cout << "请输入您想查找的学生学号：";

cin >> selected\_no;

output\_lim();

no\_limited = 0; //复位条件变量

break;

case 3:

memset(selected\_subjects, 0, sizeof(selected\_subjects));

cout << "请输入您想显示的科目序号" << endl;

cout << "1.数学 2.英语 3.C++" << endl;

cout << "输入“12”以显示数学和英语" << endl;

cin >> selected\_subjects;

for (int i = 0; i < 3; i++)

{

if (selected\_subjects[i] != '\0') selected\_subjects[(selected\_subjects[i] - 49)] = 1;

}

output\_lim();

break;

case 4:

cout << "请输入您需要限制的科目：" << endl;

cout << "1.英语 2.数学 3.C++ 4.平均分 5.总分" << endl;

cin >> outputSelect;

cout << "请输入您需要查找的最低分与最高分：" << endl;

switch (outputSelect)

{

case 1:

cin >> min\_english\_score >> max\_english\_score;

break;

case 2:

cin >> min\_math\_score >> max\_math\_score;

break;

case 3:

cin >> min\_program\_score >> max\_program\_score;

break;

case 4:

cin >> min\_aver\_score >> max\_aver\_score;

break;

case 5:

cin >> min\_total\_score >> max\_total\_score;

break;

}

output\_lim();

break;

case 5:

output\_reset();

cout << "成功！" << endl;

break;

case 6:

cout << "目前显示" << selected\_class << "班,显示科目";

if (selected\_subjects[0]) cout << "数学 ";

if (selected\_subjects[1]) cout << "英语 ";

if (selected\_subjects[2]) cout << "C++ ";

cout << endl;

break;

case 9:

show();

}

if (outputSelect == 0)

{

output\_reset(); //退出时复位变量

break;

}

}

}

int change(int insa, char posa[])

{//更改数据

int asw = 0;

if (memcmp(posa, "学号", 4) == 0) { cin >> a[insa].no; asw = 1; }

if (memcmp(posa, "姓名", 4) == 0) { cin >> a[insa].name; asw = 1; }

if (memcmp(posa, "班级", 4) == 0) { cin >> a[insa].no; asw = 1; }

if (memcmp(posa, "数学", 4) == 0) { cin >> a[insa].math, asw = 1; }

if (memcmp(posa, "英语", 4) == 0) { cin >> a[insa].english; asw = 1; }

if (memcmp(posa, "C++", 4) == 0) { cin >> a[insa].program; asw = 1; }

a[insa - 1].total = a[insa - 1].math + a[insa - 1].english + a[insa - 1].program, a[insa - 1].ave = a[insa - 1].total / 3;

write\_data();

return asw;

}

void dlt(int pos)

{//删除数据

for (int i = pos; i < stumax; i++)

{//覆盖

a[i] = a[i + 1];

}

stumax--;

write\_data();

}

void searchstuNo()

{//更改删除菜单

int recno(-1);

char yon;

int changeSelect;

char changea[15];

for (int i = 0; i < stumax; i++)

if (memcmp(a[i].no, change\_no, 25) == 0)

{

recno = i;

break;

}

if (recno == -1) cout << "未找到学生！" << endl;

else

{

cout << "\t学号\t姓名\t班级\t数学\t英语\tC++\t平均分\t总分" << endl;

cout << recno << '.' << setw(12) << a[recno].no;

cout << '\t' << a[recno].name << '\t' << a[recno].Class << '\t' << a[recno].math << '\t' << a[recno].english << '\t' << a[recno].program << '\t' << fixed << setprecision(1) << a[recno].ave << '\t' << a[recno].total;

cout << endl;

cout << "请选择-----------------" << endl;

cout << "1.更改数据 2.删除数据" << endl;

cout << "-----------------------" << endl;

cin >> changeSelect;

switch (changeSelect)

{//更改或删除

case 1:

cout << "请输入您想修改的信息：学号、姓名、班级、数学、英语、C++" << endl;

cin >> changea;

if (change(recno, changea))

cout << "修改成功！" << endl;

else

cout << "修改失败！" << endl;

break;

case 2:

cout << "您确定吗？Y是N否" << endl;

cin >> yon;

if (yon == 'y' || yon == 'Y')

dlt(recno);

if (yon == 'n' || yon == 'N')

break;

default:

cout << "输入错误！" << endl;

}

}

}

void sort\_data()

{//排序

int sorta(0), sortc;

student temp;

cout << "请输入需要排序的选项：\n1.数学 2.英语 3.C++ 4.平均分 5.总分" << endl;

cin >> sortc;

sortc -= 1;

cout << "请输入排序方式：1.由高到低 2.由低到高" << endl;

cin >> sorta;

student t;

for (int i = 0; i < stumax - 1; i++)

for (int j = i + 1; j < stumax; j++)

{

float sorts[5] = { a[i].math,a[i].english,a[i].program,a[i].ave,a[i].total };

float sortss[5] = { a[j].math,a[j].english,a[j].program,a[j].ave,a[i].total };

if ((sorts[sortc] < sortss[sortc] && sorta == 1) || (sorts[sortc] > sortss[sortc] && sorta == 2))

{

t = a[i];

a[i] = a[j];

a[j] = t;

}

}

}

int start()

{//主菜单

int mainSelect;

int insloc, insnum; //插入位置

read\_data();

while (1)

{

system("pause");

system("cls");

cout << "---欢迎使用学生信息管理系统---" << endl << endl;;

cout << "1.输入数据 2.查找数据" << endl;

cout << "3.修改数据 4.重置数据" << endl;

cout << "5.显示数据 6.存储数据" << endl;

cout << "7.排序数据 0.退出系统" << endl;

cout << "------------------------------" << endl;

cout << stumax << "个学生的信息已读取。" << endl;

cout << "请选择：\n";

cin >> mainSelect;

switch (mainSelect)

{

case 1:

cout << "请选择输入方法:\n1.指定位置添加\t2.于结尾处添加" << endl;

cin >> mainSelect;

switch (mainSelect)

{

case 1:

if (stumax == 0)

{

cout << "没有数据！" << endl;

break;

}

show();

cout << "请您输入您想添加的位置与添加的人数：" << endl;

cin >> insloc >> insnum;

if (insloc <= 0 || insloc >= stumax)

cout << "你想干嘛？没有这个位置。" << endl;

else

insstu(insloc, insnum);

break;

case 2:

cout << "请输入学生的学号，名字，班级，数学，英语，C++成绩,于学号处输入0来结束：\n";

enter\_data();

break;

default:

cout << "您的输入有误!已返回。" << endl;

break;

}

break;

case 2: if (!stumax)

{

cout << "无数据！\n"; break;

}

else

{

outPut\_menu(); break;

}

case 3:

output\_lim();

cout << "请输入您想要修改的学生的学号：（不支持修改平均分和总分）" << endl;

cin >> change\_no;

searchstuNo();

break;

case 4:

int reset;

cout << "您真的想重置数据吗？(0.否 1.是)" << endl;

cin >> reset;

if (reset)

{

for (int i = 0; i < 100; i++)

memset(&a[i], 0, sizeof(struct student));

stumax = 0;

cout << "已完成！\n";

write\_data(); //写入

}

else

cout << "已取消！" << endl;

break;

case 5:

show();

break;

case 6:

write\_data();

system("cls");

cout << "成功！" << endl;

break;

case 7:

sort\_data();

write\_data();

break;

case 0:

return 0;

default:

cout << "您的输入有误！" << endl;

break;

}

}

}

int main()

{

start();

return 0;

}

1. 参考文献
2. 《C++面向对象程序设计教程》陈维星 林小茶等编著，清华大学出版社 2018
3. 《C++大学教程》，Harvey M.Deitel等著，邱仲潘等译，电子工业出版社 2001

3、《C++精粹》，Ira Pohl著，王树武等译，机械工业出版社 2003

4、《数据结构》，严蔚敏编，清华大学出版社 2004

7. 设计体会

一开始知道课题后，以为很简单，但我想如果用数组实现的话，在内存中是以连续地址存放的，不方便操作，所以我选择了使用链表，它可以实现无序的在内存中存放，只要有一个头结点，就可以对整个链表进行相关的操作，在调式时遇到了很多问题，最多的是指针问题引起的内存不能为read，因此指针这一块知识还得好好的加强，当我将学生成绩管理系统的程序调试完毕后，心里感到轻松了许多，但还是对自己的程序不最满意，因为有时还是会出现内存不能read。我现在体会到程序的调试是相当的重要的。在使用指针时要特别注意用过后的指针不使用就要删除，已防出错。链表的尾部要指向NULL，这点需特别注意的。  
 成绩管理系统如果只是将结果在屏幕显示后就不存在了，就没有意义了，于是我添加了文件的读出和写入的功能，写入时如果没有目标文件，则系统会自动在目标文件夹里生成一个文件用来储存用户数据，该文件类型可以按用户要求生成，如txt文本文档或doc文档等。编写这次代码，在如何进行成绩排序上费了些力气，虽然老师讲过冒泡排序和选择排序，可是运用到单链表上就着实感到困难了，参考了老师PPT上的插入排序方法实现了，我调试程序的时候用无效数据测试时，没有用特殊的符号去测试，我用一些符号测试了我的程序，结果如我所料，程序出现了错误，我没有找到解决的方法。如果我能解决上述缺点，我想这个程序应该算是合格了。  
 不管怎么说，这次课程设计还是锻炼了我编写程序的能力，对以前学过的知识有了进一不的了解,特别是链表这一知识点。  
 此次课程设计最大的收获是：学会了自己学习。

纸上得来终觉浅，绝知此事要躬行！