

选题三：成绩管理系统（难度等级 A）

一、系统功能

成绩管理系统要求实现学生成绩管理的基本功能，包括学生、课程、成绩信息的录入、删除、查找和导入/导出等。学生信息的属性包括学号、姓名、性别和年龄，课程信息包括课号、课名和学分属性，选课信息包括学号、课号和成绩属性。

功能要求：

1. 装入：从指定的文件或默认的文件中导入学生信息、课程信息以及选课信息，并创建成绩单；

2. 显示：分屏显示系统中所有学生信息、课程信息、成绩信息；

3. 输入：通过键盘输入学生记录或课程记录或选课记录到成绩管理系统中；

4. 删除：删除一条已经存在的学生记录或课程记录或选课记录

5. 查找：根据用户输入的属性值查找符合条件的学生信息或课程信息或成绩信息；

6. 输出：把当前所有的学生信息、课程信息、选课信息以及学生的成绩单全部输出到指定的文件中；

1.导入初始数据

2.显示信息

3.输入记录

4.删除记录

5.查询信息

6.批量导出数据

0.退出

请给出你的选择：

程序执行过程：循环显示主菜单，用户在“请给出你的选择：”后面输入选项，即按照功能列表输入数字 0~6 中的任意数字，按回车后，执行相应的功能。请参照前面的“菜单设计练习”的要求建立右图所示主菜单结构。

二、菜单功能

1. 导入初始数据

从指定的文件或默认的文件中导入学生信息、课程信息、选课信息，并根据选课记录以及课程的学分，创建成绩单，计算加权平均分，再按照平均分的高低排序。

2. 显示信息（显示所有记录）

按学号升序顺序显示所有学生记录，每屏显示 10 条记录，每显示 10 条记录，按任意键继续显示下一屏；按课程号升序显示课程记录，按平均成绩降序显示成绩单。

3. 输入记录（具有修改功能）

从键盘输入学生信息或课程信息或选课信息，增加新的记录。每输入一条记录都检测是否已经存在该记录，如果已经存在则提示是否替换，否则直接增加记录。每输入完一条记录都要提示是否继续输入下一条记录，根据输入决定是否继续操作。

4. 删除记录（逐条记录确认）

根据输入的学生姓名或者课程名称，删除相关学生或课程具体记录，如果有多条满足条件的记录，则逐一确认是否删除。

5. 查询信息

输入学号或姓名查询学生相关信息，输入课程号或课程名查询课程相关信息，或者根据学号查询该学生的选课信息和成绩单及排名情况。

6. 批量导出数据

把当前所有的学生信息、课程信息、选课信息以及学生的成绩单全部输出到磁盘文件中，文件名以时间字符串命名。

0. 退出

结束程序运行，退出成绩管理系统程序。

上述 7 个主菜单选项中的显示、输入、删除以及查询功能都包含 3 个子功能，运行时给出对应的下一级子菜单选项，根据输入的选择，调用相关的函数。

三、本课题基本要求

1. 数据结构

用结构体数组实现成绩信息的记录和管理。

(1) 每个学生数组元素为一个结构体变量，结构如下：

```
typedef struct
{
    char xh[10]; //学号
    char xm[20]; //姓名, 10 个汉字以内
    char xb[3];  //性别, 只能保存一个汉字
    int nl;      //年龄
}Student;
```

(2) 每一门课程的信息为一个结构体变量，结构如下：

```
typedef struct
{
    char kh[10]; //课号, 长度 10 字符以内
    char km[25]; //课姓名, 12 个汉字以内
    float xf;    //课程学分, 为一个浮点数
}Course;
```

(3) 学生的选课信息为一个结构体变量，结构如下：

```
typedef struct
{
    char xh[10]; //课号, 长度 10 字符以下
    char kh[10]; //课程号
    float cj;    //成绩, 为一个浮点数
}Select;
```

(4) 所有学生的成绩单信息为一个结构体变量，结构如下：

```
typedef struct
{
    char xh[10]; //学号
    char xm[20]; //姓名
    float cj[30]; //各门课程成绩
    float zpj;    //加权平均分
    float zxf;    //总修习学分
}Score;
```

在主函数中定义结构体数组 `Student slist[]`, 用作记录学生信息的数组存储; 定义结构体数组 `Course clist[]`, 用作记录课程信息的数组存储; 定义结构体数组 `Select`

sclist[],用作保存学生选课信息和成绩,每名学生的—门选课对应—条记录;定义结构体数组 Score cjlist[],用作保存所有学生的成绩信息和统计得出的平均分、总学分以及排名情况,每名学生对应—条记录。

2. 各函数功能

以下函数原型说明中出现的函数为本课程设计的基本要求。请不要随便更改函数原型。

(1) 初始化数据

void init(Student slist[],int *scount,Course clist[],int *ccount,Score sclist[],int *sccount);程序启动运行时通过指定的文件装入(如果没有指定或指定失败,则从默认的文件装入)相关必要信息,包括学生信息、课程信息已经学生选课信息,并将学生记录条数、课程记录条数以及选课记录条数分别保存到指针变量*scount、*ccount 和*sccount 中。

(2) 建立成绩单

int create_cjlist(Select sclist[],int sccount,Course clist[],int ccount,Score cjlist[]);根据选课数组 sclist 中的学号,查询学生表中对应的姓名,把学号和姓名填入成绩单数组 cjlist 的—条记录中,再根据选课数组 sclist 中的课程号按照课程信息表中的课程顺序,把选课成绩填入到成绩单对应记录中,最后根据课程学分和所得成绩计算加权平均分和总学分。例如:某同学选了 3 门课程,成绩分别是 90、85 和 98 分,而这 3 门课程的学分分别为 2、3.5 和 3 学分,则总该同学的总修习学分为 $2+3.5+3=8.5$ 学分,加权平均分为 $(90*2+85*3.5+98*3)/8.5=85.765$ 分。

(3) 显示数据

void disp_stud(Student slist[],int scount);显示所有学生信息,每显示 10 条记录暂停—次,scount 为学生数组元素个数。

void disp_course(Course clist[],int ccount);显示所有课程信息,每显示 10 条记录暂停—次,ccount 为课程数组元素个数。

void disp_cjlist(Score cjlist[],int cjcount,Course clist[],int ccount,Student slist[],int scount);显示所有学生成绩单信息,每显示 10 行暂停—次,cjcount 为数组元素个数。显示的学生姓名来自于学生信息表 slist,课程名来自于课程信息表 clist。

上述三个显示函数运行之前都要调用排序函数,使得记录显示有序。排序函数如下:

void sort_slist(Student slist[],int scount);按照学号升序排序。

void sort_clist(Course clist[],int ccount);按照课程号升序排序。

void sort_cjlist(Score cjlist[],int cjcount);按照平均成绩降序排序。

(4) 数据输入

数据输入的功能主要是增加对应的数据记录,各函数的定义如下:

int input_stud(Student slist[],int scount);从键盘输入若干条学生信息记录,依次存放到学生信息结构体数组 slist 中,scount 为数组原有学生记录数,函数返回最后的学生记录数。

int input_course(Course clist[],int ccount);从键盘输入若干条课程信息记录,依次存放到课程信息结构体数组 clist 中,ccount 为数组原有课程记录数,函数返

回最后的课程记录数。

上述两个函数运行时，每输入一条记录都检测是否已经存在相同记录，如果已经存在则提示是否替换，否则直接增加记录，使得函数同时具有增加记录和修改已有记录的功能。每输入完一条记录都要提示是否继续输入下一条记录，根据输入决定是否继续操作。

`int input_select(Score sclist[],int sccount);`从键盘输入若干条选课记录，依次存放到学生选课信息结构体数组 `sclist` 中，`sccount` 为数组原有选课记录数，函数返回最后的学生选课记录总数。如果课程暂时还没有成绩则用-1 表示。该函数除了要兼具增加和修改记录的功能外，还要保证输入数据的正确性，如果输入的学号不是学生信息表中存在的学号，或者输入的课程号不是课程信息表中存在的课程号，或者成绩超出范围（ $-1 \leq \text{成绩} \leq 120$ ）则提示输入错误并要求重新输入。

（5）删除记录

`int delete_stud(Student slist[],int sccount);`根据姓名或学号删除满足条件的若条学生记录，返回数组中的还剩下的记录数。

`int delete_course(Course clist[],int ccount);`根据课程名或课程号删除满足条件的若条课程记录，返回数组中还剩下的记录数。

`int delete_select(Select sclist[],int sccount);`根据学号删除满足条件的若条选课记录，返回数组中还剩下的记录数。

（6）查询信息

`void query_stud(Student slist[],int sccount);`根据学号或姓名查询学生记录并显示。

`void query_course(Course clist[],int ccount);`根据课程号或课程名查询课程记录并显示。

`void query_cjlist(Score cjlist[],int cjcount, Course clist[],int ccount, Student slist[],int sccount);`根据学号查询成绩单，显示对应同学的所有课程成绩、平均分、学分以及排名情况，并显示。

（7）批量导出数据

`void write_to_file1(Student slist[],int sccount, Course clist[],int ccount, Select sclist[],int sccount);` ~~以时间序列字符串为文件名~~，把学生信息表、课程信息表和选课信息表分别保存都对应的磁盘文件中。例如，学生文件名为 `student20190918120423.txt`、课程文件名为 `course20190918120430.txt`、选课信息文件名为 `select201909181258.txt`。

`void write_to_file2(Score cjlist[],int cjcount, Course clist[],int ccount, Student slist[],int sccount);` ~~以时间序列字符串为文件名~~，把成绩单保存的磁盘文件中，包括名次、学号、姓名、性别、各门课成绩、平均分以及总学分。