

电子科技大学

实 验 报 告

实验一

一、实验室名称：

电子科技大学清水河校区主楼 A2-412

二、实验项目名称：

学生成绩管理系统

三、实验原理：

C 语言中的控制语句，数组，结构，指针，函数，文件操作函数。

四、实验目的：

掌握本学期所学 C 语言相关知识，能熟练运用指针与函数编写程序，学习文件操作相关函数。

五、实验内容：

学生主要信息：

学号，姓名，性别，年龄，学院，课程成绩（语文，数学，英语，专业课）。

学生系统主要功能：

录入学生信息，删除学生信息，修改学生某一信息，按学号查找某学生信息，查看学生成绩排名（查看单科和总成绩排名；显示所有学生排名情况，显示某一学生排名或显示某个名次学生），查看全体学生信息（当前学生人数，所有学生详细信息，各科平均分、最高分和最低分）。

六、实验器材（设备、元器件）：

硬件平台：

计算机配置：第八代英特尔酷睿 i5 处理器

CPU 内存：8GB 内存

软件平台：

操作系统和开发环境： 64 位操作系统，win10 家庭版

测试环境：dev c++

七、实验步骤：

1. 问题描述：

编写一个学生成绩管理系统，其学生主要信息包括学号，姓名，性别，年龄，学院，课程成绩（语文，数学，英语，专业课）。主要功能有增删查改，可以查看全班人数、可以对某课程按照成绩排序、可以查看某课程或者综合（总成绩）排名情况、可以查看某课程或者综合排名的第*i*名的学生信息等。

2. 算法分析与概要设计

注：关于学生结构定义如下

```
/**
 * 学生定义
 */
typedef struct student{
    char id[14]; // 学号: 13 位数字
    char name[50]; // 姓名
    char gender; // 性别: 1男, 2女
    int age; // 年龄
    char college[50]; // 学院: 学院名
    int sch; // 语文
    int sma; // 数学
    int sen; // 英语
    int spr; // 专业课
} Student;
```

(1) 主框架

1. 主函数

通过 while(1)始终保持每一次操作结束后回到主菜单，通过 getChoice 获得用户选择进行的功能，switch 实现功能跳转，输入 0 退出程序，输入其他清屏回到主菜单，等待重新输入。

2. 选择函数 getChoic

打印主菜单，通过 readInt 获得 0~6 的一个整数。

(2) 录入相关函数

1. 读入一个整数 readInt

输入：const char * hint(提示字符串), int min, int max(输入允许的最大最小值)

算法：通过 while 语句保证获得范围内的一个整数，错误输入则清屏并重新输入。

输出：返回输入的整数

2. 读入一个字符串 readStr

输入: const char * hint(提示字符串), char * result(需要输入的变量),
int min, int max(允许的最小长度和最大长度)

算法: 通过 while 语句保证获得范围内的一个字符串, 错误输入
则清屏并重新输入。输入 * 则返回-1, 否则就将输入放入需要输
入的变量。

输出: 逻辑值 -1 或 0。

(3) 文件操作相关函数

1. 增加学生信息 appendStudent

输入: Student stu(学生结构)

算法: 打开文件, 通过 fwrite 将指定学生结构追加至文件末尾,
关闭文件。

输出: 保存成功返回 0, 否则返回 -1

2. 查找学生信息 searchStudent

输入: Student * stu(学生结构:用来保存找到的学生), char * id(学生
学号)

算法: 打开文件, 通过 fread 读取学生信息, 利用循环和 strcmp
比较学号, 相同则退出循环, 关闭文件; 否则返回-1, 关闭文件。

输出: 逻辑值 -1 或学号在文件中序号。

3. 保存指定位置的学生信息 modifyStudent

输入: Student stu(学生结构:用来保存找到的学生), int p(文件中的
序号)

算法: 打开文件, 通过 fseek 移动到指定位置, 通过 fwrite 将指定
信息更改, 关闭文件。

输出: 修改成功返回 0, 否则返回 -1

4. 从文件读取所有学生信息 readAllStudents

输入: Student **stu(一个指针的地址, 用来返回所有学生数据)

算法: (核心算法, 见后)

输出: 读取的学生个数。

(4) 录入相关函数

1. 录入学生信息 addS

算法: 打印提示语句, 输入学生信息 addS_STU, 增加学生信息
appendStudent, 打印保存成功及新保存的学生信息。

2. 输入学生信息 addS_STU

输入: Student *stu(保存输入的学生信息)

算法：调用 readStr 及 readInt 打印提示语句，并获得输入的正确信息，当使用 readStr 时可利用*退出。

输出：输入成功返回 0，中途退出返回 -1。

(5) 输出学生列表相关函数

1. 打印学生列表标题 head

2. 打印一个学生信息 print

输入：Student stu(将被打印的学生)

3. 输出所有学生信息 listS

算法：readAllStudents 获得全部学生信息，通过循环比较和累加得到各科最低分，最高分，平均分并输出。调用 info 输出某一科最高或最低分学生信息。关闭文件。

4. 显示某一科最高或最低分学生 info

输入：int val, int num, Student * stu, int subj, int type(该科最高或最低分数，学生总数，学生数组，科目编号，最高或最低分)

输出：该学生信息。

(6) 查询学生信息相关函数

1. 查询一个学生信息 searchS

算法：readStr 输入待查找学号，searchStudent 进行比对，若 searchStudent 返回-1，查找失败；否则输出查找到的学生信息。

(7) 修改学生信息相关函数

1. 删除一个学生信息 modifyS

算法：readStr 输入待删除学号，searchStudent 进行比对，若 searchStudent 返回-1，查找失败；否则输出查找到的学生信息，并询问是否放弃删除，readInt 获得用户选择是否继续删除。若确认删除，strcpy 替换学号为 DELETE，modifyStudent 保存成功，打印已删除提示，否则打印删除失败提示。

2. 选择学生信息中的一项进行修改 modifyS_edit

输入：Student *stu

算法：readInt 选择要修改的学生信息，0 返回主菜单。利用字符数组常量打印不同的提示语句；利用 switch 及 readStr、readInt 完成修改。利用 readStr 修改时可中途退出。

输出：修改成功返回修改项序号，放弃修改该项返回 -1，返回主菜单则返回 0。

(8) 排序统计相关函数

1. 输出学生名次列表标题 headIdx

增加名次列

2. 指定学生信息和名次进行输出 printIdx

输入: Student stu(指定学生信息), int index(名次)

增加学生名次信息

3. 获取一个学生信息中指定科目的成绩 score

输入: Student stu(指定学生信息), int subj (科目编号 0.语文, 1.数学, 2.英语, 3.专业课, 4.总成绩, 5.平均)

算法: switch

输出: 指定科目的成绩

4. 选择排序项排序学生信息 sortS

(核心算法, 见后)

5. 交换当前学生和下一个学生的学生信息 change

输入: Student * stu(指定当前学生, 与下一个学生相交换)

6. 成绩排序 sort

输入: Student *array, int n, int subj

3.核心算法的详细设计与实现

• 从文件读取所有学生信息 readAllStudents

```
FILE * fp = fopen(StuFile, "rb");
if (fp == NULL) { return 0; }
fseek(fp, 0, SEEK_END);
int size = ftell(fp);
fseek(fp, 0, SEEK_SET);
*stu = (Student *)malloc((size / sizeof(Student) + 1) * sizeof(Student));
Student * cur = *stu;
int n = 0;
while (!feof(fp)) {
    if (fread(cur, sizeof(Student), 1, fp) == 1 && (strcmp((*cur).id, "DELETED") != 0)) {
        n++;
        cur++;
    }
}
fclose(fp);
return n;
```

• 选择排序项排序学生信息 void sortS

• 定义 const char *hint[5] = {

"语文",
"数学",
"英语",
"专业课",
"总成绩" };

Student *stu;

char id[20];

•调用函数 readInt 打印: "\n 请选择要排序的项 (1.语文, 2.数学, 3.英语, 4.专业课, 5.总成绩/平均成绩 0.

返回主菜单): "获得 0 至 5 的一个整数赋值给 subj;若 subj 为 0 退出, 否则:

- 调用函数 readAllStudents, 将所有学生信息存在 stu 中, 将学生人数赋值给 n:

```
int n = readAllStudents(&stu);
```

- 调用函数 sort, 排序:

```
sort(stu, n, subj - 1);
```

• 再次调用函数 readInt 打印: "\n 请选择操作 (1.显示所有同学排名, 2.查看某一名次同学, 3.查看某一同学名次 0.返回主菜单):" 获得 0 至 3 的一个整数赋值给 op;若 op 为 0 退出, 否则:

- 当 op 为 1:

```
printf("%s排名如下: \n", hint[subj - 1]);
headIdx();
int idx = 0;
int sco = -1;
for(int i = 0; i < n; i++){
    if (score(stu[i], subj - 1) != sco) {
        sco = score(stu[i], subj - 1);
        idx = i + 1;
    }
    printIdx(stu[i], idx);
}
```

- 当 op 为 2:

```
int p = readInt("请输入要查看的名次: ", 1, n);
printf("\n%s第 %d 名学生信息如下: \n", hint[subj - 1], p);
headIdx();
int sco = score(stu[p - 1], subj - 1);
for (int i = p - 1; i < n; i++) {
    if (score(stu[i], subj - 1) == sco) {
        printIdx(stu[i], p);
    }
}
```

- 当 op 为 3:

```
char id[14];
if (readStr("请输入待查名次学生的学号[13位, * 退出]: ", id, 13, 13)) { break; }
int idx = 0;
int sco = -1;
int found = 0;
for (int i = 0; i < n; i++){
    if (score(stu[i], subj - 1) != sco) {
        sco = score(stu[i], subj - 1);
        idx = i + 1;
    }
    if (!strcmp(stu[i].id, id)) {
        headIdx();
        printIdx(stu[i], idx);
        found = 1;
        break;
    }
}
if (found == 0) { printf("没有这个同学\n"); }
```

退出并关闭文件。

(由于花了大量时间改进算法, 时间不足以完成流程图)

八、实验数据及结果分析:

图 1-主菜单

图 2.2.2-录入学生信息错误实例 2

```

学号 [13 位, 输入 * 退出]: abcd123456789
姓名 [输入 * 退出]: aaa
性别 [1:男, 2:女, 0:退出]: 1
年龄 [0 - 150]: 1
学院 [输入 * 退出]: 1
语文成绩 [0 - 100]: 11
数学成绩 [0 - 100]: 11
英语成绩 [0 - 100]: 11
专业课成绩 [0 - 100]: 11

已录入学生信息:
-----
      学号      姓名 性别 年龄      学院  语文  数学  英语  专业  合计  平均
-----
abcd123456789    aaa   男   1          1    11   11   11    11   44   11
请按任意键继续. . .

```

图 3.1-删除学生正确实例:

请输入待删除学生的学号[13位, * 退出]: 2018081304023										
学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均
2018081304023		1	男	1	1	1	1	1	4	1

是否删除该学生信息[1: 删除, 0: 放弃]: 1

已成功删除该学生的信息

请按任意键继续. . .

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均	
2018081304001	张三	男	19	计算机	90	80	85	90	345	86	
2018081304002	李四	女	18	医学院	95	87	90	82	354	88	
2018081304003	王五	男	19	物理	83	97	86	91	357	89	
					最低分	83	80	85	82	345	86
					最高分	95	97	90	91	357	89
					平均分	89	88	87	87	352	87
当前有 3 个学生信息											

图 3.2-删除学生边界实例:

```
请输入待删除学生的学号[13位, * 退出]: 2
请重新输入: 1111111111111111111111
没有找到该学生信息。
请按任意键继续. . .
```

当输入低于 13 位学号会提示重新输入,但超出 13 位会自动取前 13 位搜索。由于采用了 readStr 函数,学号输入的边界讨论与录入功能处的学号输入相同,不再重复。

图 4.1-修改学生信息正确实例

学生的原信息如下：

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均
2018081304001	张三	男	19	计算机	90	80	85	90	345	86

请选择要修改的学生信息 (1. 学号, 2. 姓名, 3. 性别, 4. 年龄, 5. 学院, 6. 语文, 7. 数学, 8. 英语, 9. 专业课, 0. 返回主菜单): 9
 请修改学生的专业课成绩 [0 - 100]: 89
 已成功修改学生的信息如下：

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均
2018081304001	张三	男	19	计算机	90	80	85	89	344	86

学生的原信息如下：

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均
2018081304001	张三	男	19	计算机	90	80	85	89	344	86

请选择要修改的学生信息 (1. 学号, 2. 姓名, 3. 性别, 4. 年龄, 5. 学院, 6. 语文, 7. 数学, 8. 英语, 9. 专业课, 0. 返回主菜单): 10
 请重新输入: 2. 4
 请修改学生的姓名 [输入 * 退出]:

图 4.1.2-修改学生信息边界实例 2

学生的原信息如下:

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均
2018081304001	张三	男	19	计算机	90	80	85	89	344	86

请选择要修改的学生信息(1.学号, 2.姓名, 3.性别, 4.年龄, 5.学院, 6.语文, 7.数学, 8.英语, 9.专业课, 0.返回主菜单):4
请修改学生的年龄 [0 - 150]: 500
请重新输入: -30
请重新输入: 19.7
已成功修改学生的信息如下:

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均
2018081304001	张三	男	19	计算机	90	80	85	89	344	86

请按任意键继续. . .

由于采用了 readInt 函数, 选择输入时输入负数、超出范围的整数、字符均提示重新输入, 输入小数取符号前的数字保存。由于清屏,图 4.1.1 看不到测试的输入, 只能看到结果为重新输入。

由于采用了 readInt 函数和 readStr 函数, 修改时的边界及错误测试均与录入学生学习时各项的输入测试相同, 不再重复。

图 5.1 -查找学生信息正确实例

请输入待查找学生的学号[13位, * 退出]: 2018081304002
查找结果:

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均
2018081304002	李四	女	18	医学院	95	87	90	82	354	88

请按任意键继续. . .

请输入待查找学生的学号[13位, * 退出]: 2018081304023
没有找到该学生信息。
请按任意键继续. . .

可以看到已被删除的学生不再被找到。

图 5.2 -查找学生信息边界实例

请输入待查找学生的学号[13位, * 退出]: 1111
请重新输入: 1111111111111111
没有找到该学生信息。
请按任意键继续. . .

当输入低于 13 位学号会提示重新输入, 但超出 13 位会自动取前 13 位搜索。由于采用了 readStr 函数, 学号输入的边界讨论与录入功能处的学号输入相同, 不再重复。

图 6.1-成绩排序正确实例 1

请选择要排序的项 (1. 语文, 2. 数学, 3. 英语, 4. 专业课, 5. 总成绩/平均成绩 0. 返回主菜单):1

请选择操作 (1. 显示所有同学排名, 2. 查看某一名次同学, 3. 查看某一同学名次 0. 返回主菜单):1

语文排名如下:

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均	名次
2018081304002	李四	女	18	医学院	95	87	90	82	354	88	1
2018081304001	张三	男	19	计算机	90	80	85	89	344	86	2
2018081304003	王五	男	19	物理	83	97	86	91	357	89	3

请选择操作 (1. 显示所有同学排名, 2. 查看某一名次同学, 3. 查看某一同学名次 0. 返回主菜单):2

请输入要查看的名次: 2

语文第 2 名学生信息如下:

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均	名次
2018081304001	张三	男	19	计算机	90	80	85	89	344	86	2

请选择操作 (1. 显示所有同学排名, 2. 查看某一名次同学, 3. 查看某一同学名次 0. 返回主菜单):3

请输入待查名次学生的学号[13位, * 退出]: 2018081304003

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均	名次
2018081304003	王五	男	19	物理	83	97	86	91	357	89	3

请选择操作 (1. 显示所有同学排名, 2. 查看某一名次同学, 3. 查看某一同学名次 0. 返回主菜单):

以语文（选项 1）为例，可进行三种操作，结果如上图所示。按 0 则可更换科目。再次按 0 则回到主菜单。

由于保证了成绩均为 0-100 的整数，因此平均分，最高分，最低分，总分得以正确运算，保证了排序与查看的准确。（图 6.1 及图 7.1）

图 6.2-成绩排序正确实例 2

请选择要排序的项 (1. 语文, 2. 数学, 3. 英语, 4. 专业课, 5. 总成绩/平均成绩 0. 返回主菜单):3

请选择操作 (1. 显示所有同学排名, 2. 查看某一名次同学, 3. 查看某一同学名次 0. 返回主菜单):1

英语排名如下:

学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均	名次
2018081304002	李四	女	18	医学院	95	87	90	82	354	88	1
2018081304004	赵六	女	19	光电	89	85	90	90	354	88	1
2018081304003	王五	男	19	物理	83	97	86	91	357	89	3
2018081304001	张三	男	19	计算机	90	80	85	89	344	86	4

请选择操作 (1. 显示所有同学排名, 2. 查看某一名次同学, 3. 查看某一同学名次 0. 返回主菜单):

当出现并列成绩时，名次做相同处理。

图 7.1 查看学生信息正确实例 1

英语 最高分: 90	学号	姓名	性别	年龄	学院	语文	数学	英语	专业	合计	平均
2018081304002	李四	女	18	医学院	95	87	90	82	354	88	
2018081304004	赵六	女	19	光电	89	85	90	90	354	88	

九、总结及心得体会：

实验中发现的不足有不断改进。最开始当用户输入非合理值时程序运行结果会出错，经过不断地改进，逐渐避免了此类情况地发生，但对函数功能的分割较细。

本程序存在一个比较严重的问题，就是删除操作并不是真的把学生信息从文件中删除，只是将其学号更改为特殊字符串 `DELETE`，在其他功能中避免此类信息。但当删除量逐渐增大，文件也逐渐增大，最后如果一个文件中几乎全是被删除的学生信息，只有极少量学生信息可被操作，那么此时运行的时间成本和空间成本会大大增加。以及输入学号含非数字的问题没有解决。

还有在编程过程中我不够耐心，对指针与数组、指针与结构、指针与函数的结合应用不够熟练，理解还不到位，花了大量时间调试及更改才最终成功完成程序。

十、对本实验过程及方法、手段的改进建议及展望：

后续可用 `if` 语句改善学号问题，并探索解决删除学生信息的方法。

以及以后编程应先列好大框架，明确每个函数的功能，最大程度避免大量的重复代码。

报告评分：

指导教师签字：