

C 语言程序设计

实验任务：

- 一、 进一步掌握 C 语言的编程方法；进一步掌握数组的定义与使用；进一步掌握函数的定义和函数调用方法；
- 二、 学习和掌握结构体的定义和使用方法。
- 三、 学习和掌握文件的基本输入输出方法。
- 四、 编译并运行你的程序。
- 五、 提交**正式**的实验报告

- 六、 程序的具体要求如下：
编写一个通过键盘或文件输入学生信息，并将学生信息输出到文件和屏幕的程序。

- 七、 对实验报告的要求：
 - 1) 实验报告文档的书写格式与 Lab Assignment 1 的要求一致。
 - 2) 实验报告内容要求有：
 - 实验目的
 - 实验要求
 - 实验程序实现方法的流程图
 - 对流程图的简要说明
 - 实验结果
 - 对本课程的意见和建议

- 八、 对编程的具体说明如下：
 - 1) 定义一个 student 结构体如下：

```
struct student {  
    char name[20];    //姓名  
    float score[4];    //四门功课的成绩  
    float average;    //四门功课的平均成绩  
};
```

其中，name 中存放**同学的姓名**；score[4]数组中存放**同学的英语、数学、物理、化学成绩**；average 中存放上述四门功课的平均成绩。

- 2) 编写下面六个函数。其中 6 个函数中的后 2 个函数是文件操作函数。
函数的具体要求：

void inputFromKeyboard(struct student stu[], int num)

从键盘输入学生信息的函数。stu 为结构体数组，num 为数组中元素的个数。

void outputToScreen(struct student stu[], int num)

将学生信息输出到屏幕的函数。stu 为结构体数组，num 为数组中元素的个数。

void averageScore(struct student stu[], int num)

计算学生平均成绩的函数。stu 为结构体数组，num 为数组中元素的个数。

void sortByAverageScore(struct student stu[], int num)

根据学生的平均成绩对学生信息整体排序的函数。stu 为结构体数组，num 为数组中元素的个数。

void inputFromFile(struct student stu[], int num, FILE *fp)

从文件输入学生信息的函数。stu 为结构体数组，num 为数组中元素的个数，fp 为以“读”方式打开的文本文件的指针。

void outputToFile(struct student stu[], int num, FILE *fp)

将学生信息输出到文件的函数。stu 为结构体数组，num 为数组中元素的个数，fp 为以“写”方式打开的文本文件的指针。

3) **编写主函数**对你所编写的上述六个函数进行调用验证。主函数可分下面几个阶段编写：

第 1 个阶段 (可参考在第 11 章的实验作业)：

定义 3 个学生的结构体数组，编写函数 inputFromKeyboard、outputToScreen 和 averageScore。

调用函数 inputFromKeyboard 由键盘输入学生的信息(姓名、英语、数学、物理、化学成绩)，调用函数 outputToScreen 将输入的信息显示到屏幕上；

上述两个函数调试正确后，编写函数 averageScore 并调用，计算每个同学的平均成绩，然后调用函数 outputToScreen，将计算后的数据信息显示到屏幕上

第 2 个阶段：

编写函数 outputToFile。

调用函数 inputFromKeyboard 由键盘输入 3 个学生的信息(姓名、英语、数学、物理、化学成绩)，并以文本“写”方式打开文本 student1.txt 文件，调用函数 outputToFile，将这 3 个同学的信息存入 student1.txt 文本文件，关闭该文件；使用 windows XP 的“记事本”文本编辑器打开文本文件 student1.txt 查看是否正确。

其中，在程序运行后文本文件 student1.txt 中可能的内容及格式为：

刘翔 88.0 76.0 78.0 70.0 -107374176.0

姚明 98.0 67.0 79.0 65.0 -107374176.0

李娜 89.0 79.0 85.0 79.0 -107374176.0

第 3 个阶段:

编写函数 `inputFromFile`。

将上面两个阶段中调用函数 `inputFromKeyboard` 的语句变成注释行。

以文本“读”方式打开文本 `student1.txt` 文件，调用函数 `inputFromFile`，将这 3 个同学的信息读入并存放到学生信息结构体数组中，关闭该文件；

调用函数 `averageScore` 计算每个同学的平均成绩，并将平均成绩存储在结构数组的相应位置处。

调用函数 `outputToScreen` 将学生的完整的信息显示到屏幕上；

以文本“写”方式打开文本 `student2.txt` 文件，调用函数 `outputToFile`，将这 3 个同学的计算有平均成绩的所有信息存入 `student2.txt` 文本文件，关闭该文件；

在这段程序运行后文本文件 `student2.txt` 中可能的内容及格式为：

```
刘翔 88.0 76.0 78.0 70.0 78.0
姚明 98.0 67.0 79.0 65.0 77.3
李娜 89.0 79.0 85.0 79.0 83.0
```

第 4 个阶段，完整调试 6 个函数：

编写函数 `sortByAverageScore`。

在第 3 阶段编写的程序段后面添加下面的语句：

调用函数 `sortByAverageScore` 根据学生的平均成绩对学生信息整体排序。

调用函数 `outputToScreen` 将学生的完整的信息显示到屏幕上；

以文本“写”方式打开文本 `student3.txt` 文件，调用函数 `outputToFile`，将排序后的这 3 个同学的所有信息存入 `student3.txt` 文本文件，关闭该文件；

在程序运行后文本文件 `student3.txt` 中可能的内容及格式为：

```
李娜 89.0 79.0 85.0 79.0 83.0
刘翔 88.0 76.0 78.0 70.0 78.0
姚明 98.0 67.0 79.0 65.0 77.3
```

结束主程序。

九、 拓展编程（选作）：

- 1) 假设你已经使用文本编辑器将你们班 15 个同学的信息编辑为一个文本文件 `myclass.txt`，其中每个同学的信息包含：姓名、四门课程的成绩。

`myclass1.txt` 中可能的内容及格式为：

```
15
刘翔 88.0 76.0 78.0 70.0
姚明 98.0 67.0 79.0 65.0
李娜 89.0 79.0 85.0 79.0
.....
```

在上面 myclass1.txt 文件的第 1 行中的 15 是学生数。

- 2) 计算他们的平均成绩，并将计算后的信息写入文件 myclass2.txt 中。

myclass2.txt 中可能的内容及格式为：

```
15
刘翔 88.0 76.0 78.0 70.0 78.0
姚明 98.0 67.0 79.0 65.0 77.3
李娜 89.0 79.0 85.0 79.0 83.0
.....
```

- 3) 根据平均成绩对他们进行排序，排序后的结果写入文件 myclass3.txt 中。

myclass3.txt 中可能的内容及格式为：

```
15
李娜 89.0 79.0 85.0 79.0 83.0
刘翔 88.0 76.0 78.0 70.0 78.0
姚明 98.0 67.0 79.0 65.0 77.3
.....
```

- 4) 编写一个添加学生信息的程序。在该程序中要求你通过键盘输入一个名字，然后查找该同学是否已在文件中存在？若存在，则将该同学的信息显示出来；若不存在，则要求输入该同学的四门课程成绩，然后计算该同学的平均成绩，并根据平均成绩将该同学插入到合适的位置，并将添加了该同学的信息写入到文件 myclass4.txt 中。例如：

若你输入为：“刘翔”

则输出为：

该同学的信息已经录入：

刘翔 88.0 76.0 78.0 70.0 78.0

若你输入为：“周杰伦”

则输出为：

该同学的信息不存在，请录入：

.....

在程序运行后，myclass4.txt 中可能的内容及格式为：

```
16
李娜 89.0 79.0 85.0 79.0 83.0
周杰伦 88.0 78.0 84.0 78.0 82.0
刘翔 88.0 76.0 78.0 70.0 78.0
姚明 98.0 67.0 79.0 65.0 77.3
.....
```