《C 语言程序设计》实验大作业反思报告

大作业题目	基于 EGE 图形界面化库的学生信息管理系统			类型	信息管理系统
班　号	22L0101		学　号	2022111915	
所在院系	航天与自动化	学　期	2023 年　春季学期	任课教师	胡涛
实验类型	综合设计型				

实验目的：

- 掌握程序设计的基本算法和简单数据结构基础，能够综合运用基本控制语句、算法和数据结构，以及自顶向下、逐步求精的模块化设计方法，能够设计具有一定规模的系统级C语言程序，提高系统编程能力；
- 针对计算相关的复杂工程问题，能够使用恰当的算法和数据结构，完成计算、统计、排序、检索、匹配等相关的软件系统的构造、测试与实现；
- 掌握常用的程序调试和测试方法。

实验要求：

- 采用自顶向下、逐步求精的模块化设计思想设计一个小型信息库管理系统，或者闯关式游戏程序。
- 要求解释说明采用了什么数据结构和算法，为什么选择这种数据结构或算法，系统实现过程中遇到了哪些问题，这些问题是如何解决的，还有什么问题尚未解决，今后打算从哪几个方面进行改进，本设计的亮点和难点在哪里，实验结果如何，有哪些收获和学习体会；
- 编写程序完成以下实验大作业内容并完成实验大作业反思报告。

实验内容：

注：该工程调用外部库需要使用 C++语言，其余算法和接口使用 C 语言进行编写，环境为 Visual Studio2022，使用了《C 语言程序设计学习指导》中介绍的 EGE 图形界面化库（版本号为 19.01）。文件夹外部和内部均有同样的 ege 文件夹，包含了所有配置需要的文件，请您按照书中教程或 https://blog.csdn.net/qq_39151563/article/details/100161986 中寻找对应的 IDE 配置方法进行配置。若您配置失败，请直接运行 Project1 中的 main.exe 进行配置。界面主要负责交互，查看学生数据时请切换到控制台（cmd 黑窗口）进行查看。使用前请阅读“使用须知(README).txt”。

重要内容：使用前请阅读使用须知

若您在配置中遇到困难，可以直接尝试运行打包好的文件，在文件夹中右键“setup.exe”，选择“以管理员身份运行”（否则 windows10 用户可能出现安装失败）即可像安装其他软件一样完成安装，然后在桌面找到“学生信息管理系统”，双击运行即可打开，或者打开安装的目录，运行 StudentManager.exe。若使用完毕，可以点击“uninstall”对软件进行卸载。

基于 EGE 图形界面化库的学生信息管理系统

设计一个学生成绩管理系统，包括学生成绩数据的增删改查等基本操作，并根据不同学生进行包装，要求编程实现如下菜单驱动的学生成绩管理系统：

（1）初始化学生信息，录入科目数量（最大为 5），录入对应科目名称，实现第一次录入后再控制台进行预览操作。

（2）在“学生信息管理”界面中，实现了对学生的继续增加、删除。

（3）计算每个学生的总分

（4）按照总分从高到低进行排序，并在控制台中查看输出的学生数据。

（5）可以按照学生学号进行模糊查找。

（6）可以按照学生姓名进行模糊查找。

（7）在初始化学生信息后，将信息写入 score.txt 文件中进行保存。

（8）可以从 score.txt 文件中进行读取数据。

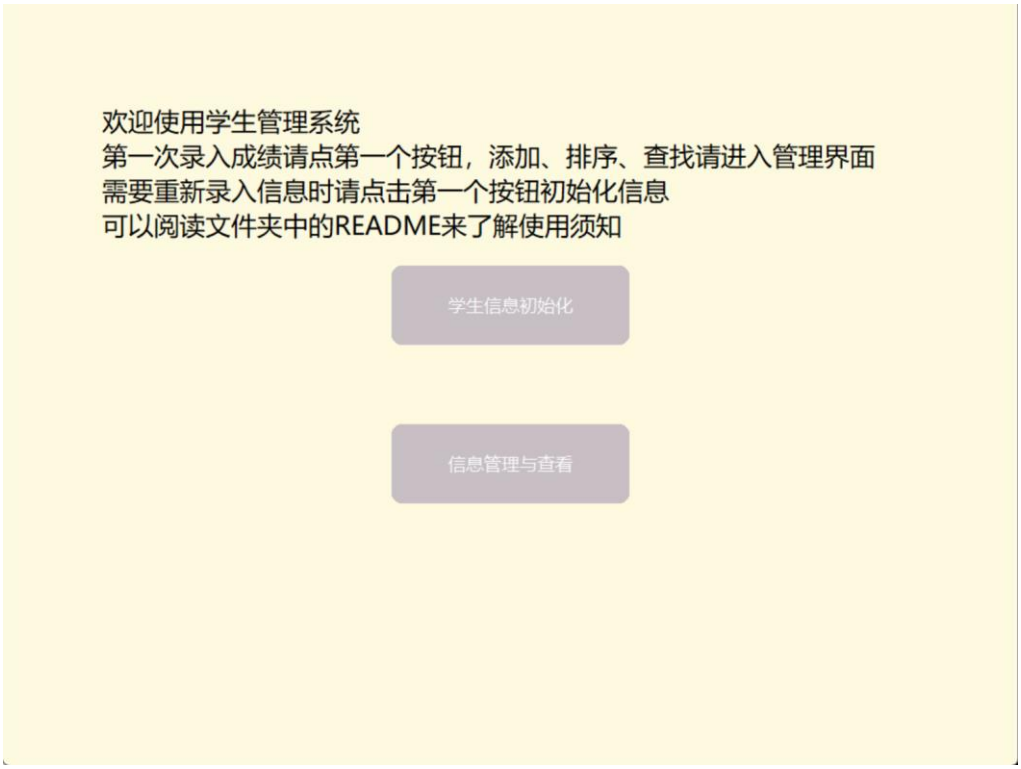
（9）使用了图形化界面，直观明朗。

（10）实现了创建按钮，创建对话框，可以直接通过图形化界面进行鼠标交互，可以直接输入数据。

（11）检查了非法输入，并提示用户重新输入，保证程序的健壮性。

（12）实现了背景音乐的播放。

要求程序运行后先显示如下菜单，并提示用户输入选项：



然后，根据用户输入的选项执行相应的操作。

实验环境：

操作系统：windows11

集成开发环境：Visual Studio 2022 编码 GB18030

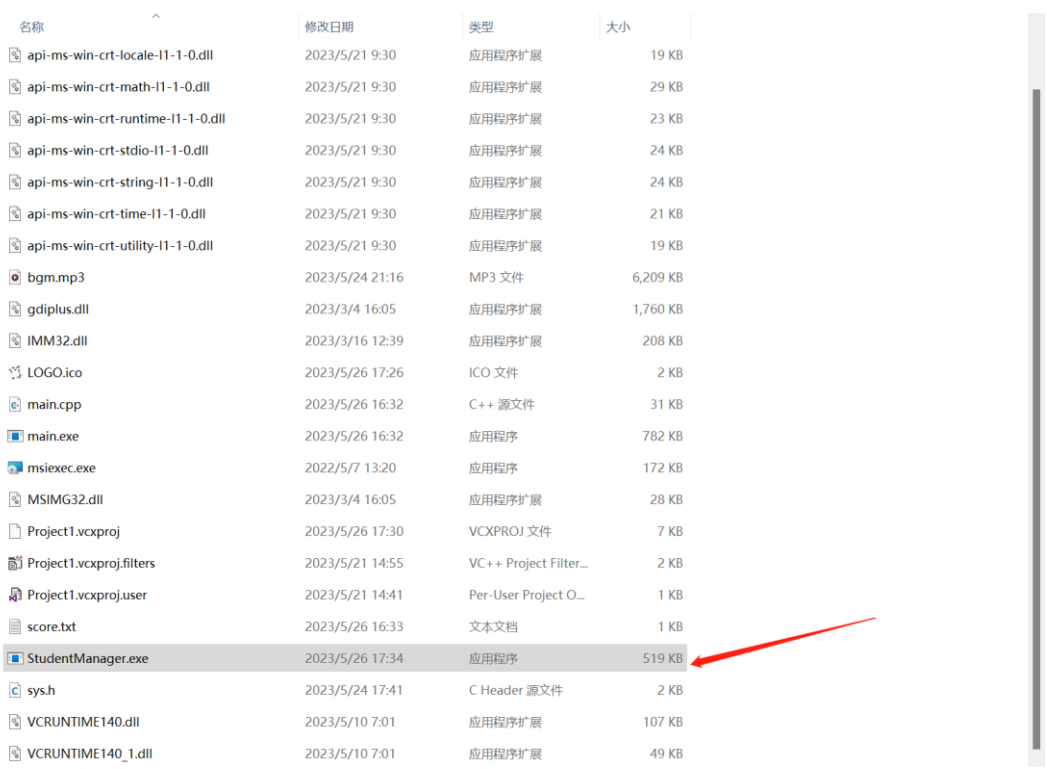
外部库：使用了《C 语言程序设计学习指导》中介绍的 EGE 图形界面化库（版本号为 19.01）。文件夹外部和内部均有同样的 ege 文件夹，包含了所有配置需要的文件，请您按照书中教程或 https://blog.csdn.net/qq_39151563/article/details/100161986 中寻找对应的 IDE 配置方法进行配置。

重要内容：使用前请阅读使用须知

若您在配置中遇到困难，可以直接尝试运行打包好的文件，在文件夹中右键“setup.exe”，选择“以管理员身份运行”（否则 windows10 用户可能出现安装失败）即可像安装其他软件一样完成安装，然后在桌面找到“学生信息管理系统”，双击运行即可打开，或者打开安装的目录，运行 StudentManager.exe。若使用完毕，可以点击“uninstall”对软件进行卸载。

桌面：

安装目录



名称	修改日期	类型	大小
api-ms-win-crt-locale-l1-1-0.dll	2023/5/21 9:30	应用程序扩展	19 KB
api-ms-win-crt-math-l1-1-0.dll	2023/5/21 9:30	应用程序扩展	29 KB
api-ms-win-crt-runtime-l1-1-0.dll	2023/5/21 9:30	应用程序扩展	23 KB
api-ms-win-crt-stdio-l1-1-0.dll	2023/5/21 9:30	应用程序扩展	24 KB
api-ms-win-crt-string-l1-1-0.dll	2023/5/21 9:30	应用程序扩展	24 KB
api-ms-win-crt-time-l1-1-0.dll	2023/5/21 9:30	应用程序扩展	21 KB
api-ms-win-crt-utility-l1-1-0.dll	2023/5/21 9:30	应用程序扩展	19 KB
bgm.mp3	2023/5/24 21:16	MP3 文件	6,209 KB
gdiplus.dll	2023/3/4 16:05	应用程序扩展	1,760 KB
IMM32.dll	2023/3/16 12:39	应用程序扩展	208 KB
LOGO.ico	2023/5/26 17:26	ICO 文件	2 KB
main.cpp	2023/5/26 16:32	C++ 源文件	31 KB
main.exe	2023/5/26 16:32	应用程序	782 KB
msiexec.exe	2022/5/7 13:20	应用程序	172 KB
MSIMG32.dll	2023/3/4 16:05	应用程序扩展	28 KB
Project1.vcxproj	2023/5/26 17:30	VCXPROJ 文件	7 KB
Project1.vcxproj.filters	2023/5/21 14:55	VC++ Project Filter...	2 KB
Project1.vcxproj.user	2023/5/21 14:41	Per-User Project O...	1 KB
score.txt	2023/5/26 16:33	文本文档	1 KB
StudentManager.exe	2023/5/26 17:34	应用程序	519 KB
sys.h	2023/5/24 17:41	C Header 源文件	2 KB
VCRUNTIME140.dll	2023/5/10 7:01	应用程序扩展	107 KB
VCRUNTIME140_1.dll	2023/5/10 7:01	应用程序扩展	49 KB

输入输出设计：

1.进行了对非法输入的检测，文件打开失败，动态内存分配失败将会在控制台中提示

2.输入：设置科目数量（int），科目名称（string）（英文）学生学号（string）（数字英文均可）学生姓名（string）（英文）学生成绩（int）

3.对大部分数据进行了非法检测和限制，比如在输入科目数量和学生分数时将会进行检查，如果输入范围或类型有误将会进行提示并要求重新输入，其他数据设置了对话框进行限制，当超过规定的最大位数+1 时就会无法输入（+1 是为了保证能完整读入），且最后一位将会被忽略（最终位数与程序要求的一致），这不是程序 bug。

4.由于 EGE 库输出文本的功能比较单一且不够美观，在项目编写初期，曾花费近一天时间进行排版，但最终为了美观性选择放弃。因此程序仅在交互和提示中在图形化界面进行了输出，这样可以使用 printf 将数据对齐，便于查看。查看学生数据时，程序在图形化界面上提示在控制台查看，使用者需要手动打开运行时自动打开的控制台(cmd)进行查看数据。

5.在进入信息管理与查看时，程序会从文件中读入数据并自动排序。

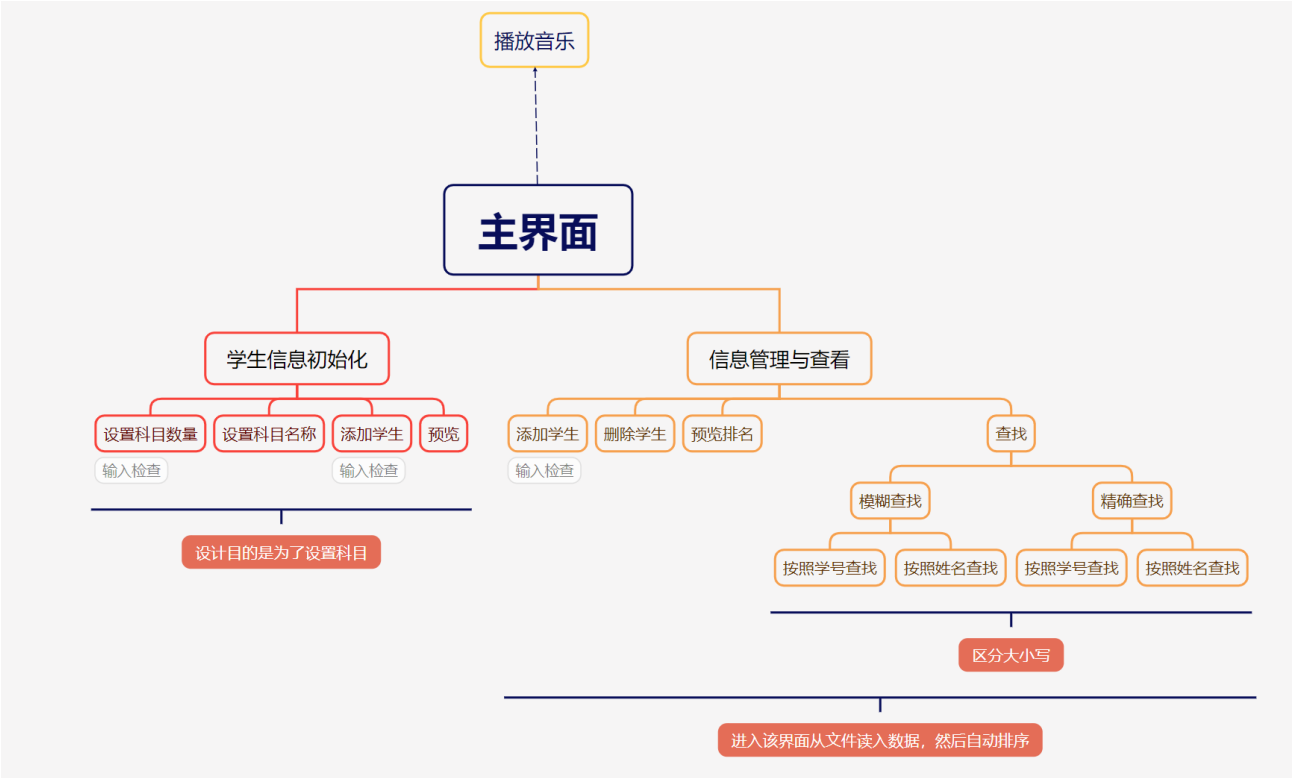
6.为了使得程序调用播放音乐的代码不被编译器认定为错误(但是其实能正常运行)，程序编译使用富文本，不是 UNICODE。

7.首次使用 score.txt 是空文件，必须先进入学生信息初始化界面对信息进行初始化，确定科目之后再录入信息，直接进入信息管理界面只能录入学号和名称，无法录入科目成绩，甚至可能导致程序崩溃（小概率）。

系统设计与实现:

1. 系统功能模块划分

对系统进行自顶向下的模块分解，画出系统各个功能模块之间的结构图如下：



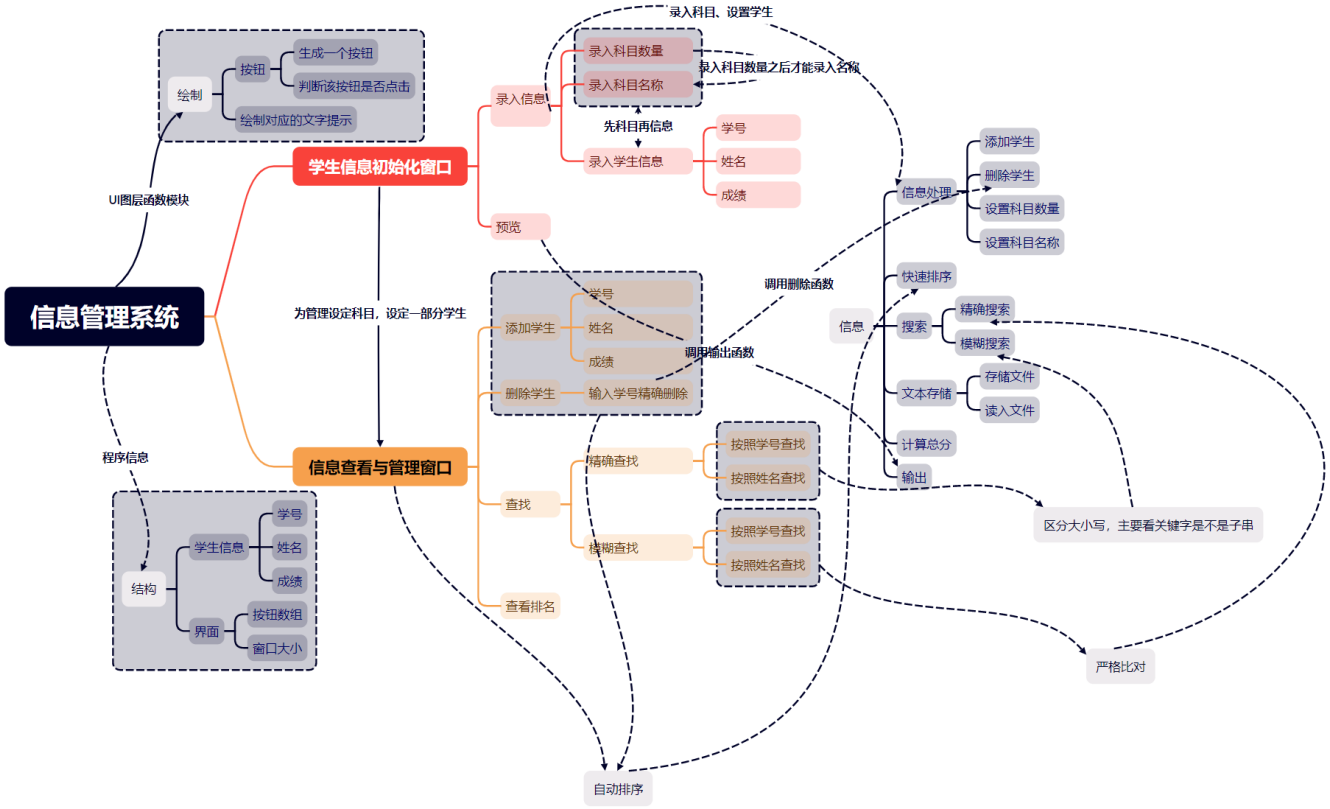
2. 函数功能和外部接口设计

本系统总计设计了19个自定义函数，每个自定义函数的功能和接口设计如下表所示：

序号	函数名	函数功能	函数参数	函数返回值
1	studentEntryWindow();	学生数据录入窗口	无	无
2	studentManagementWindow	学生数据管理窗口	无	无
3	mainWindow	主界面	无	无
4	drawButton	创建按钮	Button 结构体	无
5	click	判断鼠标点击	Button 结构体，点击坐标 x,点击坐标 y	Ture/false
6	puttext	输出文字	坐标 x,y,字体大小 size,文本 char text	无
7	InitEditbox	初始化编辑框	Editbox 类坐标 x,y，宽度 width，高度 height	无
8	draw_entry	绘制界面	返回按钮 back_x,back_y,按钮数组 Buttonarray	无
9	addstudent	添加学生	结构体 student 数量 count	count

			返回按钮 back_x,back_y 坐标,buttonarray 数组	
10	delstudent	删除学生	结构体 student 数量 count 返回按钮 back_x,back_y 坐标,buttonarray 数组	count
11	printstudent	按格式输出学生信息	结构体 student 学科名 subname	无
12	calculate_score	计算总分	结构体 student	无
13	Write_file	写入文件	结构体 student	无
14	Read_file	读入文件	结构体 student	无
15	swap	交换两个结构体位置	结构体指针 student *a,student*b	无
16	partition	快速排序确定基准位置	student arr[], 数组低位置 low，数组高位置 high	枢纽元素索引 i
17	quickSort	递归方法的快速排序	Student arr[],low,high	无
18	vaguesearch	模糊搜索	Char *key, char *str	找到的元素定位 pos，没有返回-1
19	ranksearch	随机搜索	student stu[], int count, const char* input_number	找到元素的定位 pos,没有找到返回-1

各个函数之间的调用关系如下所示：



3. 数据结构

（4分）：结构体数组（学生信息，按钮，文本） + 指针数组（输入查找时用的字符串）

加分项：

4. 算法

本工程中的加分项（算法）

（1分）：枚举、递推（任何地方随处可见）

（1分）：快速排序，精确查找

（1分）：读入文件，读出文件

（2分）：快速排序运用递归进行排序

（2分）：模糊查询算法（按学号、按姓名）

(1分)：随机算法（删除学生时查找学生学号用到）

本工程中的加分项（程序设计质量）

（1分）使用了图形界面，有足够明确的提示。且代码模块化良好

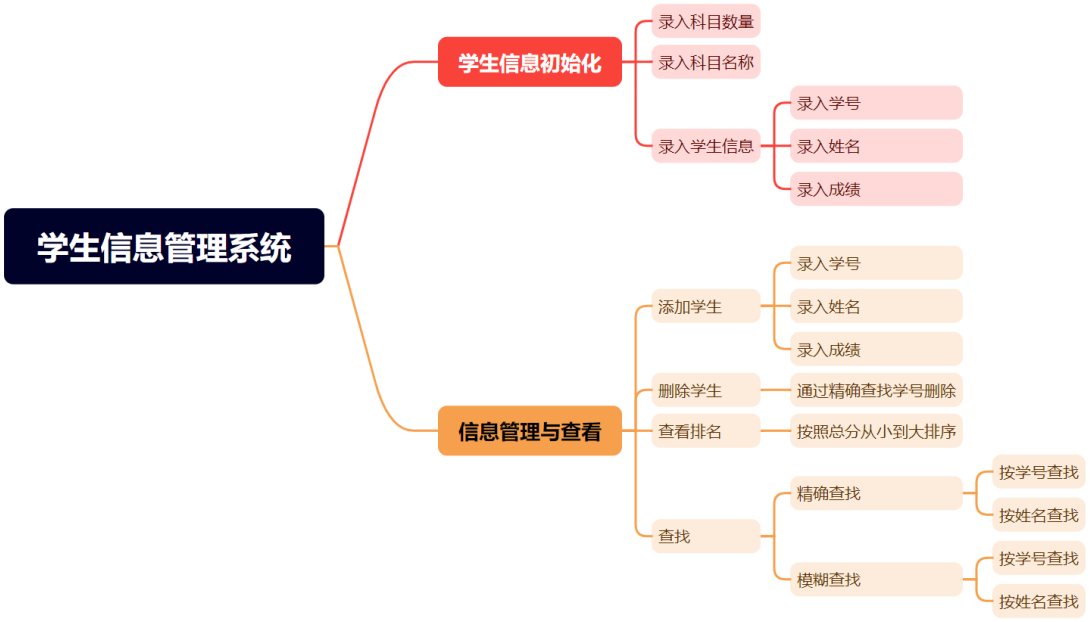
（1分）加入了背景音乐

（1分）代码有非常详细的注释，比较规范，变量名和函数名取名有效。

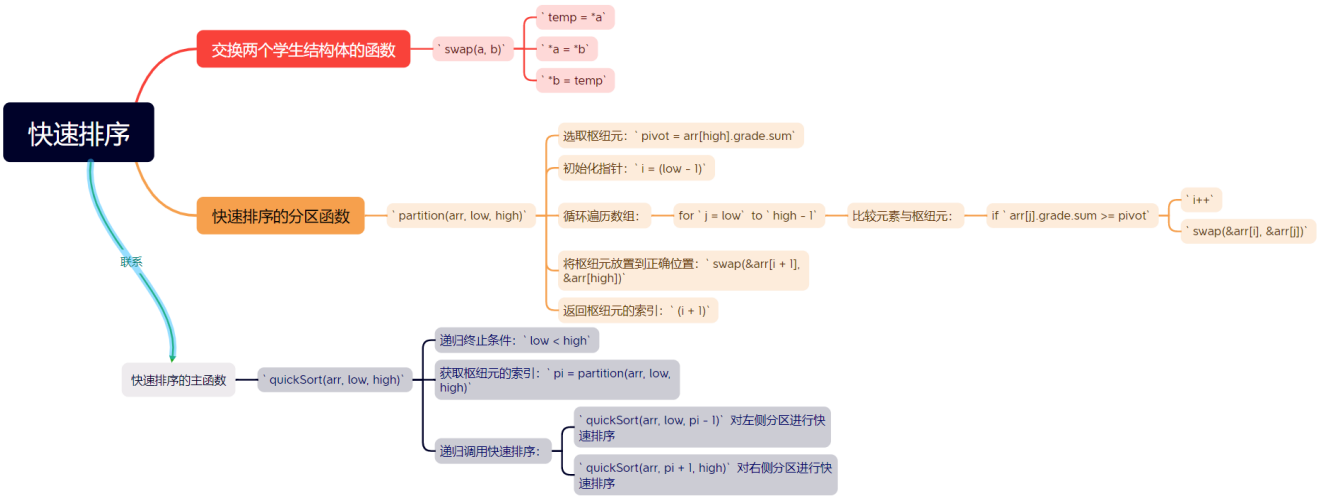
（1 分）有输入检查，代码模块化良好，函数封装和复用性较高

5. 程序流程图

系统总体流程图如下：

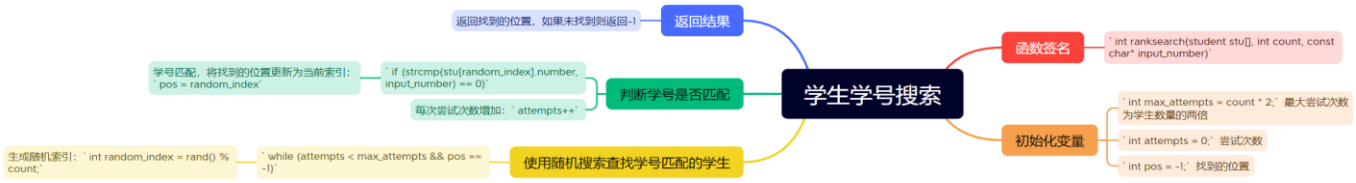


快速排序代码思维导图：



模糊搜索代码思维导图：

随机查找函数：



实验过程中遇到的问题及解决方法与思路：

问题 1： 无法生成按钮，在生成按钮时发现有较多按钮需要绘制，函数参数较多，且修改坐标、颜色、文字等特别困难。

原因：EGE 图形库中没有提供按钮组件，需要自己绘制

解决方法：自己通过编写函数，先绘制矩形，然后填充设定的颜色，最后在上面写上文字，最后优化函数参数时采用了结构体将按钮的坐标、大小、颜色、文字都存在结构体中，后期为了更加方便绘制按钮，将一个界面的按钮建立一个按钮结构体数组存放，然后用 for 循环遍历按钮结构体数组调用绘制按钮函数进行绘制，这样大大提高了复用性和可维护性。

问题 2： 输出提示信息后文本互相覆盖，使用户很难看清

原因：EGE 库的绘图逻辑是将元素依次绘制在窗口上，并不会自动刷新屏幕。

解决方法：运用 cleardevice();函数实现清屏，但是无法重新绘制，为了将界面重新绘制，我编写了 draw_entry();函数，由于界面都是以返回按钮为基准布置的，所以我需要传入按钮结构体，还有返回按钮的坐标，并在函数中吧每次都需要输出的文字放入其中，这样每次调用都会生成一样的页面，然后运用 delay();函数设置时间间隔避免处理过快。

问题 3：点击按钮无反应。

原因：按钮只是认为画出的一个图形集合，没有自动配置好处理鼠标点击的事件。

解决办法：编写 isclick();函数，传入鼠标点击的坐标，以及需要判断需要点击的按钮的结构体，由于本程序使用的是圆角按钮，四周有圆角，中间是一个矩形，因此分为五部分判断是否点击，判断鼠标坐标是否落在矩形区域内，通过极坐标变换公式鼠标点击坐标是否落在四个圆角区域内。

问题 4：处理鼠标信息有延迟

原因：鼠标信息在 EGE 图形库中是一个队列，在最初的代码中采取 if 结构来判断是否点击，但是鼠标信息包括鼠标移动，中键、左键、右键的按下和抬起，且鼠标信息的更新一秒大约 100 次，速度极快，使用 if 语句判断显然会遗漏信息造成鼠标信息处理不及时。

解决办法：使用 while(1)死循环来判断，然后在循环结束前使用 delay 函数进行延时处理，避免过快重复处理造成程序崩溃。在循环中使用 if else 语句来对各个按钮进行判断，从而很好地解决了这个问题。

问题 5：无法同时处理多个用户输入的程序。

原因：EGE 图形库中虽然提供了编辑框，但是通过查看头文件的底层代码发现，编辑框在图形界面之上，独立于图形界面，且由于 EGE 图形本身实现编辑框的原因很难实现多个文本同时输入，且使用编辑框极大的干扰了原本按钮的交互。

解决办法：通过使用 EGE 提供的 input_getline();函数来实现对文本的输入，且由于这个函数一次只能调出一个对话框，因此我采取依次输入的方式来录入信息。由于他的位置固定默认在窗口中间，因此后续排版将元素放在周围以防止调出对话框时对图形元素造成遮挡。

问题 6：调出对话框黑屏

原因：这是 EGE20.08 版本自身的 bug。

解决办法：初步采取的折衷办法是在图形界面上 puttext 出对应的提示信息来进行提示，在后来查阅书籍发现学校教材的学习指导中提供了配置 EGE19.01 版本的教程，更换为 19.01 版本发现问题解决，这样不仅解决了问题，也方便同学们根据书中的教程配置对应的库来对我的代码进行检查和交流。

问题 7：对话框输入后无法对分数和科目数量直接进行操作（例如排序、输入错误检查）。

原因：对话框输入的文本文本是字符串类型，我输入的科目数量、科目分数都是 int 类型。

解决办法：查阅 C 语言使用手册，发现了 sscanf();内置函数，他与 scanf 函数类似，只不过 scanf 是从控制台读取，sscanf 是从字符数组中读出。因此我设置了一个用于缓冲的字符数组 input,在图形化界面通过对话框读取数据时，先读入 input 缓冲区中，然后再调用 sscanf 函数将 input 字符数组中的数字类型%d 读入对应的结构体变量中实现读入。且由于 sscanf 函数本身是具有返回值的，因此可以通过判断返回值是否是-1 来判断是否输入了正确的数据类型，从而可以很方便地进行数据入口检查。

问题 8：无法规整的在图形界面上输出学生数据

原因：EGE 的输出文字的函数很有限，既没有翻页功能也没有格式化输出功能（只有按照%d,%s 等格式输出，并不能很好地按照位数进行对齐）。

解决办法：最初是设置一个“下一页”按钮实现切换，但最后由于排列实在过于困难，因此选择了在控制台上调用 printf 进行格式输出。然后在图形化界面上输出提示语“请在控制台中查看”。

问题 9：虽然我调用了 printf 函数，但是控制台没有任何输出。

原因：在调用 EGE 库之后，控制台被屏蔽，因此无法调用控制台。

解决办法：在 main.cpp 第一行加入宏定义#define SHOW_CONSOLE 来调用控制台

问题 10：图形化界面上不同信息消失过快，无法起到提示的作用

原因：由于程序代码执行是按照行进行执行的，彼此之间没有时间延迟，因此输出提示之后就立刻清屏了。EGE 本身的 delay 函数只能起到延迟循环的作用。

解决办法：包含 window.h 头文件，然后调用 Sleep（）函数让程序暂停执行一段时间。

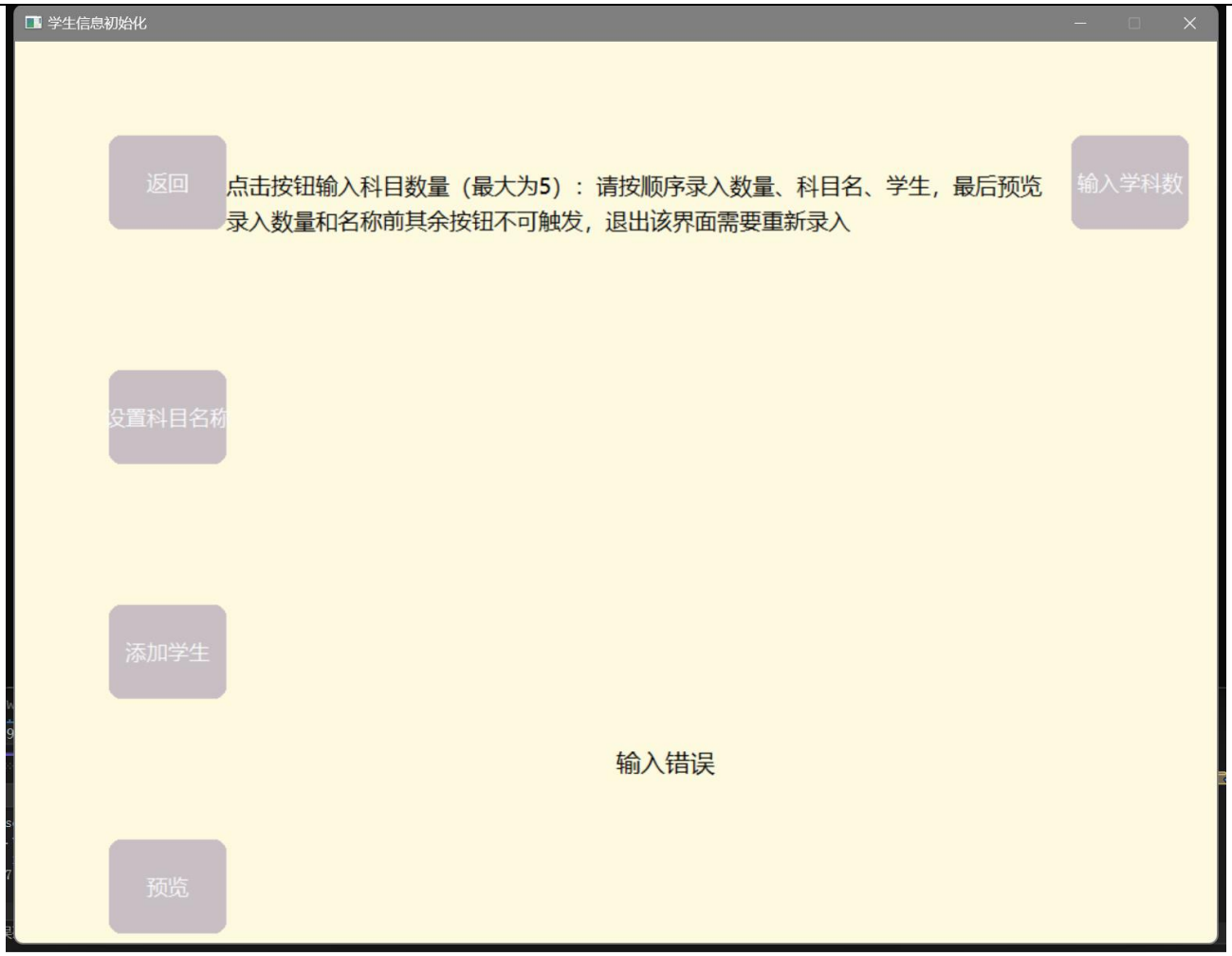
问题 11：初始化学生信息之后，再进入信息管理和查看界面，无法读取相关数据。

原因：初始化学生信息和信息管理和查看界面属于不同的函数，根据变量的生存域原理，变量会自动消失。

解决办法：为了保证在关闭程序之后仍能够进行信息管理和查看，且避免使用大量全局变量，在初始化学生信息，即在录入第一个学生信息之后对将数据存入 score.txt 文件中进行保存，下次进入信息管理与查看界面自动读入文件，添加、删除学生后自动保存文件。

测试用例和系统测试结果：

测试用例 1：
科目数量 6、 -1 、任何非数字字符。 目的：判断科目数量录入健壮性



测试用例 2：

科 目 数 5 科 目 名 称 123451234512345 abcdeabcdeabcdeabcde 123451234512345 abcdeabcdeabcdeabcde 123451234512345123451234512345

学生 1:2022111456 abcde 100 100 100 100 100

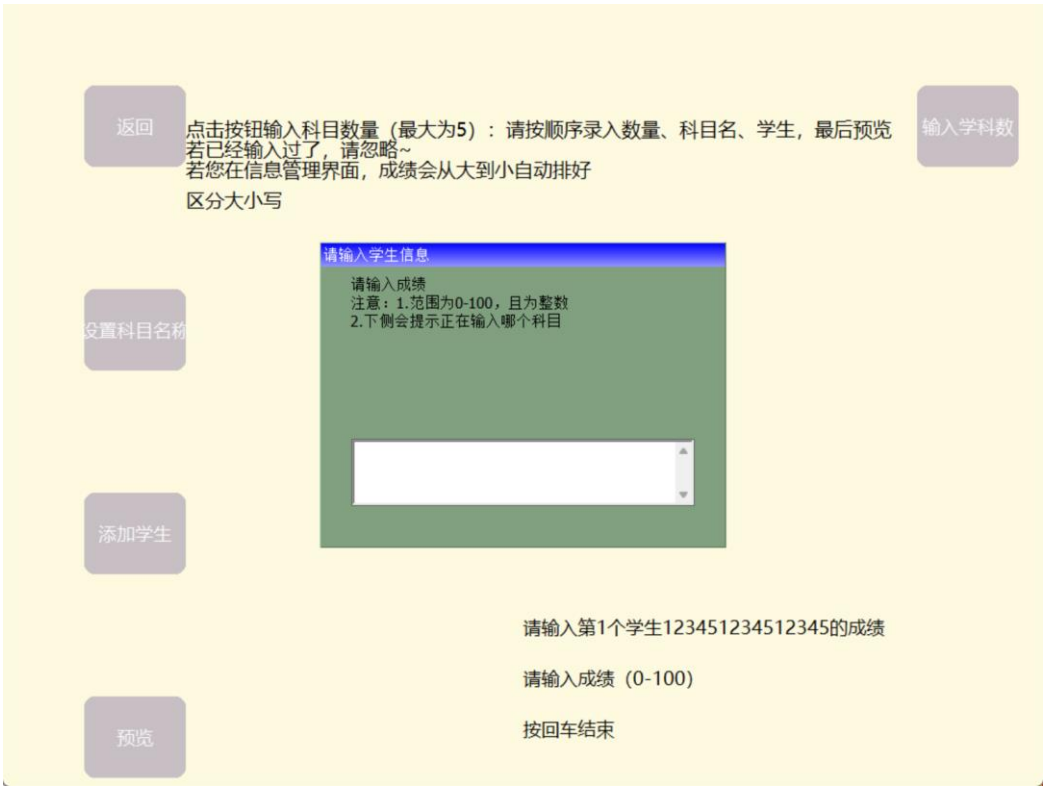
学生 2：202211145 abcdef 100 -1 98 98 98

学生 3:2022111 abcdefg a 100 100 100 96

学生 4:20221115 abcdefgh 61 61 61 61 61(后添加)

目的：科目名称为了判断输出时结构是否正确，学生 2、3 分别为了验证录入学生成绩时是否有输入数据检查，同时对学生姓名和学号按照子串的方式进行录入来验证模糊查找和精确查找的功能。

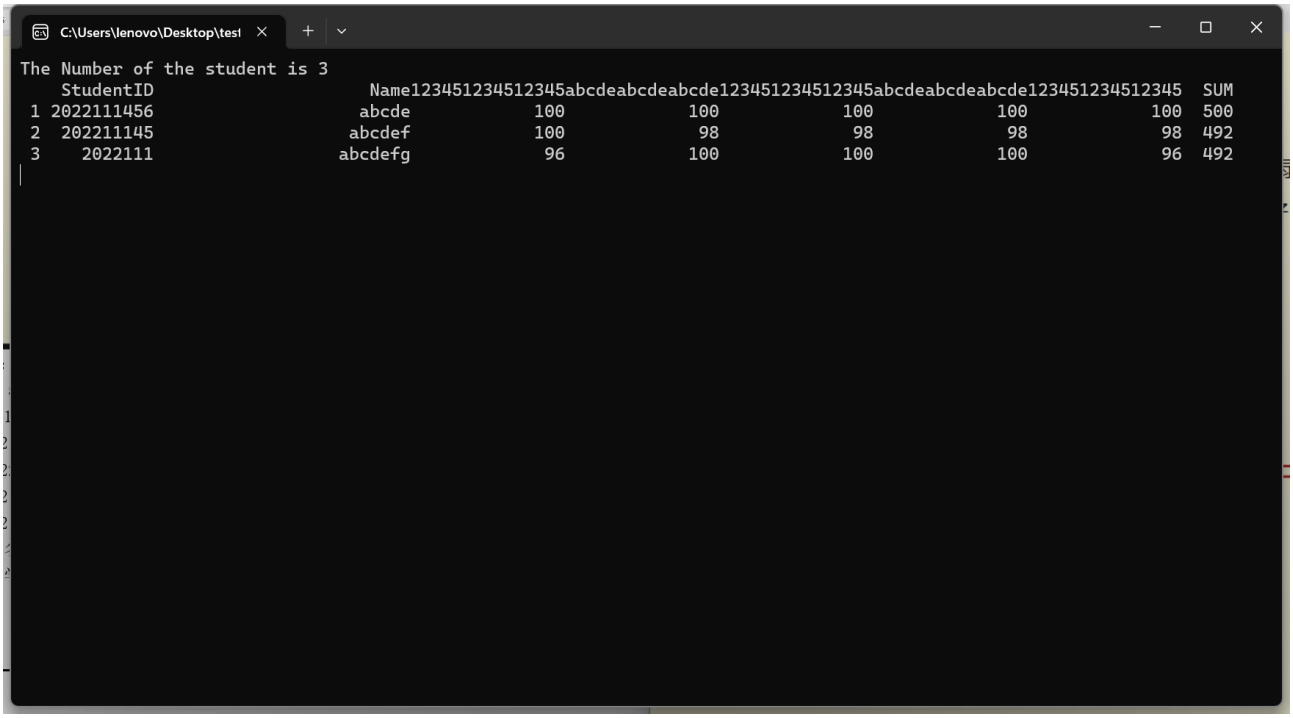

输入了5个学科
返回
点击按钮输入科目数量（最大为5）：请按顺序录入数量、科目名、学生，最后预览
若已经输入过了，请忽略~
若您在信息管理界面，成绩会从大到小自动排好
区分大小写
输入学科数
设置科目名称
添加学生
预览


返回
点击按钮输入科目数量（最大为5）：请按顺序录入数量、科目名、学生，最后预览
若已经输入过了，请忽略~
若您在信息管理界面，成绩会从大到小自动排好
区分大小写
输入学科数
请输入学生信息
请输入成绩
注意：1.范围为0-100，且为整数
2.下侧会提示正在输入哪个科目
设置科目名称
添加学生
请输入第1个学生123451234512345的成绩
请输入成绩（0-100）
按回车结束
预览

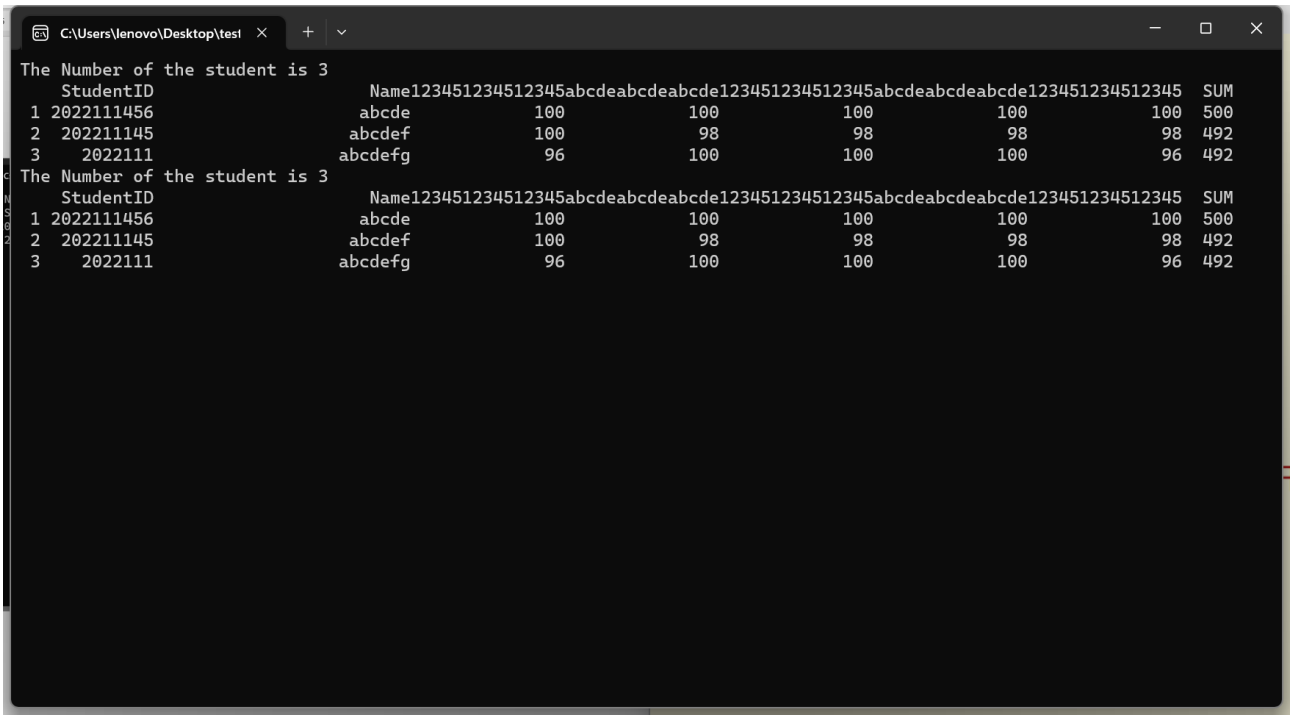

返回
点击按钮输入科目数量（最大为5）：请按顺序录入数量、科目名、学生，最后预览
若已经输入过了，请忽略~
若您在信息管理界面，成绩会从大到小自动排好
区分大小写
输入学科数
设置科目名称
添加学生
成绩输入范围有误
请输入第2个学生abcdeabcdeabcde的成绩
请输入成绩（0-100）
按回车结束
预览


返回
点击按钮输入科目数量（最大为5）：请按顺序录入数量、科目名、学生，最后预览
若已经输入过了，请忽略~
若您在信息管理界面，成绩会从大到小自动排好
区分大小写
输入学科数
设置科目名称
添加学生
成绩输入范围有误
请输入第3个学生123451234512345的成绩
请输入成绩（0-100）
按回车结束
预览

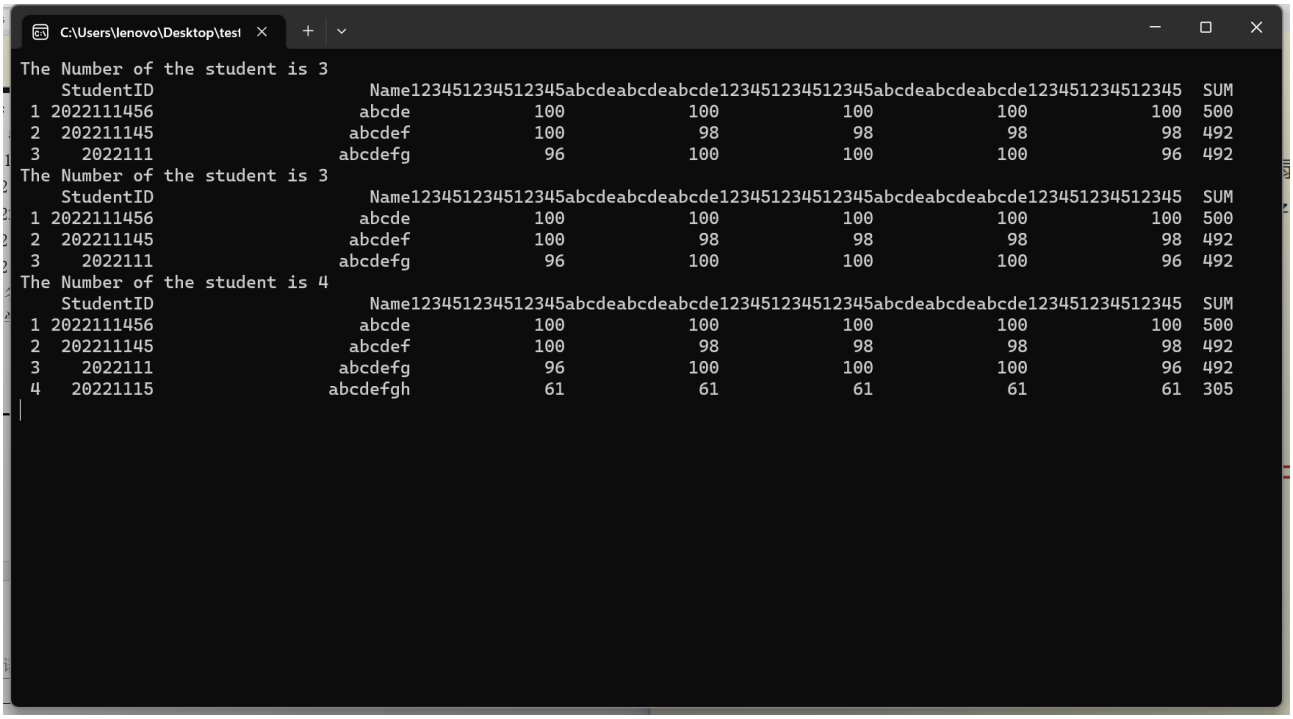

返回
进入该界面后，成绩自动排序为降序
按学号模糊查找
按姓名模糊查找
请输入学生信息
请输入学生学号
注意：1.请输入数字或英文
2.最大位数为10，多余会被忽略
添加学生成绩
按姓名精确查找
按学号精确查找
删除学生
请输入第4个学生
请输入学号
按回车结束
查看排名


C:\Users\lenovo\Desktop\test
The Number of the student is 3
StudentID Name123451234512345abcdeabcdeabcde123451234512345abcdeabcdeabcde123451234512345 SUM
1 2022111456 abcde 100 100 100 100 100 500
2 202211145 abcdef 100 98 98 98 98 492
3 2022111 abcdefg 96 100 100 100 96 492

排序结果



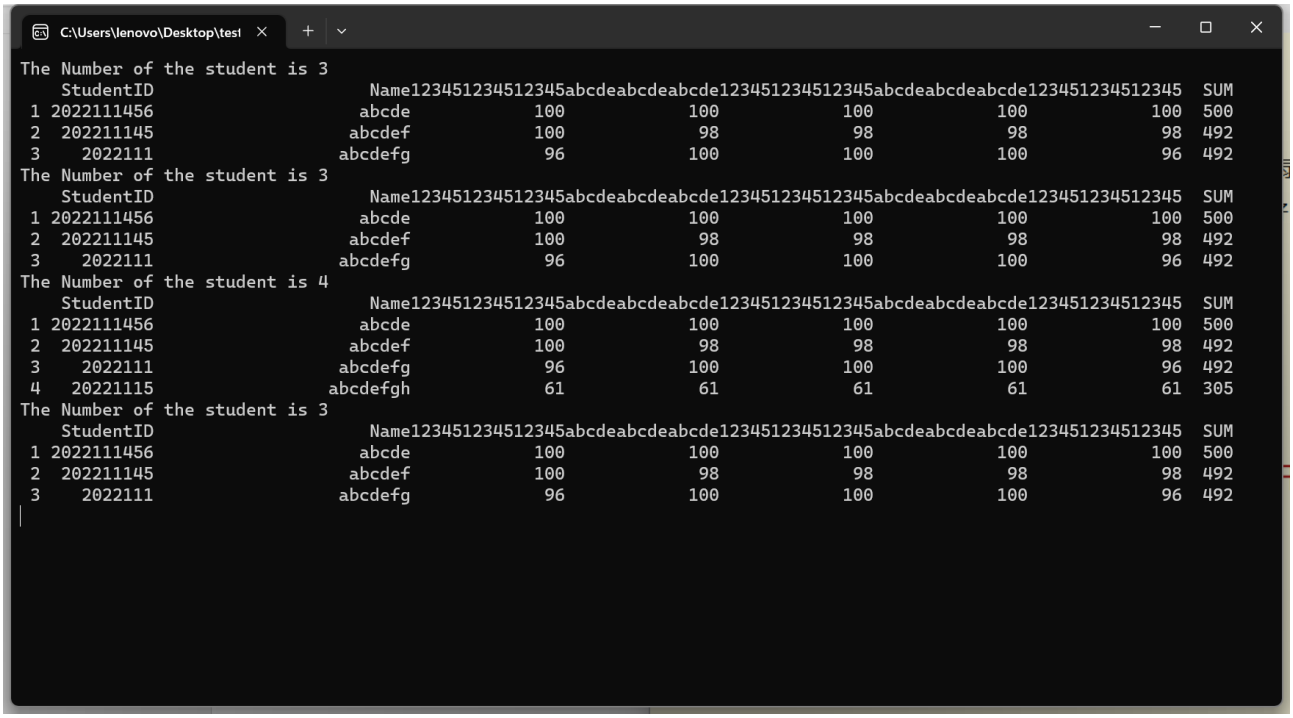
添加学生



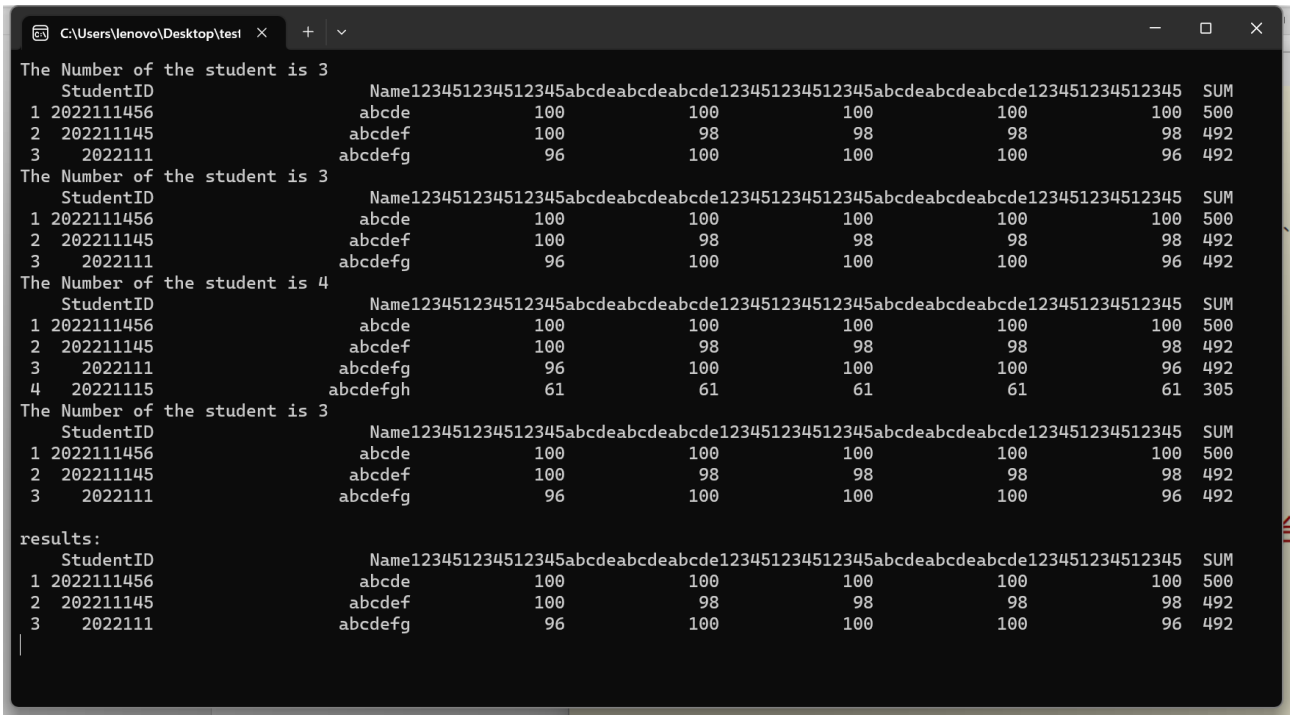
删除学生

输入错误的学号

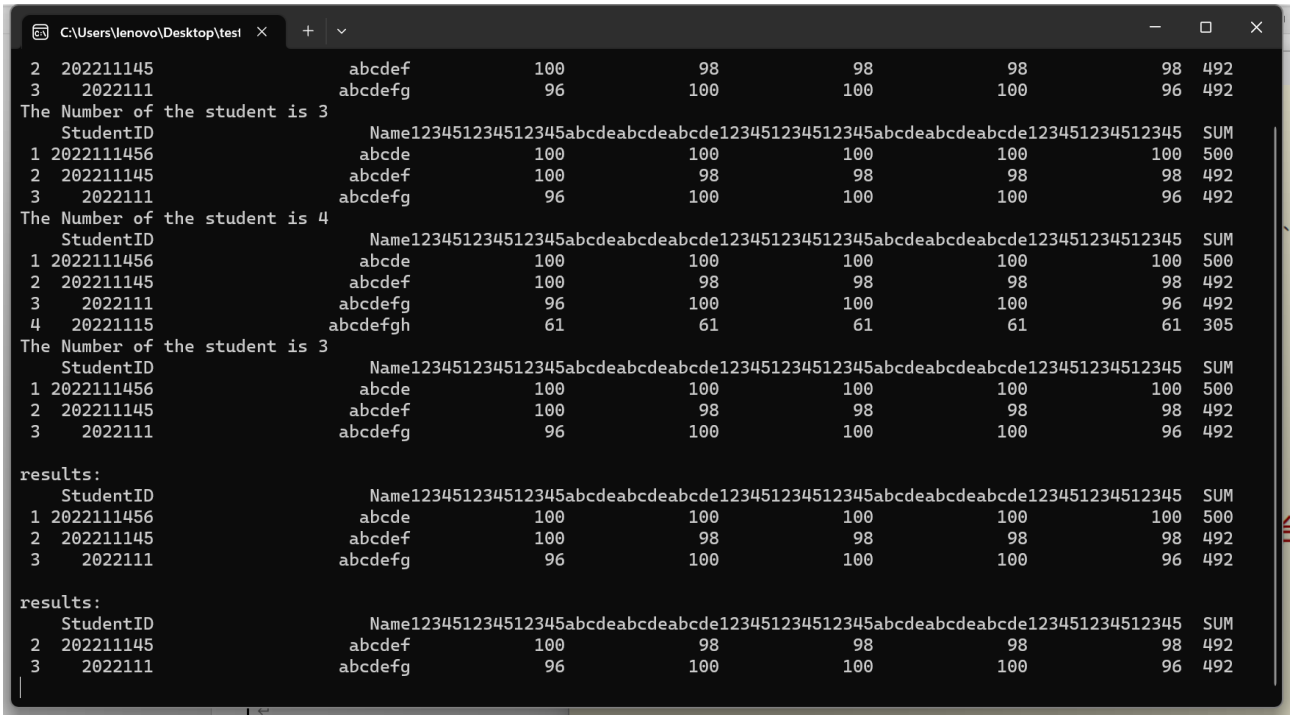
删除最后那个学生后



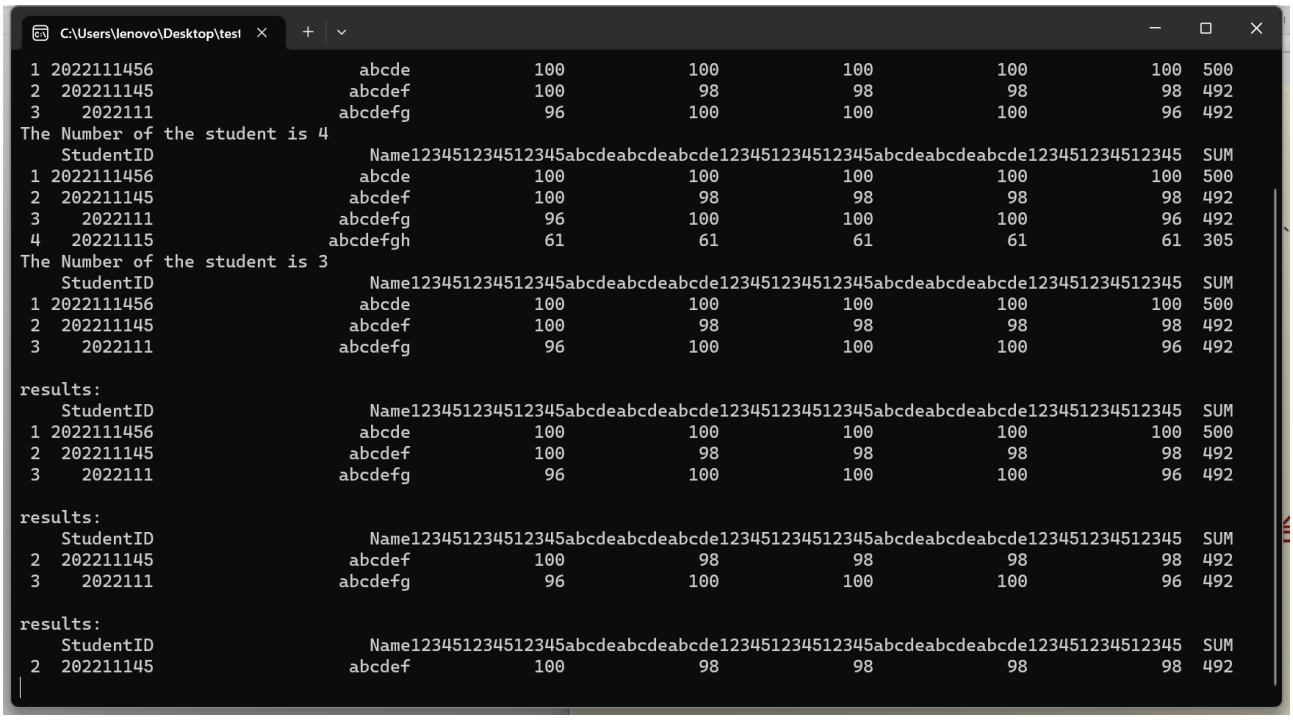
模糊查找 输入 2022111 学号



模糊查找 输入姓名 abcdef

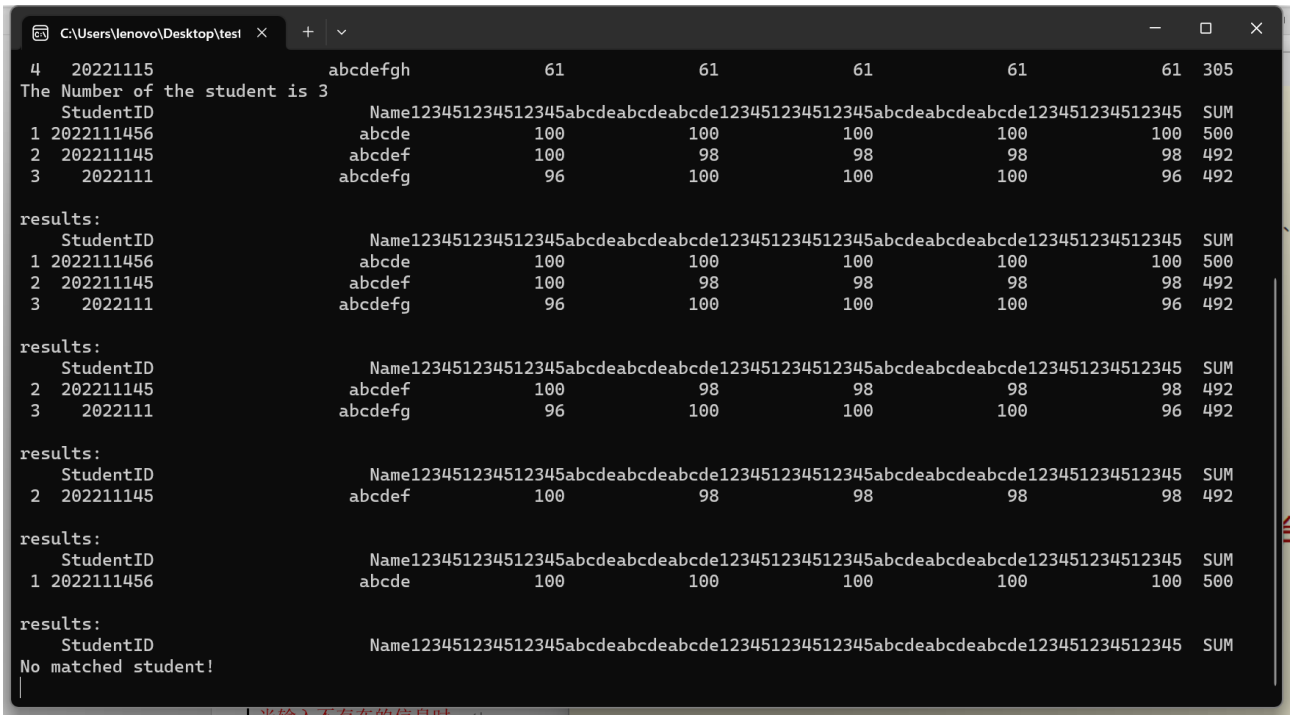


精确输入学号 202211145

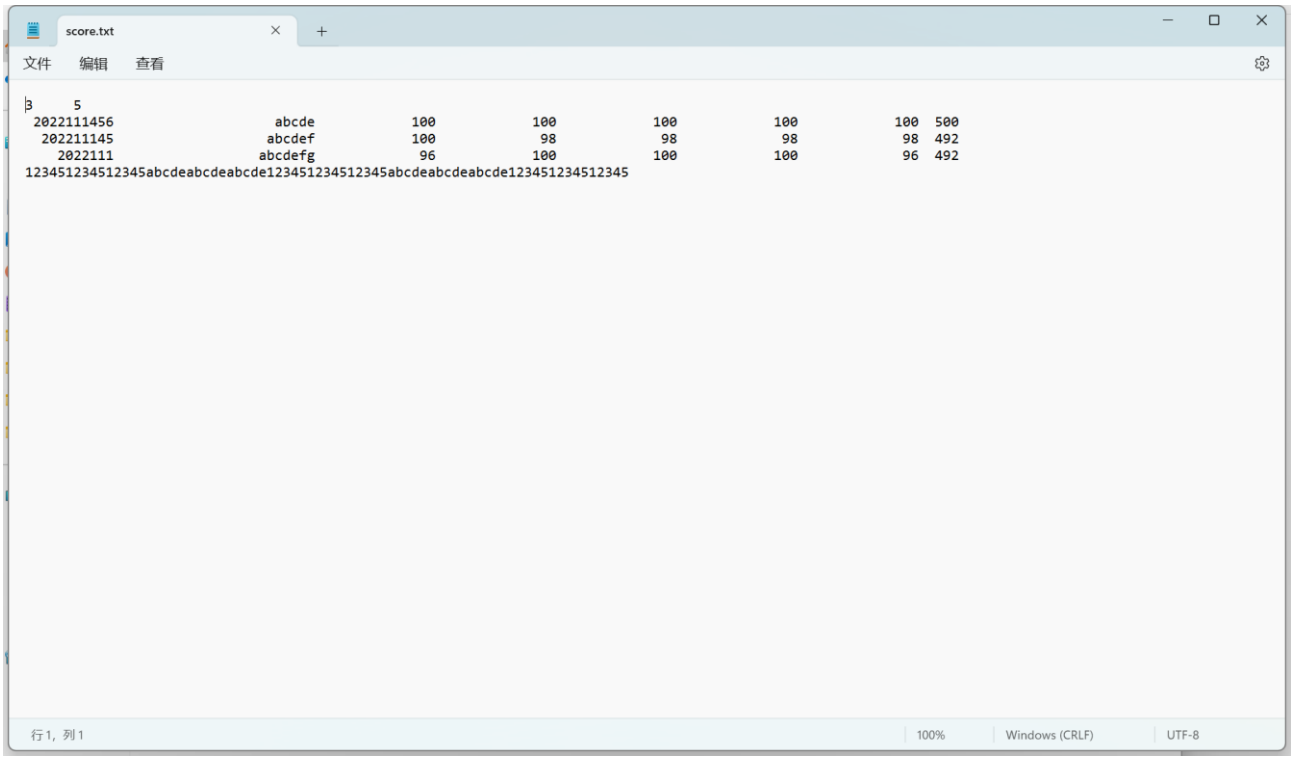


精确输入姓名 acbde

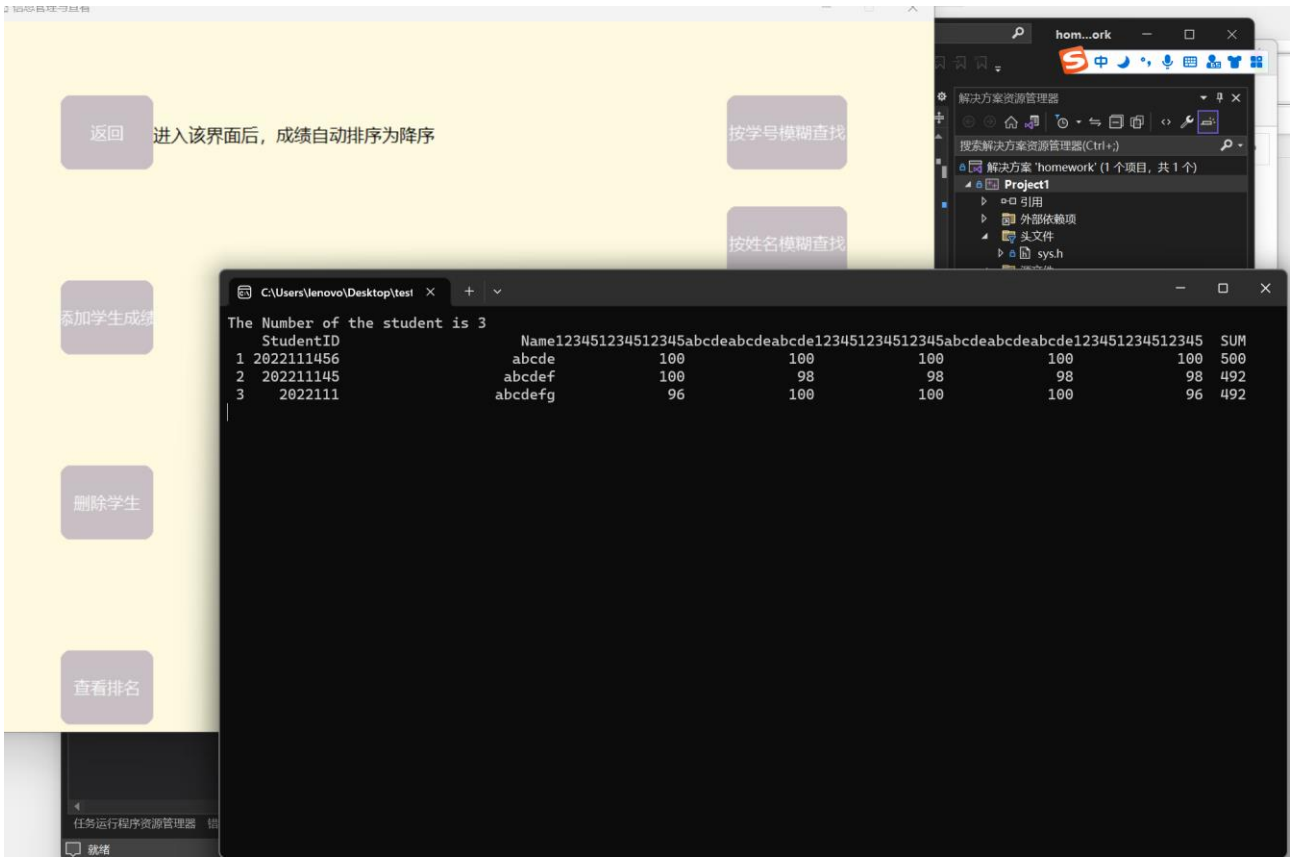
当输入不存在的信息时：



文件信息：



关闭程序，重新进入信息管理与查看界面查看信息，



程序的全部源代码：

注：建议直接运行源代码。

为了适应 word 排版，将部分较长代码进行换行，直接复制粘贴到 IDE 中可能导致编译错误！

sys.h

```
1.#pragma once
2.#define WINDOW_WIDTH 1024    // 窗口宽度
3.#define WINDOW_HEIGHT 768  // 窗口高度
4.
5.#define SUBJECTS_WIDTH 240 //文本框宽度
6.#define SUBJECTS_HEIGHT 80 //文本框高度
7.
8.#define BUTTON_WIDTH 240    // 按钮宽度
9.#define BUTTON_HEIGHT 80    // 按钮高度
10.    #define CORNERRADIUS 10    //圆角
11.
12.    #define BACK_WIDTH 100//返回按钮宽度
13.    #define BACK_HEIGHT 70//返回按钮高度
14.
15.    #define MAX_EXAM 5      //最大科目数量
16.    #define MAX_STU 50      //最大学生数量
17.
18.    #define MAX_SCORE 100//满分
```

```
#define WINDOW_WIDTH 1024    // 窗口宽度
#define WINDOW_HEIGHT 768  // 窗口高度
#define CORNERADIUS 10 //圆角
#define R 0xC7 //颜色常量 R
#define G 0xBF  //G
#define B 0xC4  //B
#define NUM 10 //一页最多 10 人
typedef struct RoundRectButton     //圆角矩形按钮
{
    int x, y; //坐标
    int width, height;  //宽度和高度
    float radius;//圆角
    int r, g, b;//颜色
    int fontsize;//字体大小
    char label[30];//按钮文字

}BUTTON;

typedef struct studentgrade
{
    int subject[MAX_EXAM];//每们学科的分数
    int sum;//总分
}stugrade;

typedef struct student
{
    char number[11];
    char name[21];
    stugrade grade;
}student;
typedef struct {
    int index;       // 学生索引
    int matchScore;  // 匹配得分
} SearchResult;
```

main.cpp

```
#define SHOW_CONSOLE
#include <graphics.h>
#include <string.h>
#include <conio.h>
#include <string.h>
#include <stdio.h>
#include <ege.h>
#include <math.h>
#include <ege/sys_edit.h>
```

```
#include <windows.h>
#include "sys.h"
#include<Windows.h>
#include<mmsystem.h>
//包含多媒体设备接口头文件
#include <cstdlib>
// 包含 rand 函数的头文件
#pragma comment (lib,"winmm.lib")
//加载静态库



// 全局变量

int STU_NUM = 0;//用户输入的学生数量

int SUB_NUM = 0;//学科数量
char subname[MAX_EXAM][16];//各个学科的名字
void studentEntryWindow();//学生数据录入窗口
void studentManagementWindow();//管理窗口
void mainWindow();//主窗口
void drawButton(const RoundRectButton* button);//创建按钮
bool click(const RoundRectButton* button, int x, int y);//判定按钮点击
void puttext(int x, int y, int size, const char* text);//输出文本
void initEditBox(sys_edit* editBox, int x, int y, int width, int height);//初始化编辑框
void draw_entry(int back_x,int back_y,BUTTON buttonarray[]);//绘制界面
int addstudent(student stu[],int count,int back_x,int back_y,BUTTON buttonarray[]);//添加学生
int delstudent(student stu[], int count, int back_x, int back_y, BUTTON buttonarray[]);//删除学生
void printstudent(student stu[], char subname[MAX_EXAM][16]);//输出成绩
void calculate_score(student stu[]);//计算学生总分
void write_file(student stu[]);//写入文件
void read_file(student stu[], int* STU_NUM, int* SUB_NUM);//读入文件
void swap(student* a, student* b);//快速排序相关，交换两数位置
int partition(student arr[], int low, int high);//快速排序相关，确定基准位置
void quickSort(student arr[], int low, int high);
//快速排序主体（因为要涉及很多次排序，
//所以使用相对效率较高的快速排序进行排序，减少资源占用）
int vaguesearch(const char* key, const char* str);//模糊搜索
int ranksearch(student stu[], int count, const char* input_number);//随机搜索



```

```
54.
55.
56.
57.  void studentManagementWindow()
58.  {
59.      initgraph(WINDOW_WIDTH, WINDOW_HEIGHT);// 初始化图形窗口
60.
61.      setbkcolor(EGERGB(0xFE, 0xFA, 0xE0));
62.
63.      setcaption("信息管理与查看");
64.
65.
66.
67.      int BackButton_x, BackButton_y; //”管理界面的返回按钮坐标“
68.      int AddInfoButton_x, AddInfoButton_y;//“管理界面的添加信息的按钮"
69.      int RemoveInfoButton_x, RemoveInfoButton_y;//"管理的删除信息的按钮"
70.
71.      int complete1button_x, complete1button_y; //"输入科目数量完成按钮"
72.
73.      int back_x = 80;
74.      int back_y = 80;
75.      BUTTON buttonarray[8];
76.      const int buffSize = 100;//输入缓冲区
77.      char strBuffer[100] = "";//输入缓冲区
78.      sys_edit editBox;//科目数量
79.      student stu[50]{}; //学生数量（最大 50）
80.      read_file(stu, &STU_NUM, &SUB_NUM);
81.      int count = STU_NUM;//学生数量，便于后续函数操作
82.      quickSort(stu, 0, count - 1);//对学生排序
83.      int i, j;
84.
85.
86.
87.      BUTTON back = {
88.          back_x,back_y,
89.          100,80,
90.          CORNERADIUS,
91.          R,G,B,
92.          28,
93.          "返回"
94.      }; buttonarray[0] = { back }; //返回按钮
95.      BUTTON number_search = {
96.          800,back_y,
97.          130,80,
```

```
98.             CORNERADIUS,
99.             R,G,B,
100.            28,
101.            "按学号模糊查找"
102.        }; buttonarray[1] = { number_search }; //学号搜索按钮
103.        BUTTON add_student = {
104.            back_x, back_y + 200,
105.            100,80,
106.            CORNERADIUS,
107.            R,G,B,
108.            28,
109.            "添加学生成绩"
110.        }; buttonarray[2] = { add_student }; //添加学生按钮
111.        BUTTON del_student = {
112.            back_x,back_y + 400,
113.            100,80,
114.            CORNERADIUS,
115.            R,G,B,
116.            28,
117.            "删除学生"
118.        }; buttonarray[3] = { del_student };//删除学生按钮
119.        BUTTON check ={
120.            back_x, back_y + 600,
121.            100,80,
122.            CORNERADIUS,
123.            R,G,B,
124.            28,
125.            "查看排名"
126.        }; buttonarray[4] = { check };//查看排名按钮
127.        BUTTON name_search = {
128.            800, back_y + 120,
129.            130,80,
130.            CORNERADIUS,
131.            R,G,B,
132.            28,
133.            "按姓名模糊查找"
134.        }; buttonarray[5] = { name_search };//姓名搜索按钮
135.        BUTTON acc_name_search = {
136.            800,2 * back_y + 160,
137.            130,80,
138.            CORNERADIUS,
139.            R,G,B,
140.            28,
141.            "按姓名精确查找"
```

```
142.        }; buttonarray[6] = { acc_name_search };//姓名搜索按钮
143.        BUTTON acc_number_search = {
144.            800,3 * back_y + 200,
145.            130,80,
146.            CORNERADIUS,
147.            R,G,B,
148.            28,
149.            "按学号精确查找"
150.        }; buttonarray[7] = { acc_number_search };
151.        //学号搜索按钮
152.
153.
154.        puttext(back_x + 100, back_y + 30, 25,
155.        "进入该界面后，成绩自动排序为降序");
156.
157.        for (i = 0; i < 8; i++)
158.        //绘制所有按钮
159.        {
160.            drawButton(&buttonarray[i]);
161.        }
162.
163.        // 无限循环，用于监听鼠标事件
164.        while (1)
165.        {
166.            // 获取鼠标消息
167.            mouse_msg msg = getmouse();
168.
169.            // 判断是否左键点击事件
170.            if (msg.is_left())
171.            {
172.                // 判断是否鼠标按下
173.                if (msg.is_down())
174.                {
175.                    // 点击了添加按钮
176.                    if (click(&add_student, msg.x, msg.y))
177.                    {
178.                        // 调用添加学生函数，返回更新后的学生数量
179.                        count = addstudent(stu, count, back_x, back_y, buttonarray);
180.
181.                        // 清空绘图设备
182.                        cleardevice();
183.
184.                        // 重新绘制界面
185.                        draw_entry(back_x, back_y, buttonarray);
```

```
186.
187.                     // 更新全局学生数量
188.                     STU_NUM = count;
189.
190.                     // 计算学生分数
191.                     calculate_score(stu);
192.
193.                     // 将学生信息写入文件
194.                     write_file(stu);
195.
196.                     // 更新全局学生数量
197.                     STU_NUM = count;
198.
199.                     // 对学生信息进行快速排序
200.                     quickSort(stu, 0, count - 1);
201.
202.                     // 将学生信息写入文件
203.                     write_file(stu);
204.                 }
205.                 // 点击了删除按钮
206.                 else if (click(&del_student, msg.x, msg.y))
207.                 {
208.                     // 调用删除学生函数，返回更新后的学生数量
209.                     count = delstudent(stu, count, back_x, back_y, buttonarray);
210.
211.                     // 更新全局学生数量
212.                     STU_NUM = count;
213.
214.                     // 对学生信息进行快速排序
215.                     quickSort(stu, 0, count - 1);
216.
217.                     // 将学生信息写入文件
218.                     write_file(stu);
219.                 }
220.                 // 点击了查看按钮
221.                 else if (click(&check, msg.x, msg.y))
222.                 {
223.                     // 输入文字提示
224.                     setcolor(RED);
225.                     setfont(35, 0, "微软雅黑");
226.                     xyprintf(WINDOW_WIDTH / 2, WINDOW_HEIGHT / 2, "请打开控制台查看");
227.
228.                     // 打印学生信息到控制台
229.                     printstudent(stu, subname);
```

```
230.                }
231.                // 点击了按学号搜索按钮
232.                else if (click(&number_search, msg.x, msg.y))
233.                {
234.                    char input[21];
235.
236.                    // 弹出输入框，获取要搜索的学号
237.                    inputbox_getline("请输入要搜索的学号", "请输入学生学号\n
238.                    注意：1.请输入数字或英文\n
239.                    2.最大位数为 10，多余会被忽略
240.                    \n3.该搜索为模糊搜索，搜索结果无先后顺序\n4.查询序号为排行榜序号",
241.                        input, 11);
242.
243.                    // 输入文字提示
244.                    setcolor(RED);
245.                    setfont(35, 0, "微软雅黑");
246.                    xyprintf(WINDOW_WIDTH / 2, WINDOW_HEIGHT / 2, "请打开控制台查看");
247.
248.                    int found = 0;
249.                    printf("\nresults:\n");
250.
251.                    // 打印表头
252.                    printf("%2s%11s%25s", "  ", "StudentID", "Name");
253.                    for (i = 0; i < SUB_NUM; i++)
254.                    {
255.                        printf("%15s", subname[i]);
256.                    }
257.                    printf("%5s", "SUM");
258.                    printf("\n");
259.
260.                    // 遍历学生信息，进行模糊搜索
261.                    for (i = 0; i < count; i++)
262.                    {
263.                        if (vaguesearch(input, stu[i].number) == 1)
264.                        {
265.                            found = 1;
266.                            // 输出学生信息
267.                            printf("%2d", i + 1);//输出序号
268.                            printf("%11s%25s", stu[i].number, stu[i].name);//姓名
269.                            for (int j = 0; j < SUB_NUM; j++)
270.                            {
271.                                printf("%15d", stu[i].grade.subject[j]);//科目分数
272.                            }
273.                            printf("%5d", stu[i].grade.sum);//总分
```

```
274.                                printf("\n");
275.                            }
276.                        }
277.
278.                        // 如果没有找到匹配的学生信息
279.                        if (found == 0)
280.                        {
281.                            printf("No matched student!\n");
282.                        }
283.                    }
284.                    // 点击了按姓名搜索按钮
285.                    else if (click(&name_search, msg.x, msg.y))
286.                    {
287.                        char input[21];
288.
289.                        // 弹出输入框，获取要搜索的姓名
290.                        inputbox_getline("请输入要搜索的姓名", "请输入学生姓名
291.                        \n 注意:
292.                        1.请输入数字或英文\n
293.                        2.最大位数为 20，多余会被忽略\n
294.                        3.该搜索为模糊搜索,搜索结果无先后顺序\
295.                        n4.查询序号为排行榜序号",
296.                            input, 21);
297.
298.                        // 输入文字提示
299.                        setcolor(RED);
300.                        setfont(35, 0, "微软雅黑");
301.                        xyprintf(WINDOW_WIDTH / 2, WINDOW_HEIGHT / 2,
302.                         "请打开控制台查看");
303.
304.                        int found = 0;
305.                        printf("\nresults:\n");
306.
307.                        // 打印表头
308.                        printf("%2s%11s%25s", "  ",
309.                        "StudentID", "Name");
310.                        for (i = 0; i < SUB_NUM; i++)
311.                        {
312.                            printf("%15s", subname[i]);
313.                        }
314.                        printf("%5s", "SUM");
315.                        printf("\n");
316.
317.                        // 遍历学生信息，进行模糊搜索
```

```
318.                     for (i = 0; i < count; i++)
319.                     {
320.                         if (vaguesearch(input, stu[i].name) == 1)
321.                         {
322.                             found = 1;
323.                             // 输出学生信息
324.                             printf("%2d", i + 1);//输出序号
325.                             printf("%11s%25s", stu[i].number, stu[i].name);//姓名
326.                             for (int j = 0; j < SUB_NUM; j++)
327.                             {
328.                                 printf("%15d", stu[i].grade.subject[j]);//科目分数
329.                             }
330.                             printf("%5d", stu[i].grade.sum);//总分
331.                             printf("\n");
332.                         }
333.                     }
334.
335.                     // 如果没有找到匹配的学生信息
336.                     if (found == 0)
337.                     {
338.                         printf("No matched student!\n");
339.                     }
340.                 }
341.                 // 点击了返回按钮
342.                 else if (click(&back, msg.x, msg.y))
343.                 {
344.                     // 清空绘图设备
345.                     cleardevice();
346.
347.                     // 返回主窗口
348.                     mainWindow();
349.
350.                     // 跳出循环
351.                     break;
352.                 }
353.                 // 点击了按姓名精准搜索按钮
354.                 else if (click(&acc_name_search, msg.x, msg.y))
355.                 {
356.                     char input[21];
357.                     int isfind = 0;
358.                     int pos = -1;
359.
360.                     // 弹出输入框，获取要搜索的姓名
361.                     inputbox_getline("请输入要搜索的姓名", "请输入学生姓名
```

```
                \n 注意：1.请输入数字或英文\n
                2.最大位数为 20，多余会被忽略\n
                3.该搜索为精确搜索\n
                4.查询序号为排行榜序号",
                    input, 21);

                // 输入文字提示
                setcolor(RED);
                setfont(35, 0, "微软雅黑");
                xyprintf(WINDOW_WIDTH / 2, WINDOW_HEIGHT / 2,
                 "请打开控制台查看");

                int found = 0;
                printf("\nresults:\n");

                // 打印表头
                printf("%2s%11s%25s", "  ", "StudentID", "Name");
                for (int i = 0; i < SUB_NUM; i++)
                {
                    printf("%15s", subname[i]);
                }
                printf("%5s", "SUM");
                printf("\n");
                for (int i = 0; i < count; i++)
                {
                    isfind = 0;
                    if (strcmp(stu[i].name, input) == 0)
                    // 完全相等 这是精确查找
                    {
                        pos = i;
                        isfind = 1;
                    }
                    if (isfind)
                    {
                        // 输出学生信息
                        printf("%2d", pos + 1);//输出序号
                        printf("%11s%25s", stu[pos].number, stu[pos].name);//姓名
                        for (int j = 0; j < SUB_NUM; j++)
                        {
                            printf("%15d", stu[pos].grade.subject[j]);//科目分数
                        }
                        printf("%5d", stu[pos].grade.sum);//总分
                        printf("\n");
                    }
```

```
406.                     }
407.                     // 如果没有找到匹配的学生信息
408.                     if (pos == -1)
409.                     {
410.                         printf("No matched student!\n");
411.                     }
412.                 }
413.                 // 点击了按学号精准搜索按钮
414.                 else if (click(&acc_number_search, msg.x, msg.y))
415.                 {
416.                     char input[21];
417.                     int isfind = 0;
418.                     int pos = -1;
419.
420.                     // 弹出输入框，获取要搜索的姓名
421.                     inputbox_getline("请输入要搜索的姓名", "请输入学生学号\n
422.                     注意：1.请输入数字或英文\n
423.                     2.最大位数为 20，多余会被忽略\n
424.                     3.该搜索为精确搜索\n4.查询序号为排行榜序号",
425.                         input, 21);
426.
427.                     // 输入文字提示
428.                     setcolor(RED);
429.                     setfont(35, 0, "微软雅黑");
430.                     xyprintf(WINDOW_WIDTH / 2, WINDOW_HEIGHT / 2, "请打开控制台查看");
431.
432.                     int found = 0;
433.                     printf("\nresults:\n");
434.
435.                     // 打印表头
436.                     printf("%2s%11s%25s", "  ", "StudentID", "Name");
437.                     for (int i = 0; i < SUB_NUM; i++)
438.                     {
439.                         printf("%15s", subname[i]);
440.                     }
441.                     printf("%5s", "SUM");
442.                     printf("\n");
443.
444.                     for (int i = 0; i < count; i++)
445.                     {
446.                         isfind = 0;
447.                         if (strcmp(stu[i].number, input) == 0) // 完全相等 这是精确查找
448.                         {
449.                             pos = i;
```

```
450.                                isfind = 1;
451.                            }
452.                            if (isfind)
453.                            {
454.                                // 输出学生信息
455.                                printf("%2d", pos + 1);//输出序号
456.                                printf("%11s%25s", stu[pos].number, stu[pos].name);//姓名
457.                                for (int j = 0; j < SUB_NUM; j++)
458.                                {
459.                                    printf("%15d", stu[pos].grade.subject[j]);//科目分数
460.                                }
461.                                printf("%5d", stu[pos].grade.sum);//总分
462.                                printf("\n");
463.                            }
464.                        }
465.                        // 如果没有找到匹配的学生信息
466.                        if (pos == -1)
467.                        {
468.                            printf("No matched student!\n");
469.                        }
470.                    }
471.                }
472.            }
473.        }
474.    }
475.    void drawButton(const RoundRectButton* button)
476.    {
477.        setfillcolor(EGEARGB(0xFF, R, G, B)); //填充颜色
478.        //绘制矩形
479.        ege_fillrect((float)(button->x + button->radius), (float)(button->y),
480.            (float)(button->width - 2 * button->radius), float(button->height)
481.        );
482.
483.        ege_fillrect((float)(button->x), (float)(button->y + button->radius),
484.            (float)(button->radius), (float)(button->height - 2 * button->radius)
485.        );
486.
487.        ege_fillrect((float)(button->x + button->width - button->radius),
488.            (float)(button->y + button->radius),
489.            (float)(button->radius), (float)(button->height - 2 * button->radius)
490.        );
491.
492.        //确定圆角半径和坐标
493.        float diameter = 2 * button->radius;
```

```
494.      float dx = button->width - diameter;
495.      float dy = button->height - diameter;
496.
497.      //绘制四个圆角
498.      ege_fillpie((float)(button->x + dx), (float)(button->y + dy),
499.      diameter, diameter, 0.0f, 90.0f);
500.      ege_fillpie((float)(button->x), (float)(button->y + dy),
501.       diameter, diameter, 90.0f, 90.0f);
502.      ege_fillpie((float)(button->x), (float)(button->y),
503.      diameter, diameter, 180.0f, 90.0f);
504.      ege_fillpie((float)(button->x + dx), (float)(button->y),
505.       diameter, diameter, 270.0f, 90.0f);
506.      //填充按钮文字
507.      setfillcolor(EGEARGB(0xff, 0x40, 0x40, 0x40));
508.      setbkmode(TRANSPARENT);
509.      setcolor(WHITE);
510.      setfont(24, 0, "微软雅黑");
511.      outtextxy(button->x + button->width / 2 - textwidth(button->label) / 2,
512.      button->y + button->height / 2 - textheight(button->label) / 2,
513.       button->label);
514.
515.  }
516.  bool click(const RoundRectButton* button, int x, int y)
517.  {
518.      bool inside = false;
519.
520.      // 点在包围矩形内
521.      if ((x >= button->x) && (y >= button->y)
522.          && (x < button->x + button->width)
523.          && (y < button->y + button->height)
524.          ) {
525.
526.          float centerx = button->x + button->width / 2.0f;
527.          float centery = button->y + button->height / 2.0f;
528.          float dx = (float)fabs(x - centerx);
529.          float dy = (float)fabs(y - centery);
530.          float interWidth = button->width / 2.0f - button->radius;
531.          float interHeight = button->height / 2.0f - button->radius;
532.
533.          // 点不在圆角空白处 ()
534.          if (!((dx > interWidth)
535.              && (dy > interHeight)
536.              && ((dx - interWidth) * (dx - interWidth) + (dy - interHeight) * (dy - interHeight)
537.                      > button->radius * button->radius)
```

```
538.                 )
539.                 ) {
540.                 inside = true;
541.             }
542.         }
543.
544.         return inside;
545.     }
546.     void puttext(int x,int y,int size ,const char * text)
547.     {
548.         setfillcolor(EGEARGB(0xff, 0x40, 0x40, 0x40));
549.         setbkmode(TRANSPARENT);
550.         setcolor(BLACK);
551.         setfont(size, 0, "微软雅黑");
552.         xyprintf(x, y , "%s",text);
553.     }
554.     void initEditBox(sys_edit* editBox, int x, int y, int width, int height)
555.     {
556.         editBox->create(false);
557.         editBox->size(width, height + 8);
558.         editBox->setbgcolor(WHITE);
559.         editBox->setfont(24, 0, "黑体");
560.         editBox->move(x, y);
561.         editBox->visable(true);
562.         editBox->setfocus();
563.     }
564.     void mainWindow()
565.     {
566.         initgraph(WINDOW_WIDTH, WINDOW_HEIGHT);// 初始化图形窗口
567.
568.         setbkcolor(EGERGB(0xFE, 0xFA, 0xE0));
569.         setcaption("学生信息管理系统");
570.         int button1_x, button1_y;  // “学生信息录入”按钮坐标
571.         int button2_x, button2_y;  // “学生信息管理”按钮坐标 z
572.         // 计算按钮坐标
573.         button1_x = WINDOW_WIDTH / 2 - BUTTON_WIDTH / 2;
574.         button1_y = WINDOW_HEIGHT / 2 - BUTTON_HEIGHT - 40;
575.         button2_x = WINDOW_WIDTH / 2 - BUTTON_WIDTH / 2;
576.         button2_y = WINDOW_HEIGHT / 2 + 40;
577.
578.         //按钮结构体
579.         BUTTON button1 = {
580.             button1_x,button1_y,
581.             BUTTON_WIDTH,BUTTON_HEIGHT,
```

```
582.            CORNERADIUS,
583.            R,G,B,
584.            28,
585.            "学生信息初始化"
586.        };
587.
588.        BUTTON button2 = {
589.            button2_x,button2_y,
590.            BUTTON_WIDTH,BUTTON_HEIGHT,
591.            CORNERADIUS,
592.            R,G,B,
593.            28,
594.            "信息管理与查看"
595.        };
596.        drawButton(&button1);
597.        drawButton(&button2);
598.        int clickbutton1 = 0;
599.        int clickbutton2 = 0;
600.        //文字说明
601.        puttext(100, 100, 35, "欢迎使用学生管理系统");
602.        puttext(100, 135, 35, "第一次录入成绩请点第一个按钮，添加、排序、查找请进入管理界面");
603.        puttext(100, 170, 35, "需要重新录入信息时请点击第一个按钮初始化信息");
604.        puttext(100, 205, 35, "可以阅读文件夹中的 README 来了解使用须知");
605.        //这段代码是一个鼠标事件处理的无限循环。
606.        //程序会不断获取鼠标消息并根据鼠标事件的类型进行相应的操作。代码逻辑如下：
607.
608.        /*当鼠标左键被点击时，进入判断。
609.            如果鼠标左键是按下状态，通过调用 click()函数检测鼠标点击的位置是否在按钮 1 或按钮 2 上，
610.            并将相应的布尔值赋给 clickbutton1 和 clickbutton2。
611.            如果鼠标左键是抬起状态，根据 clickbutton1 和 clickbutton2 的值进行判断：
612.            如果 clickbutton1 为真，则清除屏幕内容并显示学生录入窗口。
613.            如果 clickbutton2 为真，则清除屏幕内容并显示学生管理窗口。
614.            如果以上两个条件都不满足，则继续下一次循环。
615.            最后，通过 delay()函数延时 10 毫秒，以避免过于频繁的循环。*/
616.
617.        while (1) {
618.            // 获取鼠标消息
619.            mouse_msg msg = getmouse();
620.
621.            // 判断鼠标左键是否点击
622.            if (msg.is_left()) {
623.                // 判断鼠标左键是否按下
624.                if (msg.is_down()) {
625.                    // 检测点击的按钮
```

```
626.                 clickbutton1 = click(&button1, msg.x, msg.y);
627.                 clickbutton2 = click(&button2, msg.x, msg.y);
628.             }
629.             // 鼠标左键抬起
630.             else {
631.                 // 如果按钮 1 被点击
632.                 if (clickbutton1) {
633.                     // 清除屏幕内容
634.                     cleardevice();
635.                     // 显示学生录入窗口
636.                     studentEntryWindow();
637.                 }
638.                 // 如果按钮 2 被点击
639.                 else if (clickbutton2) {
640.                     // 清除屏幕内容
641.                     cleardevice();
642.                     // 显示学生管理窗口
643.                     studentManagementWindow();
644.                 }
645.                 // 没有按钮被点击，继续循环
646.                 else {
647.                     continue;
648.                 }
649.             }
650.         }
651.         // 延时 10 毫秒，避免过于频繁的循环
652.         delay(10);
653.     }
654. 
655.     closegraph();
656. }
657. void studentEntryWindow()
658. {
659.     initgraph(WINDOW_WIDTH, WINDOW_HEIGHT);  // 初始化图形窗口
660. 
661.     setbkcolor(EGERGB(0xFE,0xFA,0xE0));
662. 
663.     setcaption("学生信息初始化");
664. 
665.     int BackButton_x, BackButton_y; //”录入界面的返回按钮坐标“
666.     int AddInfoButton_x, AddInfoButton_y;//“录入界面的添加信息的按钮"
667. 
668.     int complete1button_x, complete1button_y; //"输入科目数量完成按钮"
669. 
```

```
670.      int back_x = 80;
671.      int back_y = 80;
672.      BUTTON buttonarray[6];
673.      const int buffSize = 100;//输入缓冲区
674.      char strBuffer[100] = "";//输入缓冲区
675.      sys_edit editBox;//科目数量
676.      student stu[50]{}; //学生数量（最大 50）
677.      int count = 0;//录入的学生数量
678.      int i, j;
679.
680.      BUTTON back = {
681.          back_x,back_y,
682.          100,80,
683.          CORNERADIUS,
684.          R,G,B,
685.          28,
686.          "返回"
687.      }; buttonarray[0] = { back }; //返回按钮
688.
689.      BUTTON confirm = {
690.          900,back_y,
691.          100,80,
692.          CORNERADIUS,
693.          R,G,B,
694.          28,
695.          "输入学科数"
696.      }; buttonarray[1] = { confirm }; //确定按钮
697.
698.      BUTTON add_student = {
699.          back_x, back_y + 400,
700.          100,80,
701.          CORNERADIUS,
702.          R,G,B,
703.          28,
704.          "添加学生"
705.      }; buttonarray[2] = { add_student }; //添加学生按钮
706.      BUTTON set_sub = {
707.          back_x,back_y + 200,
708.          100,80,
709.          CORNERADIUS,
710.          R,G,B,
711.          28,
712.          "设置科目名称"
713.      }; buttonarray[3] = { set_sub };
```

```
714.        BUTTON check =
715.        {
716.            back_x, back_y + 600,
717.            100,80,
718.            CORNERADIUS,
719.            R,G,B,
720.            28,
721.            "预览"
722.        }; buttonarray[4] = { check };
723.
724.        puttext(back_x + 100, back_y + 30, 25, "点击按钮输入科目数量（最大为 5）：
725.        请按顺序录入数量、科目名、学生，最后预览");
726.        puttext(back_x + 100, back_y + 60, 25, "录入数量和名称前其余按钮不可触发，
727.        退出该界面需要重新录入");
728.
729.        for (i = 0; i < 6; i++)                        //绘制所有按钮
730.        {
731.            drawButton(&buttonarray[i]);
732.        }
733.
734.        ;
735.        //第一次帧循环，确定录入科目数量
736.        while (1) {
737.            int flag = 0;
738.            int is_confirm = 0;
739.            mouse_msg msg = getmouse();
740.            //判断鼠标左键点击（左键按下确定位置，抬起为执行时刻）
741.            if (msg.is_left()) {
742.                if (msg.is_down()) {
743.                    //检测点击的按钮
744.                    if (click(&confirm, msg.x, msg.y))
745.                    {
746.                        //第一次输入
747.                        {
748.                            inputbox_getline("请输入科目数量", "请输入科目数量\n
749.                            注意：1.请输入数字\n
750.                            2.最大科目数为 5\n
751.                            3.按回车结束输入",
752.                                strBuffer, buffSize);
753.                            int res = sscanf(strBuffer, "%d", &SUB_NUM);
754.                            if (res != 1 || SUB_NUM > 5 || SUB_NUM <= 0)
755.                            {
756.                                //输入提示信息
757.                                setfillcolor(EGEARGB(0xff, 0x40, 0x40, 0x40));
```

```
758.                                setbkmode(TRANSPARENT);
759.                                setcolor(BLACK);
760.                                setfont(28, 0, "微软雅黑");
761.                                xyprintf(WINDOW_WIDTH / 2, 600, "输入错误");
762.                                Sleep(100);
763.                                cleardevice();
764.                                draw_entry(back_x, back_y, buttonarray);
765.                                flag = 1;
766.                                continue;
767.
768.                            }
769.                            else
770.                            {
771.                                cleardevice();
772.                                draw_entry(back_x, back_y, buttonarray);
773.
774.
775.                                //输入提示信息
776.                                setfillcolor(EGEARGB(0xff, 0x40, 0x40, 0x40));
777.                                setbkmode(TRANSPARENT);
778.                                setcolor(BLACK);
779.                                setfont(28, 0, "微软雅黑");
780.                                xyprintf(800, back_y - 40, "输入了%d 个学科", SUB_NUM);
781.                                break;
782.                            }
783.                        }
784.                    }
785.                    else if (click(&back, msg.x, msg.y))
786.                    {
787.                        editBox.destory();
788.                        cleardevice();
789.                        mainWindow();
790.                        break;
791.                    }
792.                }
793.            }
794.            delay(10);
795.        }
796.            //检测到 confirm 按钮点击就标记为输入完毕
797.
798.    //设置科目
799.        while (1) {
800.            mouse_msg msg = getmouse();
801.            //判断鼠标左键点击（左键按下确定位置，抬起为执行时刻）
```

```
802.          if (msg.is_left()) {
803.              if (msg.is_down()) {
804.                  //检测点击的按钮
805.                  //是否返回
806.              //设置科目
807.                  if (click(&set_sub, msg.x, msg.y))
808.                  {
809.                      for (i = 0; i < SUB_NUM; i++)
810.                      {
811.                          cleardevice();//清屏重绘
812.                          draw_entry(back_x, back_y, buttonarray);
813.
814.                          /*setfillcolor(EGEARGB(0xff, 0x40, 0x40, 0x40));
815.                          setbkmode(TRANSPARENT);*/
816.                          setcolor(BLACK);
817.                          setfont(28, 0, "微软雅黑");
818.                          xyprintf(WINDOW_WIDTH / 2, 600, "请输入第%d 个学科", i + 1);
819.                          xyprintf(WINDOW_WIDTH / 2, 650, "请输入英文名称");
820.                          xyprintf(WINDOW_WIDTH / 2, 700, "按回车结束");
821.
822.                          inputbox_getline("请输入科目名称", "请输入科目名称\n 注意:
823.                          1.请输入英文\n
824.                          2.下侧会提示正在输入第几个科目\n
825.                          3.字符长度不超过 15",
826.                              subname[i], 16);
827.
828.                          cleardevice();//清屏重绘
829.                          delay(200);
830.                      }
831.                      draw_entry(back_x, back_y, buttonarray);
832.                      break;
833.                  }
834.                  //返回
835.                  else if (click(&back, msg.x, msg.y))
836.                  {
837.                      cleardevice();
838.                      mainWindow();
839.                      break;
840.                  }
841.              }
842.          }
843.      }
844.  //录入学生与查看学生
845.      while (1)
```

```
846.      {
847.          mouse_msg msg = getmouse();
848.          //判断鼠标左键点击（左键按下确定位置）
849.          if (msg.is_left())
850.          {
851.              if (msg.is_down())
852.              {
853.                  //添加学生
854.                  if (click(&add_student, msg.x, msg.y))
855.                  {
856.                      count = addstudent(stu, count, back_x, back_y, buttonarray);
857.                      cleardevice();
858.                      draw_entry(back_x, back_y, buttonarray);
859.                      STU_NUM = count;
860.                      calculate_score(stu);
861.                      write_file(stu);
862.
863.                  }
864.                  //预览
865.                  else if (click(&check, msg.x, msg.y))
866.                  {
867.                      setcolor(RED);
868.                      setfont(35, 0, "微软雅黑");
869.                      xyprintf(WINDOW_WIDTH / 2, WINDOW_HEIGHT / 2, "请打开控制台查看");
870.                      printstudent(stu,subname);
871.                  }
872.                  //是否返回
873.                  else if (click(&back, msg.x, msg.y))
874.                  {
875.                      cleardevice();
876.                      mainWindow();
877.                      break;
878.                  }
879.              }
880.          }
881.      }
882.  }
883.  void draw_entry(int back_x, int back_y, BUTTON buttonarray[])
884.  {
885.    // 初始化图形窗口
886.      int i;
887.      setbkcolor(EGERGB(0xFE, 0xFA, 0xE0));
888.      puttext(back_x + 100, back_y + 30, 25, "点击按钮输入科目数量（最大为 5）：
889.      请按顺序录入数量、科目名、学生，最后预览");
```

```
890.        puttext(back_x + 100, back_y + 50, 25, "若已经输入过了，请忽略~");
891.        puttext(back_x + 100, back_y + 70, 25, "若您在信息管理界面，成绩会从大到小自动排好");
892.        puttext(back_x + 100, back_y + 100, 25, "区分大小写");
893.        for (i = 0; i < 8; i++)                        //绘制所有按钮
894.        {
895.            drawButton(&buttonarray[i]);
896.
897.        }
898.        setfillcolor(EGEARGB(0xff, 0x40, 0x40, 0x40));//文字
899.        setbkmode(TRANSPARENT);
900.        setcolor(BLACK);
901.        setfont(28, 0, "微软雅黑");
902.    }
903.    // 函数名称：addstudent
904.    // 功能：添加学生信息
905.    // 参数：
906.    // - stu：学生信息数组
907.    // - count：当前已添加的学生数量
908.    // - back_x：返回按钮的 x 坐标
909.    // - back_y：返回按钮的 y 坐标
910.    // - buttonarray：按钮数组
911.    // 返回值：更新后的学生数量
912.    int addstudent(student stu[], int count, int back_x, int back_y, BUTTON buttonarray[])
913.    {
914.        int i;
915.        int add_flag = 0; // 添加标志，用于记录是否成功添加学生信息
916.        char num[10]; // 学号输入的字符数组
917.
918.        // 录入学生学号
919.        setcolor(BLACK);
920.        setfont(28, 0, "微软雅黑");
921.        xyprintf(WINDOW_WIDTH / 2, 600, "请输入第%d 个学生", count + 1);
922.        xyprintf(WINDOW_WIDTH / 2, 650, "请输入学号");
923.        xyprintf(WINDOW_WIDTH / 2, 700, "按回车结束");
924.
925.        // 获取学生学号输入
926.        inputbox_getline("请输入学生信息", "请输入学生学号\n
927.        注意：1.请输入数字或英文\n
928.        2.最大位数为 10，多余会被忽略",
929.            stu[count].number, 11);
930.
931.        // 清屏重绘
932.        cleardevice();
933.        draw_entry(back_x, back_y, buttonarray);
```

```
934.
935.
936.        // 录入学生姓名
937.        setcolor(BLACK);
938.        setfont(28, 0, "微软雅黑");
939.        xyprintf(WINDOW_WIDTH / 2, 600, "请输入第%d 个学生", count + 1);
940.        xyprintf(WINDOW_WIDTH / 2, 650, "请输入姓名");
941.        xyprintf(WINDOW_WIDTH / 2, 700, "按回车结束");
942.
943.
944.        // 获取学生姓名输入
945.        inputbox_getline("请输入学生信息", "请输入学生姓名\n
946.        注意：1.请输入英文\n
947.        2.名字最长为 20 个字符，多余会被忽略",
948.            stu[count].name, 21);
949.
950.        cleardevice();
951.        draw_entry(back_x, back_y, buttonarray);
952.
953.        // 按顺序录入学生成绩
954.        for (i = 0; i < SUB_NUM; i++)
955.        {
956.            cleardevice();
957.            draw_entry(back_x, back_y, buttonarray);
958.            setcolor(BLACK);
959.            setfont(24, 0, "微软雅黑");
960.            xyprintf(WINDOW_WIDTH / 2, 600, "请输入第%d 个学生%s 的成绩", count + 1, subname[i]);
961.            xyprintf(WINDOW_WIDTH / 2, 650, "请输入成绩 (0-100) ");
962.            xyprintf(WINDOW_WIDTH / 2, 700, "按回车结束");
963.
964.            // 获取学生成绩输入
965.            inputbox_getline("请输入学生信息", "请输入成绩\n
966.            注意：1.范围为 0-100，且为整数\n
967.            2.下侧会提示正在输入哪个科目",
968.                num, sizeof(stu[count].name) / sizeof(char));
969.            int res = sscanf(num, "%d", &stu[count].grade.subject[i]); // 整型格式录入
970.
971.
972.            if (res != 1 || stu[count].grade.subject[i] < 0 || stu[count].grade.subject[i] > MAX_SCORE)
973.            {
974.                setcolor(BLACK);
975.                setfont(24, 0, "微软雅黑");
976.                xyprintf(WINDOW_WIDTH / 2, 550, "成绩输入范围有误");
977.                i--;
```

```
978.                 Sleep(200);
979.                 cleardevice(); // 清屏
980.                 draw_entry(back_x, back_y, buttonarray);
981.                 continue;
982.             }
983.
984.             delay(200);
985.         }
986.
987.         count++; // 更新学生数量
988.         return count;
989.     }
990.     // 函数名: delstudent
991.     // 功能: 删除学生信息
992.     // 参数:
993.     //   - stu: 学生信息数组
994.     //   - count: 学生信息数组的大小
995.     //   - back_x, back_y: 返回按钮的坐标
996.     //   - buttonarray: 按钮数组
997.     // 返回值: 删除后学生信息数组的大小
998.     int delstudent(student stu[], int count, int back_x, int back_y, BUTTON buttonarray[]) {
999.         int pos = -1;
1000.        char input_number[11];
1001.        int i;
1002.
1003.        xyprintf(WINDOW_WIDTH / 2, 650, "学号应完全匹配，并非模糊搜索,一次只能删除一个");
1004.        xyprintf(WINDOW_WIDTH / 2, 700, "按回车结束");
1005.
1006.        inputbox_getline("请输入学生学号", "请输入学生学号\n
1007.        注意：1.请输入数字或英文\n
1008.        2.最大位数为 10,多余会被忽略\n
1009.        3.一次只能删除一个学生\n4.请精确输入学号，按回车结束",
1010.            input_number, 11);
1011.
1012.
1013.        pos = ranksearch(stu, count, input_number);
1014.        if (pos != -1) {
1015.            // 将后面的元素依次向前移动
1016.            for (i = pos + 1; i < count; i++) {
1017.                stu[i - 1] = stu[i];
1018.            }
1019.            count--; // 更新学生信息数组的大小
1020.
1021.            cleardevice();
```

```
1022.           draw_entry(back_x, back_y, buttonarray);
1023.           setcolor(BLACK);
1024.           setfont(24, 0, "微软雅黑");
1025.           xyprintf(WINDOW_WIDTH / 2, 600, "删除成功！");
1026.           Sleep(300); // 延迟一段时间清屏重绘
1027.           cleardevice();
1028.           draw_entry(back_x, back_y, buttonarray);
1029.       }
1030.       else {
1031.           setcolor(BLACK);
1032.           setfont(24, 0, "微软雅黑");
1033.           xyprintf(WINDOW_WIDTH / 2, 600, "没有找到该学生！");
1034.           Sleep(300); // 延迟一段时间清屏重绘
1035.           cleardevice();
1036.           draw_entry(back_x, back_y, buttonarray);
1037.       }
1038.
1039.       return count;
1040.   }
1041.
1042.   //按照格式打印成绩
1043.   void printstudent(student stu[], char subname[MAX_EXAM][16])
1044.   {
1045.       int i,j;
1046.       printf("The Number of the student is %d\n", STU_NUM); // 打印学生数量
1047.
1048.       printf("%2s%11s%25s", "  ", "StudentID", "Name"); // 打印表头
1049.
1050.       for (i = 0; i < SUB_NUM; i++)
1051.       {
1052.           printf("%15s", subname[i]); // 打印科目名称
1053.       }
1054.
1055.       printf("%5s", "SUM"); // 打印总分列标题
1056.       printf("\n");
1057.
1058.       // 输出分数
1059.       for (i = 0; i < STU_NUM; i++)
1060.       {
1061.           printf("%2d", i + 1); // 输出序号
1062.           printf("%11s%25s", stu[i].number, stu[i].name); // 打印学生学号和姓名
1063.
1064.           for (j = 0; j < SUB_NUM; j++)
1065.           {
```

```
1066.             printf("%15d", stu[i].grade.subject[j]); // 打印学生各科目分数
1067.         }
1068.
1069.         printf("%5d", stu[i].grade.sum); // 打印学生总分
1070.         printf("\n");
1071.     }
1072. }
1073. //算总分
1074. void calculate_score(student stu[])
1075. {
1076.     //算总分
1077.     int i, j;
1078.     int sum = 0;
1079.     for (i = 0; i < STU_NUM; i++)
1080.     {
1081.         sum = 0;
1082.         for (j = 0; j < SUB_NUM; j++)
1083.         {
1084.             sum += stu[i].grade.subject[j];
1085.         }
1086.         stu[i].grade.sum = sum;
1087.     }
1088. }
1089. //将信息写入文件
1090. void write_file(student stu[])
1091. {
1092.     FILE* fp;
1093.     int i, j;
1094.     if ((fp = fopen("score.txt", "w")) == NULL)
1095.     {
1096.         printf("FAIL");
1097.     }
1098.     fprintf(fp, "%d\t%d\n", STU_NUM, SUB_NUM);
1099.     for (i = 0; i < STU_NUM; i++)
1100.     {
1101.         fprintf(fp, "%11s%25s", stu[i].number, stu[i].name);
1102.         for (j = 0; j < SUB_NUM; j++)
1103.         {
1104.             fprintf(fp, "%15d", stu[i].grade.subject[j]);
1105.         }
1106.         fprintf(fp, "%5d", stu[i].grade.sum);//总分
1107.         fprintf(fp, "\n");
1108.     }
1109.     for (i = 0; i < SUB_NUM; i++)
```

```
1110.     {
1111.         fprintf(fp, "%15s", subname[i]);
1112.     }
1113.     fclose(fp);
1114. }
1115. //将信息从文件读出
1116. void read_file(student stu[], int* STU_NUM, int* SUB_NUM)
1117. {
1118.     FILE* fp;
1119.     int i, j;
1120.     if ((fp = fopen("score.txt", "r")) == NULL)
1121.     {
1122.         puttext(180, 140, 28, "open fail");
1123.     }
1124.     fscanf(fp, "%d\t%d\n", STU_NUM, SUB_NUM);
1125.     for (i = 0; i < *STU_NUM; i++)
1126.     {
1127.         fscanf(fp, "%11s%25s", stu[i].number, stu[i].name);
1128.         for (j = 0; j < *SUB_NUM; j++)
1129.         {
1130.             fscanf(fp, "%15d", &stu[i].grade.subject[j]);
1131.         }
1132.         fscanf(fp, "%5d", &stu[i].grade.sum);//总分
1133.     }
1134.     for (i = 0; i < *SUB_NUM; i++)
1135.     {
1136.         fscanf(fp, "%15s", subname[i]);
1137.     }
1138.     fclose(fp);
1139. }
1140. // 交换两个学生结构体的函数
1141. void swap(student* a, student* b) {
1142.     student temp = *a;
1143.     *a = *b;
1144.     *b = temp;
1145. }
1146. // 快速排序的分区函数
1147. int partition(student arr[], int low, int high) {
1148.     int pivot = arr[high].grade.sum; // 选取最后一个元素作为枢纽元（总成绩）
1149.     int i = (low - 1);
1150.     int j;
1151.
1152.     for (j = low; j <= high - 1; j++) {
1153.         if (arr[j].grade.sum >= pivot) { // 如果当前元素的总成绩大于等于枢纽元，则将其交换到前半部分
```

```
1154.             i++;
1155.             swap(&arr[i], &arr[j]);
1156.         }
1157.     }
1158.     swap(&arr[i + 1], &arr[high]); // 将枢纽元放置到正确的位置
1159.     return (i + 1); // 返回枢纽元的索引
1160. }
1161. // 快速排序的主函数
1162. void quickSort(student arr[], int low, int high) {
1163.     if (low < high) {
1164.         int pi = partition(arr, low, high); // 获取枢纽元的索引
1165.
1166.         quickSort(arr, low, pi - 1);  // 对枢纽元的左侧进行递归排序
1167.         quickSort(arr, pi + 1, high); // 对枢纽元的右侧进行递归排序
1168.     }
1169. }
1170. //模糊搜索函数
1171. int vaguesearch(const char* key, const char* str)
1172. {
1173.     /*这段代码实现了模糊搜索的函数。它接受两个参数，一个是关键词 key，另一个是待搜索的字符串 str。
1174.         函数会在字符串 str 中寻找与关键词 key 匹配的子串。它通过逐个比较子串的方式进行搜索。
1175.         首先，它获取关键词和待搜索字符串的长度。
1176.         然后，它循环遍历待搜索字符串中的每个可能的子串，
1177.         逐个字符与关键词进行比较。如果找到了匹配的子串，
1178.         就返回 1 表示找到了，否则返回 0 表示未找到。*/
1179.     int i, j;
1180.     int keyLen = strlen(key); // 获取关键词的长度
1181.     int strLen = strlen(str); // 获取待搜索字符串的长度
1182.     // 依次比较子串
1183.     for (i = 0; i <= strLen - keyLen; i++) // 循环遍历待搜索字符串中的每个可能的子串
1184.     {
1185.         int match = 1; // 匹配标志，用于判断是否找到关键词
1186.
1187.         // 比较 str 和 key 的每个字符
1188.         for (j = 0; j < keyLen; j++) // 循环遍历关键词中的每个字符
1189.         {
1190.             if (str[i + j] != key[j]) // 比较对应位置的字符是否相等
1191.             {
1192.                 match = 0; // 如果有不匹配的字符，将匹配标志设为 0
1193.                 break; // 跳出内层循环
1194.             }
1195.         }
1196.
1197.         if (match == 1) // 如果匹配标志为 1，表示找到了关键词
```

```
1198.         {
1199.             return 1; // 找到了
1200.         }
1201.     }
1202.
1203.     return 0; // 未找到
1204. }
1205. //随机搜索函数
1206. int ranksearch(student stu[], int count, const char* input_number) {
1207.     // 使用随机搜索查找学号匹配的学生
1208.     int max_attempts = count * 5; // 最大尝试次数为学生数量的五倍
1209.     int attempts = 0; // 尝试次数
1210.     int pos = -1; // 找到的位置
1211.
1212.     while (attempts < max_attempts && pos == -1) {
1213.         int random_index = rand() % count; // 生成随机索引
1214.         if (strcmp(stu[random_index].number, input_number) == 0) {
1215.             pos = random_index;
1216.         }
1217.         attempts++;
1218.     }
1219.
1220.     return pos;
1221. }
1222. //短小精悍的主函数
1223. int main()
1224. {
1225.     mciSendString("open bgm.mp3 alias a ", 0, 0, 0);//来点音乐
1226.     mciSendString("play a", 0, 0, 0);
1227.     mainWindow();
1228.     return 0;
1229. }
```

分析总结、收获和体会:

优点：

1.使用了图形化界面库，界面直观

2.实现了鼠标交互和对话框输入，可以十分优雅地选择功能和输入数据

3.程序封装良好，对不同元素使用结构体进行封装，对绘制界面的操作使用函数封装，且参数规整，易于维护。

4.代码注释详尽，便于了解实现思路

新颖之处：

1.有背景音乐!!（Wildest Dreams -Taylor Swift）

2.实现了模糊查找，便于对学生信息进行管理

不足之处：

1. 功能较少，只能增删查改，没有额外添加设置及格线的操作
2. 无法直接将信息输出到界面中，且由于 EGE 库自身情况，可能会在运行中对计算机占用较大。

需要改进的地方：

1. 增加更多的功能，如设定优秀分数、及格分数，然后对不同分数段的同学进行统计
2. 直接将学生信息输出到图形界面上
3. 可以使用功能更加强大的图形界面库来布置更多组件，同时使界面更加优美。
4. 可以将不同函数封装到不同的程序中分而治之。

收获与学习体会：

1. 学会了如何调用图形界面化库来编写带有图形界面的程序，而不是仅仅停留在控制台程序层面上。
2. 现有知识已经可以实现一些简单程序的设计，难点是如何封装，本次程序设计经过大量重复优化，最终实现了由最初一次设定按钮信息，到最后通过结构体将单个按钮信息封装，再通过结构体数组将同一个界面的按钮封装在一起，最后只需要传入简单的参数就可以实现清屏重新绘制
3. 更加深刻地理解了递归、迭代、枚举等算法的思想，掌握了快速排序的操作方法，同时知道了在写递归时要避免进入死循环和出现爆栈的现象。
4. 学会了良好的进行注释，项目开发初期由于缺少良好的注释习惯，后期阅读代码很痛苦，后期通过添加详细的注释、优化变量名称使得不仅作者能够更好地维护，读者也能较快地理解代码的思路和变量的意义。
5. 学会了如何管理代码量较大的程序。
6. 学会了如何将程序封装成可执行文件，让其他人也能够在不配置环境的情况下使用。

自我评价：	是	否
程序运行是否无 bug?	√	
是否在撰写报告之前观看了 spoc 里的代码规范视频？	√	
程序代码是否符合代码规范(对齐与缩进，有必要的注释)?	√	
是否按模块化要求进行了程序设计，系统功能是否完善？	√	
是否是独立完成？	√	

自我评语：

编写一个上千行的代码、实现图形化操作真的不是一件容易的事情，他需要反复调试，反复优化才能实现。在这次程序设计中，我学会了如何更好地对函数进行封装，以及如何设计函数接口来尽可能的提高函数的复用性，同时知道了如何更好地选择数据结构来简化操作的同时提高计算机的执行效率。

提高了自己的编程习惯，养成了良好的注释代码的习惯，同时能够更加熟练的编写大程序。

学会了如何在陷入绝境的时候提振信心，同时能够寻找不同的灵感，调试这个图形化程序真的是花费了大量的时间，从最开始的判断按钮的鼠标点击信息，到最后的如何处理输入框提供的输入信息，每一步都十分坎坷，可以说 EGE 图形库的坑都踩了一遍，一个都没有少，但是最后也学到了很多知识，甚至在闲暇之余打开了 EGE 库的头文件看了看他是如何实现的，也初步了解了 C++语言 class 的作用，本质上创建图形库的过程中就是面向对象编程的思想。

综上，这是一个十分考验人耐力、思维的过程，这种代码量特别大的程序很锻炼人的编写和维护能力，在以后工程的建设中有很大帮助。

报告完成日期：2023/5/26