排序班级: A1611 姓名: 谷汶峻 学号: 34

**一. 题目信息:**

班级成绩统计与学籍管理系统

假设有1个班的原始成绩单形式如下:

学号 姓名 高等数学 大学英语 普通物理 电路 C程序设计

20160204961 小谷 91 74 82 61 70

20160205248 小李 92 85 89 73 80

......

功能要求:

1. 系统应对原始成绩单给予保留,原始数据的排序是无序的.
2. 将该班学生的成绩按学号顺序排序并存入相应数组,以方便打印.
3. 求出每位学生的平均分,并按平均分从高到低的顺序进行排序后,写入对应数组.
4. 统计出各门课的各分数段学生的人数,有序排列输出.
5. 为3门及3门以上不及格的学生打印退学通知单.通知单上要写明学生信息及其不及格科目的类别和成绩).
6. 删除已退学学生信息,将该班中升入高年级的学生信息写入对应数组中.

|------------------------------------------------------------|

| 请输入选项编号(0 - 3) : |

|------------------------------------------------------------|

| 1--------成绩录入 |

| 2--------成绩统计 |

| 3--------学籍处理 |

| 0--------退出 |

|------------------------------------------------------------|

图1 学籍管理系统主菜单

菜单要求:

1. 能够从主菜单(如图1所示)中重复选择选项.
2. 在主菜单选择2后,出现成绩统计子菜单(如图2所示),并在该菜单中重复选择选项(选择0时返回主菜单).
3. 在主菜单选择3后,出现学籍处理子菜单(如图3所示),并在该菜单中重复选择选项(选择0时返回到主菜单).

|-----------------------------------------------------------|

| 请输入选项编号(0 - 3) : |

|-----------------------------------------------------------|

| 1-----按学号排序 |

| 2-----计算平均值并排序 |

| 3-----统计分数段 |

| 0-----退出 |

|----------------------------------------------------------|

图2 成绩统计子菜单

|-----------------------------------------------------------|

| 请输入选项编号(0 - 3) : |

|-----------------------------------------------------------|

| 1-----生成并打印补考通知单 |

| 2-----生成并打印退学通知单 |

| 3-----生成并打印新名册 |

| 0-----退出 |

|-----------------------------------------------------------|

图3 学籍处理子菜单

1. **实现过程**
2. **数据说明[数据声明均在头文件stu.h中]:**
3. 定义学生结构体
4. typedef struct Student
5. {
6. int ID;
7. char Name[20];
8. float math;
9. float engl;
10. float phys;
11. float elec;
12. float CII;
13. float Average;
14. }student;

首先,定义学生结构体,包含一个学生的基本信息,学号ID为int型, 姓名Name为char型数组,其余各科成绩均为float型.

然后声明如下系统所需数组:

1. 声明学生数组及学生数量
2. student students[1000];
4. 挂科的学生的数组
5. student students\_bukao[1000];
6. 退学学生数组
7. student students\_fail[1000];
8. 成功升学学生数组
9. student next[1000];
10. **函数声明[函数声明均在头文件stu.h中]:**

/\*\*\*\*\*\*主菜单\*\*\*\*\*\*\*/

1. int mainMenu();
2. int mainMenu\_C();

对应系统的两个语言版本**(注:函数名以”\_C”结尾的函数,均是中文语言版本调用的,下文同样**),被最终的测试文件调用.

/\*\*\*\*\*\*\*\*\*\*子菜单\*\*\*\*\*\*\*\*\*\*/

1. //成绩录入
2. void GradeInput();
3. void GradeInput\_C();
5. //成绩查询
6. void GradeSearch();
7. void GradeSearch\_C();
9. //成绩统计
10. void GradeTongJi();
11. void GradeTongJi\_C();
13. //学籍处理
14. void Student\_status\_management();
15. void Student\_status\_management\_C();

/\*\*\*\*\*\*\*\*\*\*成绩录入模块\*\*\*\*\*\*\*\*\*\*/

1. /\*求平均值\*/
2. float Average(student stu);
3. float Average\_C(student stu);
5. /\*通过学号返回数组下标\*/
6. int Student\_SearchByID(int id);
7. int Student\_SearchByID\_C(int id);
9. /\*通过姓名返回数组下标\*/
10. int Student\_SearchByName(char name[]);
11. int Student\_SearchByName\_C(char name[]);
13. /\*显示单条学生记录\*/
14. void Student\_DisplaySingle(int);
15. void Student\_DisplaySingle\_C(int);
17. /\*插入学生信息\*/
18. void Student\_Insert();
19. void Student\_Insert\_C();
21. /\*修改学生信息\*/
22. void Student\_Modify();
23. void Student\_Modify\_C();
25. /\*删除学生信息\*/
26. void Student\_Delete();
27. void Student\_Delete\_C();
29. /\*按学号查询\*/
30. void Student\_SelectByID();
31. void Student\_SelectByID\_C();
33. /\*按姓名查询\*/
34. void Student\_SelectByName();
35. void Student\_SelectByName\_C();

/\*\*\*\*\*\*\*\*\*\*成绩统计模块\*\*\*\*\*\*\*\*\*\*/

1. /\*按平均值排序\*/
2. void Student\_SortByAverage();
3. void Student\_SortByAverage\_C();
5. /\*按学号排序\*/
6. void Student\_SortByID();
7. void Student\_SortByID\_C();
9. //各门课各分数段学生人数统计并打印
10. void Student\_Sort\_EachSubject();
11. void Student\_Sort\_EachSubject\_C();
13. /\*显示学生信息\*/
14. void Student\_Display();
15. void Student\_Display\_C();

/\*\*\*\*\*\*\*\*\*\*学籍处理模块\*\*\*\*\*\*\*\*\*\*/

1. //补考通知
2. void print\_information\_bukao();
3. void print\_information\_bukao\_C();
5. //退学通知
6. void print\_information\_tuixue();
7. void print\_information\_tuixue\_C();
9. //升学学生
10. void print\_information\_up();
11. void print\_information\_up\_C();

/\*\*\*\*\*\*\*\*\*\*文件读写模块\*\*\*\*\*\*\*\*\*\*/

1. /\*将学生信息从文件读出\*/
2. void IO\_ReadInfo();
4. /\*将学生信息写入文件\*/
5. void IO\_WriteInfo();

**3.分模块(函数具体实现)及各模块测试:**

1. **菜单显示模块**(菜单模块大多是利用printf函数和while语句来实现菜单打印及重复选择菜单功能的,此处仅以中文版本的主菜单为例作简单介绍.)[代码位置StudentManagementSystem/src/mainMenu.c].
2. //中文菜单
3. int mainMenu\_C()
4. {
5. int choice;
6. do{
7. //system("sl");
8. //system("sh do.sh");
9. system("clear");
10. int a = 0;
11. a++;
12. if(a == 1){
13. printf(YELLOW\_COLOR,"----------------------欢迎使用学生成绩与学籍管理系统-----------------------\n");
14. }
15. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
16. printf(GREEN\_COLOR,"                |                  主菜单                     |\n");
17. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
18. printf(GREEN\_COLOR,"                |      1 ---- 成绩录入与删改                  |\n");
19. printf(GREEN\_COLOR,"                |      2 ---- 成绩查询                        |\n");
20. printf(GREEN\_COLOR,"                |      3 ---- 成绩统计                        |\n");
21. printf(GREEN\_COLOR,"                |      4 ---- 学籍管理                        |\n");
22. printf(GREEN\_COLOR,"                |      0 ---- 退出                            |\n");
23. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
24. printf(GREEN\_COLOR,"                请选择(1 ~ 3, 0: exit):\n");
25. printf(YELLOW\_COLOR,">>>>>>>>>>>>>>>:");
26. scanf("%d", &choice);
27. switch(choice)
28. {
29. case 1: GradeInput\_C(); break;
30. case 2: GradeSearch\_C(); break;
31. case 3: GradeTongJi\_C(); break;
32. case 4: Student\_status\_management\_C(); break;
33. case 0: printf(YELLOW\_COLOR,"-----------------------感谢您的使用!---------------------------------------\n");
34. exit(0);
35. default: printf(RED\_COLOR,"您输入的非法字符,请按任意键以再次选择.\n");
36. getchar();
37. getchar();
38. }
39. }while(1);
41. return 0;
43. }

函数细节简介:

int型变量choice结合switch语句实现各个子菜单(或称各个功能)的调用.

system函数调用终端命令clear实现每次切换回主菜单的清屏效果.

int型变量a及其自增实现只有在第一次进入主菜单时才有欢迎语句的打印,避免来回切换主菜单和子菜单时,多次打印欢迎语句.

功能测试如图1所示:

b**.成绩录入与修改模块:**

[代码位置StudentManagementSystem/src/stu\_C.c]

成绩录入设定只有有管理员密码的人才可以进行添加修改和删除成绩操作,此细节借用linux终端的echoing功能的开与关来实现隐式密码输入.

功能测试如图2所示:

成功进入该子菜单后,根据选项编号进入相应功能.

**选择选项1:** 添加学生成绩.依次输入学生个人信息与各科成绩.如:2016 gu 66 66 66 66 66, 输入完成后可选择继续输入与否.

功能测试如图3所示:

该功能是利用一个学生结构体数组来存储各个学生的信息的.

当添加学号有重复时,会提示拥有该学号的学生已存在,中断本次成绩添加.

功能测试如图4所示:

该功能是将新输入的学号与学生数组中的各个学生的学号进行遍历比较,若没有与之相等项,则可继续添加此学生的其他信息,否则弹出错误.

**选择选项2:** 修改学生成绩.

输入想要修改的学生对应的学号,查询到该生信息存在后,将该生信息打印出来,并可以进行信息的针对性修改和全部修改.

功能测试如图5所示:

该功能主要借助一个查询函数 打印单个学生信息的函数及if判断语句来实现的.首先查询函数接受用户输入的学号,依据此数据在学生数组中寻找与之对应的元素,找到后将元素下标返回,打印函数接收此下标并将下标对应元素的各项信息全部打印出来.if语句根据输入的修改选择的编号进行判断,来调用相应修改语句.

**选择选项3:** 删除学生成绩.

输入想要删除的学生对应的学号,查询到该生信息存在后,将该生信息打印出来,选择是否确认要删除.

功能测试如图6所示:

类似修改功能的原理,该功能也是借助一个查询函数 打印单个学生的函数及if判断语句来实现的.首先查询函数接受用户输入的学号,依据此数据在学生数组中寻找与之对应的元素,找到后将元素下标返回,打印函数接收此下标并将下标对应元素的各项信息全部打印出来.if语句根据输入的选项,来确定是否要删除该学生的信息.

**c.成绩查询模块**

[代码位置StudentManagementSystem/src/stu\_C.c]

该子菜单的两个功能原理类似,均是借助查询函数 打印单个学生信息的函数.

功能测试如图7 8所示:

这两项功能主要借助一个查询函数 打印单个学生信息的函数来实现的.首先查询函数接受用户输入的学号,依据此数据在学生数组中寻找与之对应的元素,找到后将元素下标返回,打印函数接收此下标并将下标对应元素的各项信息全部打印出来.

**d.成绩统计模块**

[代码位置StudentManagementSystem/src/stu\_C.c]

**选择选项1: 按学号排序.**

功能测试如图9所示:

此功能主要是借助冒泡排序法,排序依据是每个学生元素的”学号”数据值.

**选择选项2: 按平均分排序.**

功能测试如图10所示:

此功能主要是借助冒泡排序法,排序依据是每个学生元素的”平均分”数据值.

**选择选项3:各科成绩分数段统计.**

功能测试如图11所示:

此功能主要是借助for循环和if else判断语句. for循环遍历每个学生,if else中判断各科分数值在某个区域中时,该区域对应设定的变量值自增一次(该int型变量初始赋值为0).

**e.学籍管理模块**

[代码位置StudentManagementSystem/src/stu\_C.c]

**选择选项1: 统计并打印出补考学生信息**

功能测试如图12所示:

此功能主要是借助for循环和if else语句.该功能内部实现设定一个int型数据存储各个学生挂科数,当其>=0时,将此学生存储到设定的需要补考的学生数组中,然后再遍历补考学生数组,将其分数小于60的科目及成绩输出.

**选择选项2: 统计并打印出退学学生信息**

功能测试如图13所示:

此功能主要是借助for循环和if else语句.该功能内部实现设定一个int型数据存储各个学生挂科数,当其>=3时,将此学生存储到设定的被退学的学生数组中,然后再遍历被退学学生数组,将其分数小于60的科目及成绩输出.

**选择选项3: 统计并打印出升学学生名单**

功能测试如图14所示:

此功能主要是借助for循环和if else语句.该功能内部实现设定一个int型数据存储各个学生挂科数,当其<=2时,将此学生存储到设定的需要补考的学生数组中,然后再遍历补考学生数组,将其学号及姓名有序输出.

**4.错误调试**

**5.完整性测试**

(1)系统已经实现的功能:

系统对原始成绩单保留.

主菜单与子菜单的成功切换.

学生成绩录入与删改.

学生成绩查询.

成绩统计.

学籍管理.

退出.

1. 自己创新功能:

中英双语版本系统.

密码进入学生成绩添加与删改模块.

修改学生成绩时可针对性修改和全部修改.

1. 修改功能展示:

借鉴同学们的示例代码,将printf打印彩色字体的用句宏定义包在头文件stu.h中:

**stu.h:**

1. #ifndef stu\_h
2. #define stu\_h
3. ......
4. #define RED\_COLOR        "\e[31m\e[1m%s\e[0m"
5. #define GREEN\_COLOR      "\e[32m\e[1m%s\e[0m"
6. #define YELLOW\_COLOR     "\e[33m\e[1m%s\e[0m"
7. ......

1. 不足与展望:

添加学生信息时的深度筛选(学号 成绩不能为字母和特殊符号等).

或许可以考虑比数组更合适的数据结构来实现系统.

选择更合适的文件存储格式.

每个学生可有多次考试成绩的存储.

老师可能需要了解单个科目的成绩状况.

或许中英双版本可以做的更好些,而不是大量的代码复用.

附录:

(下列代码您均可直接从我的github仓库中获取,url为: <https://github.com/Guwenjun2017/StudentManagementSystem>

仓库中的README.md文件中有较详细的使用说明.欢迎您提出建议).

**stu.h:**

1. #ifndef stu\_h
2. #define stu\_h
3. #include <stdio.h>
4. #include <stdlib.h>
5. #include <string.h>
6. #define RED\_COLOR        "\e[31m\e[1m%s\e[0m"
7. #define GREEN\_COLOR      "\e[32m\e[1m%s\e[0m"
8. #define YELLOW\_COLOR     "\e[33m\e[1m%s\e[0m"
10. void IO\_ReadInfo();
11. void Student\_Display();
13. /\*定义学生结构体\*/
14. typedef struct Student
15. {
16. int ID;
17. char Name[20];
18. float math;
19. float engl;
20. float phys;
21. float elec;
22. float CII;
23. float Average;
24. }student;
26. /\*声明学生数组及学生数量\*/
27. student students[1000];
29. //有挂科的学生的数组
30. student students\_bukao[1000];
31. //退学学生数组
32. student students\_fail[1000];
33. //升学学生数组
34. student next[1000];
36. /\*\*\*\*\*\*\*\*\*\*成绩录入模块\*\*\*\*\*\*\*\*\*\*/
37. /\*求平均值\*/
38. float Average(student stu);
39. float Average\_C(student stu);
41. /\*通过学号返回数组下标\*/
42. int Student\_SearchByID(int id);
43. int Student\_SearchByID\_C(int id);
45. /\*通过姓名返回数组下标\*/
46. int Student\_SearchByName(char name[]);
47. int Student\_SearchByName\_C(char name[]);
49. /\*显示单条学生记录\*/
50. void Student\_DisplaySingle(int);
51. void Student\_DisplaySingle\_C(int);
53. /\*插入学生信息\*/
54. void Student\_Insert();
55. void Student\_Insert\_C();
57. /\*修改学生信息\*/
58. void Student\_Modify();
59. void Student\_Modify\_C();
61. /\*删除学生信息\*/
62. void Student\_Delete();
63. void Student\_Delete\_C();
65. /\*按学号查询\*/
66. void Student\_SelectByID();
67. void Student\_SelectByID\_C();
69. /\*按姓名查询\*/
70. void Student\_SelectByName();
71. void Student\_SelectByName\_C();
73. /\*\*\*\*\*\*\*\*\*\*成绩统计模块\*\*\*\*\*\*\*\*\*\*/
74. /\*按平均值排序\*/
75. void Student\_SortByAverage();
76. void Student\_SortByAverage\_C();
78. /\*按学号排序\*/
79. void Student\_SortByID();
80. void Student\_SortByID\_C();
82. //各门课各分数段学生人数统计并打印
83. void Student\_Sort\_EachSubject();
84. void Student\_Sort\_EachSubject\_C();
86. /\*显示学生信息\*/
87. void Student\_Display();
88. void Student\_Display\_C();
90. /\*\*\*\*\*\*\*\*\*\*学籍处理模块\*\*\*\*\*\*\*\*\*\*/
91. //补考通知
92. void print\_information\_bukao();
93. void print\_information\_bukao\_C();
95. //退学通知
96. void print\_information\_tuixue();
97. void print\_information\_tuixue\_C();
99. //升学学生
100. void print\_information\_up();
101. void print\_information\_up\_C();
103. /\*\*\*\*\*\*\*\*\*\*文件读写模块\*\*\*\*\*\*\*\*\*\*/
104. /\*将学生信息从文件读出\*/
105. void IO\_ReadInfo();
107. /\*将学生信息写入文件\*/
108. void IO\_WriteInfo();
110. /\*\*\*\*\*\*\*\*\*\*子菜单\*\*\*\*\*\*\*\*\*\*/
111. //成绩录入
112. void GradeInput();
113. void GradeInput\_C();
115. //成绩查询
116. void GradeSearch();
117. void GradeSearch\_C();
119. //成绩统计
120. void GradeTongJi();
121. void GradeTongJi\_C();
123. //学籍处理
124. void Student\_status\_management();
125. void Student\_status\_management\_C();
127. /\*\*\*\*\*\*\*\*\*\*主菜单\*\*\*\*\*\*\*\*\*\*/
128. int mainMenu();
129. int mainMenu\_C();
130. #endif

**test.c:**

1. #include "../inc/stu.h"
3. int main(int argc, char \*\*argv)
4. {
5. char language;
6. printf("Please choose your language, English please input E/e, Chinese please input C/c, After input, press the Enter key.\n");
7. printf("[请选择您的地区语言,英文请输入E,中文请输入C,输入后Enter键结束].\n");
8. scanf("%c", &language);
9. if( language == 'E' || language == 'e')
10. mainMenu();
11. if( language == 'C' || language == 'c')
12. mainMenu\_C();
14. return 0;
15. }

**mainMenu.c:**

1. #include "../inc/stu.h"
3. /\*\*\*\*\*\*\*\*\*\*主模块\*\*\*\*\*\*\*\*\*\*/
4. //English menu
5. int mainMenu()
6. {
7. int choice;
8. do{
9. //system("sl");
10. //system("sh do.sh");
11. system("clear");
12. int a = 0;
13. a++;
14. if(a == 1){
15. printf(YELLOW\_COLOR,"----------------------WELCOME TO StudentManagementSystem-------------------\n");
16. }
17. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
18. printf(GREEN\_COLOR,"                |                 main                        |\n");
19. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
20. printf(GREEN\_COLOR,"                |      1 ---- Result input                    |\n");
21. printf(GREEN\_COLOR,"                |      2 ---- Score query                     |\n");
22. printf(GREEN\_COLOR,"                |      3 ---- Achievement statistics          |\n");
23. printf(GREEN\_COLOR,"                |      4 ---- Student status management       |\n");
24. printf(GREEN\_COLOR,"                |      0 ---- Exit                            |\n");
25. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
26. printf(GREEN\_COLOR,"                Please choose(1 ~ 3, 0: exit):\n");
27. printf(YELLOW\_COLOR,">>>>>>>>>>>>>>>:");
28. scanf("%d", &choice);
29. switch(choice)
30. {
31. case 1: GradeInput(); break;
32. case 2: GradeSearch(); break;
33. case 3: GradeTongJi(); break;
34. case 4: Student\_status\_management(); break;
35. case 0: printf(YELLOW\_COLOR,"-----------------------THANK YOU FOR YOUR USING!---------------------------\n");
36. exit(0);
37. default: printf(RED\_COLOR,"It is an illegal option, and press any key to choose again!\n");
38. getchar();
39. getchar();
40. }
41. }while(1);
43. return 0;
45. }
47. //中文菜单
48. int mainMenu\_C()
49. {
50. int choice;
51. do{
52. //system("sl");
53. //system("sh do.sh");
54. system("clear");
55. int a = 0;
56. a++;
57. if(a == 1){
58. printf(YELLOW\_COLOR,"----------------------欢迎使用学生成绩与学籍管理系统-----------------------\n");
59. }
60. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
61. printf(GREEN\_COLOR,"                |                  主菜单                     |\n");
62. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
63. printf(GREEN\_COLOR,"                |      1 ---- 成绩录入与删改                  |\n");
64. printf(GREEN\_COLOR,"                |      2 ---- 成绩查询                        |\n");
65. printf(GREEN\_COLOR,"                |      3 ---- 成绩统计                        |\n");
66. printf(GREEN\_COLOR,"                |      4 ---- 学籍管理                        |\n");
67. printf(GREEN\_COLOR,"                |      0 ---- 退出                            |\n");
68. printf(GREEN\_COLOR,"                |---------------------------------------------|\n");
69. printf(GREEN\_COLOR,"                请选择(1 ~ 3, 0: exit):\n");
70. printf(YELLOW\_COLOR,">>>>>>>>>>>>>>>:");
71. scanf("%d", &choice);
72. switch(choice)
73. {
74. case 1: GradeInput\_C(); break;
75. case 2: GradeSearch\_C(); break;
76. case 3: GradeTongJi\_C(); break;
77. case 4: Student\_status\_management\_C(); break;
78. case 0: printf(YELLOW\_COLOR,"-----------------------感谢您的使用!---------------------------------------\n");
79. exit(0);
80. default: printf(RED\_COLOR,"您输入的非法字符,请按任意键以再次选择.\n");
81. getchar();
82. getchar();
83. }
84. }while(1);
86. return 0;
88. }

**son\_of\_main.c:**

1. #include "../inc/stu.h"
3. //英文子菜单
4. /\*\*\*\*\*\*\*\*\*\*子菜单模块\*\*\*\*\*\*\*\*\*\*/
5. //成绩录入模块
6. void GradeInput()
7. {
8. //IO\_ReadInfo();
9. int choice;
11. while(1)
12. {
13. system("clear");
14. printf(GREEN\_COLOR,"\n------ Student store manage--------------\n");
15. printf(GREEN\_COLOR,"-------1. Add student Grade----------------\n");
16. printf(GREEN\_COLOR,"-------2. Modify student Grade-------------\n");
17. printf(GREEN\_COLOR,"-------3. Delete student Grade-------------\n");
18. printf(GREEN\_COLOR,"-------0. Return to main-------------------\n");
19. printf(GREEN\_COLOR,"-------Please choose(1 ~ 4), 0 to main:\n");
20. printf(">>>>>>>:");
21. scanf("%d",&choice);
22. getchar();
23. switch(choice){
24. case 1: Student\_Insert(); break;
25. case 2: Student\_Modify(); break;
26. case 3: Student\_Delete(); break;
27. case 0: return;
28. default: printf(GREEN\_COLOR,"It is an illegal option, and press Enter to choose again!\n");
29. getchar();
30. }
31. }
33. //IO\_WriteInfo();
34. return;
35. }
37. //成绩查询
38. void GradeSearch(){
39. int choice;
41. while(1){
42. system("clear");
43. printf(GREEN\_COLOR,"\n------ Student Score Query--------------\n");
44. printf(GREEN\_COLOR,"-------1. Searh\_by\_ID--------------------\n");
45. printf(GREEN\_COLOR,"-------2. Searh\_by\_name--------------------\n");
46. printf(GREEN\_COLOR,"-------0. Return to main-------------------\n");
47. printf(">>>>>>>:");
48. scanf("%d", &choice);
49. getchar();
50. switch(choice){
51. case 1: Student\_SelectByID(); break;
52. case 2: Student\_SelectByName(); break;
53. case 0: return;
54. default: printf(GREEN\_COLOR,"It is an illegal option, and press Enter to choose again!\n");
55. getchar();
56. }
57. }
58. return;
59. }
60. //成绩统计模块
61. void GradeTongJi(){
62. //IO\_ReadInfo();
63. int choice;
64. system("clear");
66. while(1){
67. printf(GREEN\_COLOR,"\n|---------Achievement\_Statistics-------------|\n");
68. printf(GREEN\_COLOR,"|      1 ---- Sort\_by\_ID                     |\n");
69. printf(GREEN\_COLOR,"|      2 ---- Sort\_by\_Average                |\n");
70. printf(GREEN\_COLOR,"|      3 ---- Statistical\_franction\_segment  |\n");
71. printf(GREEN\_COLOR,"|      0 ---- return to main                 |\n");
72. printf(GREEN\_COLOR,"|--------------------------------------------|\n");
73. printf(GREEN\_COLOR,"Please choose(1 ~ 3, 0 to main):\n");
74. printf(">>>>>>>:");
75. scanf("%d", &choice);
76. getchar();
77. switch(choice){
78. case 1: Student\_SortByID(); break;
79. case 2: Student\_SortByAverage(); break;
80. case 3: Student\_Sort\_EachSubject(); break;
81. case 0: return; break;
82. default: printf(GREEN\_COLOR,"It is an illegal option, and press Enter to choose again!\n");
83. getchar();
84. }
85. }
87. return;
88. }
90. //学籍处理
91. void Student\_status\_management(){
92. //IO\_ReadInfo();
93. int choice;
94. system("clear");
95. do{
96. printf(GREEN\_COLOR,"|---------------------------------------------------|\n");
97. printf(GREEN\_COLOR,"|      Student\_status\_management                    |\n");
98. printf(GREEN\_COLOR,"|---------------------------------------------------|\n");
99. printf(GREEN\_COLOR,"|      1 --- Generate and print notice of make-up   |\n");
100. printf(GREEN\_COLOR,"|      2 --- Generate and print withdrawal notice   |\n");
101. printf(GREEN\_COLOR,"|      3 --- Generate and print entry lists         |\n");
102. printf(GREEN\_COLOR,"|      0 --- return to main                         |\n");
103. printf(GREEN\_COLOR,"|---------------------------------------------------|\n");
104. printf(GREEN\_COLOR,"Please choose(1 ~ 3, 0 to main):\n");
105. printf(YELLOW\_COLOR,">>>>>>>:");
106. scanf("%d", &choice);
107. switch(choice)
108. {
109. case 1: print\_information\_bukao(); break;
110. case 2: print\_information\_tuixue(); break;
111. case 3: print\_information\_up(); break;
112. case 0: return; break;
113. default: printf(GREEN\_COLOR,"It is an illegal option, and press Enter to choose again!\n");
114. getchar();
115. }
116. }while(1);
118. return;
119. }

122. //中文子菜单
123. //成绩录入模块
124. void GradeInput\_C()
125. {
126. //IO\_ReadInfo();
127. int choice;
129. while(1)
130. {
131. system("clear");
132. printf(GREEN\_COLOR,"\n-----------学生成绩管理--------------\n");
133. printf(GREEN\_COLOR,"-------1. 添加学生成绩\n");
134. printf(GREEN\_COLOR,"-------2. 修改学生成绩\n");
135. printf(GREEN\_COLOR,"-------3. 删除学生成绩\n");
136. printf(GREEN\_COLOR,"-------0. 返回主菜单\n");
137. printf(GREEN\_COLOR,"-------请选择(1 ~ 4), 0 返回主菜单:\n");
138. printf(">>>>>>>:");
139. scanf("%d",&choice);
140. getchar();
141. switch(choice){
142. case 1: Student\_Insert\_C(); break;
143. case 2: Student\_Modify\_C(); break;
144. case 3: Student\_Delete\_C(); break;
145. case 0: return;
146. default: printf(GREEN\_COLOR,"您输入的是非法字符,请按任意键以重新输入.\n");
147. getchar();
148. }
149. }
151. //IO\_WriteInfo();
152. return;
153. }
155. //成绩查询
156. void GradeSearch\_C(){
157. int choice;
159. while(1){
160. system("clear");
161. printf(GREEN\_COLOR,"\n-------- 学生成绩查询 ---------------\n");
162. printf(GREEN\_COLOR,"-------1. 按学号查询-----------------\n");
163. printf(GREEN\_COLOR,"-------2. 按姓名查询-----------------\n");
164. printf(GREEN\_COLOR,"-------0. 返回主菜单-----------------\n");
165. printf(">>>>>>>:");
166. scanf("%d", &choice);
167. getchar();
168. switch(choice){
169. case 1: Student\_SelectByID\_C(); break;
170. case 2: Student\_SelectByName\_C(); break;
171. case 0: return;
172. default: printf(GREEN\_COLOR,"您输入的是非法字符,请按任意键以重新输入.\n");
173. getchar();
174. }
175. }
176. return;
177. }
178. //成绩统计模块
179. void GradeTongJi\_C(){
180. //IO\_ReadInfo();
181. int choice;
182. system("clear");
184. while(1){
185. printf(GREEN\_COLOR,"\n|----------------成绩统计--------------------|\n");
186. printf(GREEN\_COLOR,"|      1 ---- 按学号排序                     |\n");
187. printf(GREEN\_COLOR,"|      2 ---- 按平均分排序                   |\n");
188. printf(GREEN\_COLOR,"|      3 ---- 各科成绩段统计                 |\n");
189. printf(GREEN\_COLOR,"|      0 ---- 返回主菜单                     |\n");
190. printf(GREEN\_COLOR,"|--------------------------------------------|\n");
191. printf(GREEN\_COLOR,"请选择(1 ~ 3, 0 返回主菜单):\n");
192. printf(">>>>>>>:");
193. scanf("%d", &choice);
194. getchar();
195. switch(choice){
196. case 1: Student\_SortByID\_C(); break;
197. case 2: Student\_SortByAverage\_C(); break;
198. case 3: Student\_Sort\_EachSubject\_C(); break;
199. case 0: return; break;
200. default: printf(GREEN\_COLOR,"您输入的是非法字符,请按任意键以重新输入.\n");
201. getchar();
202. }
203. }
205. return;
206. }
208. //学籍处理
209. void Student\_status\_management\_C(){
210. //IO\_ReadInfo();
211. int choice;
212. system("clear");
213. do{
214. printf(GREEN\_COLOR,"|---------------------------------------------------|\n");
215. printf(GREEN\_COLOR,"|                   学生学籍管理                    |\n");
216. printf(GREEN\_COLOR,"|---------------------------------------------------|\n");
217. printf(GREEN\_COLOR,"|      1 --- 统计并打印出补考学生信息               |\n");
218. printf(GREEN\_COLOR,"|      2 --- 统计并打印出退出学生信息               |\n");
219. printf(GREEN\_COLOR,"|      3 --- 统计并打印出升学学生名单               |\n");
220. printf(GREEN\_COLOR,"|      0 --- 返回主菜单                             |\n");
221. printf(GREEN\_COLOR,"|---------------------------------------------------|\n");
222. printf(GREEN\_COLOR,"请选择(1 ~ 3, 0 返回主菜单):\n");
223. printf(YELLOW\_COLOR,">>>>>>>:");
224. scanf("%d", &choice);
225. switch(choice)
226. {
227. case 1: print\_information\_bukao\_C(); break;
228. case 2: print\_information\_tuixue\_C(); break;
229. case 3: print\_information\_up\_C(); break;
230. case 0: return; break;
231. default: printf(GREEN\_COLOR,"您输入的是非法字符,请按任意键以重新输入.\n");
232. getchar();
233. }
234. }while(1);
236. return;
237. }

**stu.c:**

1. #include "../inc/stu.h"
3. //初始学生数为0
4. int num\_of\_students = 0;
6. /\*求平均值\*/
7. float Average(student stu){
8. return (stu.math + stu.engl + stu.phys + stu.elec + stu.CII) / 5;
9. }
11. /\*通过学号返回数组下标\*/
12. int Student\_SearchByID(int id){
13. int i;
14. for(i=0;i<num\_of\_students;i++){
15. if (students[i].ID == id)
16. {
17. return i;
18. }
19. }
20. return -1;
21. }
23. /\*通过姓名返回数组下标\*/
24. int Student\_SearchByName(char name[]){
25. int i;
26. for(i=0;i<num\_of\_students;i++){
27. if (strcmp(students[i].Name, name) == 0){
28. return i;
29. }
30. }
31. return -1;
32. }
34. /\*显示单条学生记录\*/
35. void Student\_DisplaySingle(int index){
36. printf("%5s %9s %9s %9s %9s %9s %9s \t%9s  \n","ID","Name","math","engl","phys","elec","CII","平均成绩");
37. printf("------------------------------------------------------------------------------------------\n");
38. printf("%7d %7s %10.2f %9.2f %9.2f %9.2f %10.2f %10.2f\n",students[index].ID,students[index].Name,
39. students[index].math, students[index].engl, students[index].phys, students[index].elec, \
40. students[index].CII, students[index].Average);
42. return;
43. }
45. /\*插入学生信息\*/
46. void Student\_Insert(){
47. //IO\_ReadInfo();
48. int  a = 0;
49. while(1){
50. system("clear");
51. printf("Input\_ID:");
52. scanf("%d",&students[num\_of\_students].ID);
53. getchar();
54. for(int i = 0; i < num\_of\_students - 1; i++){
55. if(students[num\_of\_students].ID == students[i].ID){
56. printf("\e[31m\e[1m%s\e[0m","add error!This student is existed.");
57. a = 1;
58. }
59. }
60. if(a == 1){
61. break;
62. }
64. printf("Input\_Name:");
65. scanf("%s",&students[num\_of\_students].Name);
66. getchar();
68. printf("Input\_math:");
69. do{
70. scanf("%f",&students[num\_of\_students].math);
71. if(students[num\_of\_students].math >= 0 && students[num\_of\_students].math <= 100){
72. break;
73. }else
74. printf(RED\_COLOR,"Grade not in the limit!please input again.\n");
75. printf(">>>>>>:");
77. }while(1);
78. getchar();
80. printf("Input\_engl:");
81. do{
82. scanf("%f",&students[num\_of\_students].engl);
83. if(students[num\_of\_students].engl >= 0 && students[num\_of\_students].engl <= 100){
84. break;
85. }else
86. printf(RED\_COLOR,"Grade not in the limit!please input again.\n");
87. printf(">>>>>>:");
89. }while(1);
90. getchar();
92. printf("Input\_phys:");
93. do{
94. scanf("%f",&students[num\_of\_students].phys);
95. if(students[num\_of\_students].phys >= 0 && students[num\_of\_students].phys <= 100){
96. break;
97. }else
98. printf(RED\_COLOR,"Grade not in the limit!please input again.\n");
99. printf(">>>>>>:");
101. }while(1);
102. getchar();
104. printf("Input\_elec:");
105. do{
106. scanf("%f",&students[num\_of\_students].elec);
107. if(students[num\_of\_students].elec >= 0 && students[num\_of\_students].elec <= 100){
108. break;
109. }else
110. printf(RED\_COLOR,"Grade not in the limit!please input again.\n");
111. printf(">>>>>>:");
113. }while(1);
114. getchar();
116. printf("Input\_CII:");
117. do{
118. scanf("%f",&students[num\_of\_students].CII);
119. if(students[num\_of\_students].CII >= 0 && students[num\_of\_students].CII <= 100){
120. break;
121. }else
122. printf(RED\_COLOR,"Grade not in the limit!please input again.\n");
123. printf(">>>>>>:");
125. }while(1);
126. getchar();
128. students[num\_of\_students].Average=Average(students[num\_of\_students]);
129. num\_of\_students++;
131. printf("\e[31m\e[1m%s\e[0m","Continue?(y/n)");
133. if (getchar()=='n'){
134. break;
135. }
136. }
138. //IO\_WriteInfo();
139. return;
140. }
142. /\*修改学生信息\*/
143. void Student\_Modify(){
144. while(1){
145. int id;
146. int index;
148. printf("Please input the student's ID, the one you want to modify:");
149. scanf("%d",&id);
150. getchar();
151. index=Student\_SearchByID(id);
152. if(index==-1){
153. printf(GREEN\_COLOR,"Not exist!\n");
154. }
155. else{
156. printf("The infomation of the student you want to modify is:\n");
157. Student\_DisplaySingle(index);
158. int choice;
159. printf("please choose the information you want to modify:\n");
160. printf("-1.math\n");
161. printf("-2.engl\n");
162. printf("-3.phys\n");
163. printf("-4.elec\n");
164. printf("-5.CII\n");
165. printf("-6.all information\n");
166. printf("-0.break\n");
167. printf(">>>>:");
168. scanf("%d", &choice);
169. if(1 == choice){
170. printf("please input the new math score:");
171. scanf("%f",&students[index].math);
172. students[index].Average=Average(students[index]);
173. }
174. if(2 == choice){
175. printf("please input the new english score:");
176. scanf("%f",&students[index].engl);
177. students[index].Average=Average(students[index]);
178. }
179. if(3 == choice){
180. printf("please input the new physics score:");
181. scanf("%f",&students[index].phys);
182. students[index].Average=Average(students[index]);
183. }
184. if(4 == choice){
185. printf("please input the new electric score:");
186. scanf("%f",&students[index].elec);
187. students[index].Average=Average(students[index]);
188. }
189. if(5 == choice){
190. printf("please input the new CII score:");
191. scanf("%f",&students[index].CII);
192. students[index].Average=Average(students[index]);
193. }
194. if(6 == choice){
195. printf("-- please input the all new information--\n");
196. printf("Input\_ID:");
197. scanf("%d",&students[index].ID);
198. getchar();
200. printf("Input\_Name:");
201. scanf("%s",&students[index].Name);
202. getchar();
204. printf("Input\_math:");
205. scanf("%f",&students[index].math);
206. getchar();
208. printf("Input\_engl:");
209. scanf("%f",&students[index].engl);
210. getchar();
212. printf("Input\_phys:");
213. scanf("%f",&students[index].phys);
214. getchar();
216. printf("Input\_elec:");
217. scanf("%f",&students[index].elec);
218. getchar();
220. printf("Input\_CII:");
221. scanf("%f",&students[index].CII);
222. getchar();
224. students[index].Average=Average(students[index]);
225. }
226. }
227. getchar();
228. printf(GREEN\_COLOR,"Continue?(y/n)");
229. if(getchar()=='n'){
230. break;
231. }
232. }
234. return;
235. }
237. /\*删除学生信息\*/
238. void Student\_Delete(){
239. int i;
240. while(1){
241. int id;
242. int index;
243. printf("Please input ID of student which you want to delete:");
244. scanf("%d",&id);
245. getchar();
246. index=Student\_SearchByID(id);
248. if(index==-1){
249. printf("Not exist!\n");
250. }
251. else{
252. printf("The infomation of the student you want to delete is:\n");
253. Student\_DisplaySingle(index);
254. printf(GREEN\_COLOR,"Sure delete?(y/n)");
255. if(getchar()=='y'){
256. for(i=index;i<num\_of\_students-1;i++){
257. //把后边的对象都向前移动
258. students[i]=students[i+1];
259. }
260. num\_of\_students--;
261. }
262. getchar();
263. }
265. printf(GREEN\_COLOR,"Continue?(Enter/n)");
266. if(getchar()=='n'){
267. break;
268. }
269. }
271. //IO\_WriteInfo();
272. return;
273. }
275. /\*按学号查询\*/
276. void Student\_SelectByID(){
277. while(1){
278. int a;
279. int index;
280. printf("Please input the student's ID, the one you want to search:");
281. scanf("%d",&a);
282. getchar();
283. index=Student\_SearchByID(a);
285. if (index==-1){
286. printf(GREEN\_COLOR,"Not exist!\n");
287. }else{
288. printf("The infomation of the student you want to search is:\n");
289. Student\_DisplaySingle(index);
290. }
292. printf(GREEN\_COLOR,"Continue?(y/n)");
293. if (getchar()=='n'){
294. break;
295. }
296. }
298. return;
299. }
301. /\*按姓名查询\*/
302. void Student\_SelectByName(){
303. while(1){
304. char name[20];
305. int index;
306. printf("Please input the student's Name, the one you want to search:");
307. scanf("%s",&name);
308. getchar();
309. index=Student\_SearchByName(name);
311. if (index==-1){
312. printf(GREEN\_COLOR,"Not exist!\n");
313. }else{
314. printf("The infomation of the student you want to search is:\n");
315. Student\_DisplaySingle(index);
316. }
318. printf(GREEN\_COLOR,"Continue?(y/n)");
319. if (getchar()=='n'){
320. break;
321. }
322. }
324. return;
325. }
327. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*成绩统计模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
328. /\*按平均值排序\*/
329. void Student\_SortByAverage(){
330. //IO\_ReadInfo();
331. system("clear");
332. int i, j;
333. student tmp;
335. for(i = 0; i < num\_of\_students; i++){
336. for (j = 1; j < num\_of\_students - i; j++){
337. if(students[j - 1].Average < students[j].Average){
338. tmp = students[j - 1];
339. students[j - 1] = students[j];
340. students[j] = tmp;
341. }
342. }
343. }
344. Student\_Display();
346. return;
347. }
349. /\*按学号排序\*/
350. void Student\_SortByID(){
351. //IO\_ReadInfo();
352. system("clear");
353. int i, j;
354. student tmp;
356. for(j = 0; j < num\_of\_students - 1; j++){
357. for(i = 0; i < num\_of\_students - 1; i++){
358. if(students[i].ID > students[i + 1].ID){
359. tmp = students[i];
360. students[i] = students[i + 1];
361. students[i + 1] = tmp;
362. }
363. }
364. }
365. Student\_Display();
367. return;
368. }
370. //各门课各分数段学生人数统计并打印
371. void Student\_Sort\_EachSubject(){
372. //    IO\_ReadInfo();
373. system("clear");
374. //IO\_ReadInfo();
375. int math1 = 0, math2 = 0, math3 = 0, math4 = 0, math5 = 0;
376. int engl1 = 0, engl2 = 0, engl3 = 0, engl4 = 0, engl5 = 0;
377. int phys1 = 0, phys2 = 0, phys3 = 0, phys4 = 0, phys5 = 0;
378. int elec1 = 0, elec2 = 0, elec3 = 0, elec4 = 0, elec5 = 0;
379. int CII1 = 0, CII2 = 0, CII3 = 0, CII4 = 0, CII5 = 0;
381. //math
382. for(int i = 0; i < num\_of\_students; i++){
383. if(students[i].math >= 90)
384. math1++;
385. else if(students[i].math >= 80)
386. math2++;
387. else if(students[i].math >= 70)
388. math3++;
389. else if(students[i].math >= 60)
390. math4++;
391. else
392. math5++;
393. }
395. //engl
396. for(int i = 0; i < num\_of\_students; i++){
397. if(students[i].engl >= 90)
398. engl1++;
399. else if(students[i].engl >= 80)
400. engl2++;
401. else if(students[i].engl >= 70)
402. engl3++;
403. else if(students[i].engl >= 60)
404. engl4++;
405. else
406. engl5++;
407. }
409. //phys
410. for(int i = 0; i < num\_of\_students; i++){
411. if(students[i].phys >= 90)
412. phys1++;
413. else if(students[i].phys >= 80)
414. phys2++;
415. else if(students[i].phys >= 70)
416. phys3++;
417. else if(students[i].phys >= 60)
418. phys4++;
419. else
420. phys5++;
421. }
423. //elec
424. for(int i = 0; i < num\_of\_students; i++){
425. if(students[i].elec >= 90)
426. elec1++;
427. else if(students[i].elec >= 80)
428. elec2++;
429. else if(students[i].elec >= 70)
430. elec3++;
431. else if(students[i].elec >= 60)
432. elec4++;
433. else
434. elec5++;
435. }
437. //CII
438. for(int i = 0; i < num\_of\_students; i++){
439. if(students[i].CII >= 90)
440. CII1++;
441. else if(students[i].CII >= 80)
442. CII2++;
443. else if(students[i].CII >= 70)
444. CII3++;
445. else if(students[i].CII >= 60)
446. CII4++;
447. else
448. CII5++;
449. }
451. printf(GREEN\_COLOR,"\t\t90分以上\t80 ~ 89\t\t70~79\t\t60~69\t\t60以下\t(单位:人)\n");
452. printf("--------------------------------------------------------------------------------------\n");
453. printf("math:\t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",math1, math2, math3, math4, math5);
454. printf("engl:\t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",engl1, engl2, engl3, engl4, engl5);
455. printf("phys:\t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",phys1, phys2, phys3, phys4, phys5);
456. printf("elec:\t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",elec1, elec2, elec3, elec4, elec5);
457. printf("CII: \t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",CII1, CII2, CII3, CII4, CII5);
458. printf("--------------------------------------------------------------------------------------\n");
460. return;
461. }
462. /\*显示学生信息\*/
463. void Student\_Display()
464. {
465. int i;
466. int Ranking = 0;
468. printf("%10s| %10s| %10s| %10s| %10s| %10s| %10s| %10s| %10s| \n","ID","Name","math","engl","phys","elec","CII","Average","Ranking");
469. printf("-----------------------------------------------------------------------------------------------------------------\n");
470. for (i=0;i<num\_of\_students;i++){
471. Ranking++;
472. printf("   %7d| %10s| %10.2f| %10.2f| %10.2f| %10.2f| %10.2f| %10.2f| %10d|\n", students[i].ID, students[i].Name,
473. students[i].math, students[i].engl, students[i].phys, students[i].elec, students[i].CII, students[i].Average, Ranking);
474. }
476. return;
477. }
479. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*学籍处理模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
480. //补考通知
481. void print\_information\_bukao(){
482. system("clear");
483. //初始挂科数为0
484. int num\_of\_students\_of\_fail = 0;
485. //初始补考人数设为-1,因计数从0开始
486. int num\_of\_students\_of\_studentbukao = -1;
487. for(int i = 0; i < num\_of\_students; i++){
488. if(students[i].math < 60)
489. num\_of\_students\_of\_fail++;
490. if(students[i].engl < 60)
491. num\_of\_students\_of\_fail++;
492. if(students[i].phys < 60)
493. num\_of\_students\_of\_fail++;
494. if(students[i].elec < 60)
495. num\_of\_students\_of\_fail++;
496. if(students[i].CII < 60)
497. num\_of\_students\_of\_fail++;
498. if(num\_of\_students\_of\_fail > 0){
499. num\_of\_students\_of\_studentbukao++;
500. }
501. if(num\_of\_students\_of\_studentbukao != -1){
502. students\_bukao[num\_of\_students\_of\_studentbukao] = students[i];
503. }
504. }
506. if(num\_of\_students\_of\_studentbukao == -1){
507. printf(GREEN\_COLOR,"No students need to bukao.\n");
508. }
510. printf(GREEN\_COLOR,"\n-----------补考通知(学生ID Name和挂科科目成绩)-------------\n");
511. for(int j = 0; j < num\_of\_students\_of\_studentbukao; j++){
512. printf("%d\t", students\_bukao[j].ID);
513. printf("%s\t", students\_bukao[j].Name);
514. if(students\_bukao[j].math < 60){
515. printf("math:  %5.2f|", students\_bukao[j].math);
516. }
517. if(students\_bukao[j].engl < 60){
518. printf("engl:  %5.2f|", students\_bukao[j].engl);
519. }
520. if(students\_bukao[j].phys < 60){
521. printf("phys:  %5.2f|", students\_bukao[j].phys);
522. }
523. if(students\_bukao[j].elec < 60){
524. printf("elec:  %5.2f|", students\_bukao[j].elec);
525. }
526. if(students\_bukao[j].CII < 60){
527. printf("CII:  %5.2f|", students\_bukao[j].CII);
528. }
529. printf("\n");
530. }
532. return;
533. }
534. //退学通知
535. void print\_information\_tuixue(){
536. system("clear");
537. //初始挂科数为0
538. int num\_of\_students\_of\_fail = 0;
539. //初始退学人数设为-1,因计数从0开始
540. int num\_of\_students\_of\_studentfail = -1;
541. for(int i = 0; i < num\_of\_students; i++){
542. if(students[i].math < 60)
543. num\_of\_students\_of\_fail++;
544. if(students[i].engl < 60)
545. num\_of\_students\_of\_fail++;
546. if(students[i].phys < 60)
547. num\_of\_students\_of\_fail++;
548. if(students[i].elec < 60)
549. num\_of\_students\_of\_fail++;
550. if(students[i].CII < 60)
551. num\_of\_students\_of\_fail++;
552. if(num\_of\_students\_of\_fail >= 3){
553. num\_of\_students\_of\_studentfail++;
554. students\_fail[num\_of\_students\_of\_studentfail] = students[i];
555. }
556. }
557. if(num\_of\_students\_of\_studentfail == -1){
558. printf("No students need to fail.\n");
559. }
560. printf(GREEN\_COLOR,"\n-----------退学通知(学生ID Name和挂科科目成绩)-------------\n");
561. for(int j = 0; j < num\_of\_students\_of\_studentfail; j++){
562. printf("%d\t", students\_fail[j].ID);
563. printf("%s\t", students\_fail[j].Name);
564. if(students\_fail[j].math < 60){
565. printf("math:  %5.2f|", students\_fail[j].math);
566. }
567. if(students\_fail[j].engl < 60){
568. printf("engl:  %5.2f|", students\_fail[j].engl);
569. }
570. if(students\_fail[j].phys < 60){
571. printf("phys:  %5.2f|", students\_fail[j].phys);
572. }
573. if(students\_fail[j].elec < 60){
574. printf("elec:  %5.2f|", students\_fail[j].elec);
575. }
576. if(students\_fail[j].CII < 60){
577. printf("CII:  %5.2f|", students\_fail[j].CII);
578. }
579. printf("\n");
580. }
582. return;
583. }
585. //升学学生
586. void print\_information\_up() {
587. system("clear");
588. //初始挂科数为0
589. int num\_of\_students\_of\_fail = 0;
590. //初始升学人数设为-1,因计数从0开始
591. int num\_of\_students\_of\_up = -1;
592. for(int i = 0; i < num\_of\_students; i++){
593. if(students[i].math < 60)
594. num\_of\_students\_of\_fail++;
595. if(students[i].engl < 60)
596. num\_of\_students\_of\_fail++;
597. if(students[i].phys < 60)
598. num\_of\_students\_of\_fail++;
599. if(students[i].elec < 60)
600. num\_of\_students\_of\_fail++;
601. if(students[i].CII < 60)
602. num\_of\_students\_of\_fail++;
603. if(num\_of\_students\_of\_fail <= 2){
604. num\_of\_students\_of\_up++;
605. next[num\_of\_students\_of\_up] = students[i];
606. }
607. }
608. printf(GREEN\_COLOR,"\nthe ID and name of students entering senior class:\n");
609. for(int j = 0; j < num\_of\_students\_of\_up; j++){
610. printf("ID: %d\t", next[j].ID);
611. printf("Name: %s\n", next[j].Name);
612. }
614. return;
615. }
617. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*文件读写模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
618. void IO\_WriteInfo(){
619. FILE \*fp = fopen("record.txt","a"); //追加写入
620. for(int i = 0; i < num\_of\_students; i++){
621. fprintf(fp,"%s %s %f %f %f %f %f %f", students[i].ID, students[i].Name, students[i].math,
622. students[i].engl, students[i].phys, students[i].elec, students[i].CII, students[i].Average);
623. }
624. printf("Done!\n");
625. fclose(fp);
627. return;
628. }
629. void IO\_ReadInfo(){
630. FILE \*fp = fopen("record.txt", "r");
631. if(fp == NULL){
632. IO\_WriteInfo();
633. fp = fopen("record.txt", "r");
634. }
635. for(int i = 0; i < num\_of\_students; i++){
636. fscanf(fp, "%s %s %f %f %f %f %f %f", &students[i].ID, &students[i].Name, &students[i].math,
637. &students[i].engl, &students[i].phys, &students[i].elec, &students[i].CII, &students[i].Average);
638. }
639. fclose(fp);
641. return;
642. }

**stu\_C.c:**

1. #include "../inc/stu.h"
3. //初始学生数为0
4. int num\_of\_students\_C = 0;
6. /\*求平均值\*/
7. float Average\_C(student stu){
8. return (stu.math + stu.engl + stu.phys + stu.elec + stu.CII) / 5;
9. }
11. /\*通过学号返回数组下标\*/
12. int Student\_SearchByID\_C(int id){
13. int i;
14. for(i=0;i<num\_of\_students\_C;i++){
15. if (students[i].ID == id)
16. {
17. return i;
18. }
19. }
20. return -1;
21. }
23. /\*通过姓名返回数组下标\*/
24. int Student\_SearchByName\_C(char name[]){
25. int i;
26. for(i=0;i<num\_of\_students\_C;i++){
27. if (strcmp(students[i].Name, name) == 0){
28. return i;
29. }
30. }
31. return -1;
32. }
34. /\*显示单条学生记录\*/
35. void Student\_DisplaySingle\_C(int index){
36. printf("%5s %9s %9s %9s %9s %9s %9s \t%9s  \n","学号","姓名","数学","英语","物理","电路","C指针","平均成绩");
37. printf("------------------------------------------------------------------------------------------\n");
38. printf("%7d %7s %10.2f %9.2f %9.2f %9.2f %10.2f %10.2f\n",students[index].ID,students[index].Name,
39. students[index].math, students[index].engl, students[index].phys, students[index].elec, \
40. students[index].CII, students[index].Average);
42. return;
43. }
45. /\*插入学生信息\*/
46. void Student\_Insert\_C(){
47. int a = 0;
48. while(1){
49. system("clear");
50. printf("请输入学号:");
51. scanf("%d",&students[num\_of\_students\_C].ID);
52. getchar();
53. for(int i = 0; i < num\_of\_students\_C - 1; i++){
54. if(students[num\_of\_students\_C].ID == students[i].ID){
55. printf("\e[31m\e[1m%s\e[0m","添加错误,该学生已存在!");
56. a = 1;
57. }
58. }
59. if(a == 1){
60. break;
61. }
63. printf("请输入姓名:");
64. scanf("%s",&students[num\_of\_students\_C].Name);
65. getchar();
67. printf("请输入数学成绩:");
68. do{
69. scanf("%f",&students[num\_of\_students\_C].math);
70. if(students[num\_of\_students\_C].math >= 0 && students[num\_of\_students\_C].math <= 100){
71. break;
72. }else
73. printf(RED\_COLOR,"输入的成绩不合规范(0 ~ 100),请再次输入.\n");
74. printf(">>>>>>:");
76. }while(1);
77. getchar();
79. printf("请输入英语成绩:");
80. do{
81. scanf("%f",&students[num\_of\_students\_C].engl);
82. if(students[num\_of\_students\_C].engl >= 0 && students[num\_of\_students\_C].engl <= 100){
83. break;
84. }else
85. printf(RED\_COLOR,"输入的成绩不合规范(0 ~ 100),请再次输入.\n");
86. printf(">>>>>>:");
88. }while(1);
89. getchar();
91. printf("请输入物理成绩:");
92. do{
93. scanf("%f",&students[num\_of\_students\_C].phys);
94. if(students[num\_of\_students\_C].phys >= 0 && students[num\_of\_students\_C].phys <= 100){
95. break;
96. }else
97. printf(RED\_COLOR,"输入的成绩不合规范(0 ~ 100),请再次输入.\n");
98. printf(">>>>>>:");
100. }while(1);
101. getchar();
103. printf("请输入电路成绩:");
104. do{
105. scanf("%f",&students[num\_of\_students\_C].elec);
106. if(students[num\_of\_students\_C].elec >= 0 && students[num\_of\_students\_C].elec <= 100){
107. break;
108. }else
109. printf(RED\_COLOR,"输入的成绩不合规范(0 ~ 100),请再次输入.\n");
110. printf(">>>>>>:");
112. }while(1);
113. getchar();
115. printf("请输入C指针成绩:");
116. do{
117. scanf("%f",&students[num\_of\_students\_C].CII);
118. if(students[num\_of\_students\_C].CII >= 0 && students[num\_of\_students\_C].CII <= 100){
119. break;
120. }else
121. printf(RED\_COLOR,"输入的成绩不合规范(0 ~ 100),请再次输入.\n");
122. printf(">>>>>>:");
124. }while(1);
125. getchar();
127. students[num\_of\_students\_C].Average=Average\_C(students[num\_of\_students\_C]);
128. num\_of\_students\_C++;
130. printf("\e[31m\e[1m%s\e[0m","继续?(Enter键继续/n退出)");
132. if (getchar()=='n'){
133. break;
134. }
135. }
137. //IO\_WriteInfo();
138. return;
139. }
141. /\*修改学生信息\*/
142. void Student\_Modify\_C(){
143. while(1){
144. int id;
145. int index;
147. printf("请输入您想要修改的学生的学号:");
148. scanf("%d",&id);
149. getchar();
150. index=Student\_SearchByID\_C(id);
151. if(index==-1){
152. printf(GREEN\_COLOR,"抱歉,无该生信息!\n");
153. }
154. else{
155. printf("您想要修改的学生的当前信息为:\n");
156. Student\_DisplaySingle\_C(index);
157. int choice;
158. printf("请选择您想要修改的具体信息:\n");
159. printf("-1.数学\n");
160. printf("-2.英语\n");
161. printf("-3.物理\n");
162. printf("-4.电路\n");
163. printf("-5.C指针\n");
164. printf("-6.全部信息\n");
165. printf("-0.退出\n");
166. printf(">>>>:");
167. scanf("%d", &choice);
168. if(1 == choice){
169. printf("请输入新的数学成绩:");
170. scanf("%f",&students[index].math);
171. students[index].Average=Average(students[index]);
172. }
173. if(2 == choice){
174. printf("请输入新的英语成绩:");
175. scanf("%f",&students[index].engl);
176. students[index].Average=Average(students[index]);
177. }
178. if(3 == choice){
179. printf("请输入新的物理成绩:");
180. scanf("%f",&students[index].phys);
181. students[index].Average=Average(students[index]);
182. }
183. if(4 == choice){
184. printf("请输入新的电路成绩:");
185. scanf("%f",&students[index].elec);
186. students[index].Average=Average(students[index]);
187. }
188. if(5 == choice){
189. printf("请输入新的C指针成绩:");
190. scanf("%f",&students[index].CII);
191. students[index].Average=Average(students[index]);
192. }
193. if(6 == choice){
194. printf("-- 请输入新的全部信息--\n");
195. printf("请输入学号:");
196. scanf("%d",&students[index].ID);
197. getchar();
199. printf("请输入姓名:");
200. scanf("%s",&students[index].Name);
201. getchar();
203. printf("请输入数学成绩:");
204. scanf("%f",&students[index].math);
205. getchar();
207. printf("请输入英语成绩:");
208. scanf("%f",&students[index].engl);
209. getchar();
211. printf("请输入物理成绩:");
212. scanf("%f",&students[index].phys);
213. getchar();
215. printf("请输入电路成绩:");
216. scanf("%f",&students[index].elec);
217. getchar();
219. printf("请输入C指针成绩:");
220. scanf("%f",&students[index].CII);
221. getchar();
223. students[index].Average=Average\_C(students[index]);
224. }
225. }
226. getchar();
227. printf(GREEN\_COLOR,"继续?(Enter键继续/n退出)");
228. if(getchar()=='n'){
229. break;
230. }
231. }
233. return;
234. }
236. /\*删除学生信息\*/
237. void Student\_Delete\_C(){
238. int i;
239. while(1){
240. int id;
241. int index;
242. printf("请输入您想要删除的学生的学号:");
243. scanf("%d",&id);
244. getchar();
245. index=Student\_SearchByID\_C(id);
247. if(index==-1){
248. printf("抱歉,无该生信息!\n");
249. }
250. else{
251. printf("您想要删除的学生的当前信息为:\n");
252. Student\_DisplaySingle\_C(index);
253. printf(GREEN\_COLOR,"确定删除吗?(y/n)");
254. if(getchar()=='y'){
255. for(i=index;i<num\_of\_students\_C-1;i++){
256. //把后边的对象都向前移动
257. students[i]=students[i+1];
258. }
259. num\_of\_students\_C--;
260. }
261. getchar();
262. }
264. printf(GREEN\_COLOR,"继续?(Enter键继续/n退出)");
265. if(getchar()=='n'){
266. break;
267. }
268. }
270. //IO\_WriteInfo();
271. return;
272. }
274. /\*按学号查询\*/
275. void Student\_SelectByID\_C(){
276. while(1){
277. int a;
278. int index;
279. printf("请输入您想要查询的学生的学号:");
280. scanf("%d",&a);
281. getchar();
282. index=Student\_SearchByID\_C(a);
284. if (index==-1){
285. printf(GREEN\_COLOR,"抱歉,无该生信息!\n");
286. }else{
287. printf("您查询的学生的信息为:\n");
288. Student\_DisplaySingle\_C(index);
289. }
291. printf(GREEN\_COLOR,"继续?(Enter键继续/n退出)");
292. if (getchar()=='n'){
293. break;
294. }
295. }
297. return;
298. }
300. /\*按姓名查询\*/
301. void Student\_SelectByName\_C(){
302. while(1){
303. char name[20];
304. int index;
305. printf("请输入您想要查询的学生的姓名:");
306. scanf("%s",&name);
307. getchar();
308. index=Student\_SearchByName\_C(name);
310. if (index==-1){
311. printf(GREEN\_COLOR,"抱歉,无该生信息!\n");
312. }else{
313. printf("您查询的学生的信息为:\n");
314. Student\_DisplaySingle\_C(index);
315. }
317. printf(GREEN\_COLOR,"继续?(Enter键继续/n退出)");
318. if (getchar()=='n'){
319. break;
320. }
321. }
323. return;
324. }
326. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*成绩统计模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
327. /\*按平均值排序\*/
328. void Student\_SortByAverage\_C(){
329. //IO\_ReadInfo();
330. system("clear");
331. int i, j;
332. student tmp;
334. for(i = 0; i < num\_of\_students\_C; i++){
335. for (j = 1; j < num\_of\_students\_C - i; j++){
336. if(students[j - 1].Average < students[j].Average){
337. tmp = students[j - 1];
338. students[j - 1] = students[j];
339. students[j] = tmp;
340. }
341. }
342. }
343. Student\_Display\_C();
345. return;
346. }
348. /\*按学号排序\*/
349. void Student\_SortByID\_C(){
350. //IO\_ReadInfo();
351. system("clear");
352. int i, j;
353. student tmp;
355. for(j = 0; j < num\_of\_students\_C - 1; j++){
356. for(i = 0; i < num\_of\_students\_C - 1; i++){
357. if(students[i].ID > students[i + 1].ID){
358. tmp = students[i];
359. students[i] = students[i + 1];
360. students[i + 1] = tmp;
361. }
362. }
363. }
364. Student\_Display\_C();
366. return;
367. }
369. //各门课各分数段学生人数统计并打印
370. void Student\_Sort\_EachSubject\_C(){
371. //    IO\_ReadInfo();
372. system("clear");
373. //IO\_ReadInfo();
374. int math1 = 0, math2 = 0, math3 = 0, math4 = 0, math5 = 0;
375. int engl1 = 0, engl2 = 0, engl3 = 0, engl4 = 0, engl5 = 0;
376. int phys1 = 0, phys2 = 0, phys3 = 0, phys4 = 0, phys5 = 0;
377. int elec1 = 0, elec2 = 0, elec3 = 0, elec4 = 0, elec5 = 0;
378. int CII1 = 0, CII2 = 0, CII3 = 0, CII4 = 0, CII5 = 0;
380. //math
381. for(int i = 0; i < num\_of\_students\_C; i++){
382. if(students[i].math >= 90)
383. math1++;
384. else if(students[i].math >= 80)
385. math2++;
386. else if(students[i].math >= 70)
387. math3++;
388. else if(students[i].math >= 60)
389. math4++;
390. else
391. math5++;
392. }
394. //engl
395. for(int i = 0; i < num\_of\_students\_C; i++){
396. if(students[i].engl >= 90)
397. engl1++;
398. else if(students[i].engl >= 80)
399. engl2++;
400. else if(students[i].engl >= 70)
401. engl3++;
402. else if(students[i].engl >= 60)
403. engl4++;
404. else
405. engl5++;
406. }
408. //phys
409. for(int i = 0; i < num\_of\_students\_C; i++){
410. if(students[i].phys >= 90)
411. phys1++;
412. else if(students[i].phys >= 80)
413. phys2++;
414. else if(students[i].phys >= 70)
415. phys3++;
416. else if(students[i].phys >= 60)
417. phys4++;
418. else
419. phys5++;
420. }
422. //elec
423. for(int i = 0; i < num\_of\_students\_C; i++){
424. if(students[i].elec >= 90)
425. elec1++;
426. else if(students[i].elec >= 80)
427. elec2++;
428. else if(students[i].elec >= 70)
429. elec3++;
430. else if(students[i].elec >= 60)
431. elec4++;
432. else
433. elec5++;
434. }
436. //CII
437. for(int i = 0; i < num\_of\_students\_C; i++){
438. if(students[i].CII >= 90)
439. CII1++;
440. else if(students[i].CII >= 80)
441. CII2++;
442. else if(students[i].CII >= 70)
443. CII3++;
444. else if(students[i].CII >= 60)
445. CII4++;
446. else
447. CII5++;
448. }
450. printf(GREEN\_COLOR,"\t\t90分以上\t80 ~ 89\t\t70~79\t\t60~69\t\t60以下\t(单位:人)\n");
451. printf("--------------------------------------------------------------------------------------\n");
452. printf("数学:\t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",math1, math2, math3, math4, math5);
453. printf("英语:\t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",engl1, engl2, engl3, engl4, engl5);
454. printf("物理:\t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",phys1, phys2, phys3, phys4, phys5);
455. printf("电路:\t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",elec1, elec2, elec3, elec4, elec5);
456. printf("C指针: \t\t %d\t\t %d\t\t %d\t\t %d\t\t %d\n",CII1, CII2, CII3, CII4, CII5);
457. printf("--------------------------------------------------------------------------------------\n");
459. return;
460. }
461. /\*显示学生信息\*/
462. void Student\_Display\_C()
463. {
464. int i;
465. int Ranking = 0;
467. printf("%10s| %10s| %10s| %10s| %10s| %10s| %10s| %10s| %10s| \n","学号","姓名","数学","英语","物理","电路","C指针","平均成绩","排名");
468. printf("-----------------------------------------------------------------------------------------------------------------\n");
469. for (i=0;i<num\_of\_students\_C;i++){
470. Ranking++;
471. printf("   %7d| %10s| %10.2f| %10.2f| %10.2f| %10.2f| %10.2f| %10.2f| %10d|\n", students[i].ID, students[i].Name,
472. students[i].math, students[i].engl, students[i].phys, students[i].elec, students[i].CII, students[i].Average, Ranking);
473. }
475. return;
476. }
478. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*学籍处理模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
479. //补考通知
480. void print\_information\_bukao\_C(){
481. system("clear");
482. //初始挂科数为0
483. int num\_of\_students\_C\_of\_fail = 0;
484. //初始补考人数设为-1,因计数从0开始
485. int num\_of\_students\_C\_of\_studentbukao = -1;
486. for(int i = 0; i < num\_of\_students\_C; i++){
487. if(students[i].math < 60)
488. num\_of\_students\_C\_of\_fail++;
489. if(students[i].engl < 60)
490. num\_of\_students\_C\_of\_fail++;
491. if(students[i].phys < 60)
492. num\_of\_students\_C\_of\_fail++;
493. if(students[i].elec < 60)
494. num\_of\_students\_C\_of\_fail++;
495. if(students[i].CII < 60)
496. num\_of\_students\_C\_of\_fail++;
497. if(num\_of\_students\_C\_of\_fail > 0){
498. num\_of\_students\_C\_of\_studentbukao++;
499. }
500. if(num\_of\_students\_C\_of\_studentbukao != -1){
501. students\_bukao[num\_of\_students\_C\_of\_studentbukao] = students[i];
502. }
503. }
505. if(num\_of\_students\_C\_of\_studentbukao == -1){
506. printf(GREEN\_COLOR,"没有学生需要补考\n");
507. }
509. printf(GREEN\_COLOR,"\n-----------补考通知(学生ID Name和挂科科目成绩)-------------\n");
510. for(int j = 0; j < num\_of\_students\_C\_of\_studentbukao; j++){
511. printf("%d\t", students\_bukao[j].ID);
512. printf("%s\t", students\_bukao[j].Name);
513. if(students\_bukao[j].math < 60){
514. printf("数学:  %5.2f|", students\_bukao[j].math);
515. }
516. if(students\_bukao[j].engl < 60){
517. printf("英语:  %5.2f|", students\_bukao[j].engl);
518. }
519. if(students\_bukao[j].phys < 60){
520. printf("物理:  %5.2f|", students\_bukao[j].phys);
521. }
522. if(students\_bukao[j].elec < 60){
523. printf("电路:  %5.2f|", students\_bukao[j].elec);
524. }
525. if(students\_bukao[j].CII < 60){
526. printf("C指针:  %5.2f|", students\_bukao[j].CII);
527. }
528. printf("\n");
529. }
531. return;
532. }
533. //退学通知
534. void print\_information\_tuixue\_C(){
535. system("clear");
536. //初始挂科数为0
537. int num\_of\_students\_C\_of\_fail = 0;
538. //初始退学人数设为-1,因计数从0开始
539. int num\_of\_students\_C\_of\_studentfail = -1;
540. for(int i = 0; i < num\_of\_students\_C; i++){
541. if(students[i].math < 60)
542. num\_of\_students\_C\_of\_fail++;
543. if(students[i].engl < 60)
544. num\_of\_students\_C\_of\_fail++;
545. if(students[i].phys < 60)
546. num\_of\_students\_C\_of\_fail++;
547. if(students[i].elec < 60)
548. num\_of\_students\_C\_of\_fail++;
549. if(students[i].CII < 60)
550. num\_of\_students\_C\_of\_fail++;
551. if(num\_of\_students\_C\_of\_fail >= 3){
552. num\_of\_students\_C\_of\_studentfail++;
553. students\_fail[num\_of\_students\_C\_of\_studentfail] = students[i];
554. }
555. }
556. if(num\_of\_students\_C\_of\_studentfail == -1){
557. printf("没有学生退学.\n");
558. }
559. printf(GREEN\_COLOR,"\n-----------退学通知(学生ID Name和挂科科目成绩)-------------\n");
560. for(int j = 0; j < num\_of\_students\_C\_of\_studentfail; j++){
561. printf("%d\t", students\_fail[j].ID);
562. printf("%s\t", students\_fail[j].Name);
563. if(students\_fail[j].math < 60){
564. printf("数学:  %5.2f|", students\_fail[j].math);
565. }
566. if(students\_fail[j].engl < 60){
567. printf("英语:  %5.2f|", students\_fail[j].engl);
568. }
569. if(students\_fail[j].phys < 60){
570. printf("物理:  %5.2f|", students\_fail[j].phys);
571. }
572. if(students\_fail[j].elec < 60){
573. printf("电路:  %5.2f|", students\_fail[j].elec);
574. }
575. if(students\_fail[j].CII < 60){
576. printf("C指针: %5.2f|", students\_fail[j].CII);
577. }
578. printf("\n");
579. }
581. return;
582. }
584. //升学学生
585. void print\_information\_up\_C() {
586. system("clear");
587. //初始挂科数为0
588. int num\_of\_students\_C\_of\_fail = 0;
589. //初始升学人数设为-1,因计数从0开始
590. int num\_of\_students\_C\_of\_up = -1;
591. for(int i = 0; i < num\_of\_students\_C; i++){
592. if(students[i].math < 60)
593. num\_of\_students\_C\_of\_fail++;
594. if(students[i].engl < 60)
595. num\_of\_students\_C\_of\_fail++;
596. if(students[i].phys < 60)
597. num\_of\_students\_C\_of\_fail++;
598. if(students[i].elec < 60)
599. num\_of\_students\_C\_of\_fail++;
600. if(students[i].CII < 60)
601. num\_of\_students\_C\_of\_fail++;
602. if(num\_of\_students\_C\_of\_fail <= 2){
603. num\_of\_students\_C\_of\_up++;
604. next[num\_of\_students\_C\_of\_up] = students[i];
605. }
606. }
607. printf(GREEN\_COLOR,"\n成功进入下一年级的学生的学号与姓名:\n");
608. for(int j = 0; j < num\_of\_students\_C\_of\_up; j++){
609. printf("学号: %d\t", next[j].ID);
610. printf("姓名: %s\n", next[j].Name);
611. }
613. return;
614. }
616. ///\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*文件读写模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
617. //void IO\_WriteInfo(){
618. //    FILE \*fp = fopen("record.txt","a"); //追加写入
619. //    for(int i = 0; i < num\_of\_students\_C; i++){
620. //  fprintf(fp,"%s %s %f %f %f %f %f %f", students[i].ID, students[i].Name, students[i].math,
621. //      students[i].engl, students[i].phys, students[i].elec, students[i].CII, students[i].Average);
622. //    }
623. //    printf("Done!\n");
624. //    fclose(fp);
625. //
626. //    return;
627. //}
628. //void IO\_ReadInfo(){
629. //    FILE \*fp = fopen("record.txt", "r");
630. //    if(fp == NULL){
631. //  IO\_WriteInfo();
632. //  fp = fopen("record.txt", "r");
633. //    }
634. //    for(int i = 0; i < num\_of\_students\_C; i++){
635. //  fscanf(fp, "%s %s %f %f %f %f %f %f", &students[i].ID, &students[i].Name, &students[i].math,
636. //      &students[i].engl, &students[i].phys, &students[i].elec, &students[i].CII, &students[i].Average);
637. //    }
638. //    fclose(fp);
639. //
640. //    return;
641. //}