



**哈尔滨工业大学**  
Harbin Institute of Technology

# 实验大作业

课程名称： C 语言程序设计

报告类型： 程序设计报告

实验题目：

实验类型： (综合设计型)

所在院系：

学生类别： 本科生

指导教师：

学生姓名：

班级和学号：

实验时间：      年      季学期

成绩评定：

## (一) 实验目的

熟悉一维、二维数组做函数参数，排序、查找、统计分析等常用算法，自顶向下、逐步求精的模块化设计以及增量测试方法

## (二) 实验内容和要求

### 实验内容：

对学生成绩管理系统 V6.0 的升级版

某班有最多不超过 30 人（具体人数由键盘输入）参加哈工大本专业大一本学期所有课程的考试，用结构体作函数参数编程实现对学生成绩的管理。

根据实际需求，另外增加了对学生成绩的分析报告，分析每个科目及格率，成绩分层统计；

增加了关于学分绩的计算。更加便利了老师使用本学生成绩管理系统。

### 实验要求：

采用自顶向下、逐步求精的模块化设计思想设计的一个小型数据库管理系统，也可以是一个游戏，或者其他作品。要求按系统的需求分析、系统设计（总体设计、各个模块的设计）、系统实现（各个模块的实现）、系统测试等过程组织报告内容，说明采用了什么数据结构和算法，遇到哪些问题，这些问题是如何解决的，本设计的亮点在哪里。

本系统主要实现下面的功能：

- 01: 录入 每个学生的学号、姓名和各科的考试成绩
- 02: 计算 每门课程的总分和平均分
- 03: 计算 每个学生的总分和平均分
- 04: 依据每个学生的总分从高到低排出成绩表
- 05: 依据每个学生的总分从低到高排出成绩表
- 06: 依据学号由小到大排出成绩表
- 07: 依据姓名的字典顺序排出成绩表
- 08: 依据学号查询学生姓名及其考试成绩
- 09: 依据姓名查询学生学号及其考试成绩
- 10: 输出 每个学生的学号、姓名、各科考试成绩以及每门课程的总分和平均分
- 11: 成绩数据分析（成绩分级统计等）
- 12: 将每个学生的记录信息写入文件。
- 13: 从文件读取每个学生的记录信息
- 14: 退出系统。
15. **新增：**计算每位学生的平均学分绩。



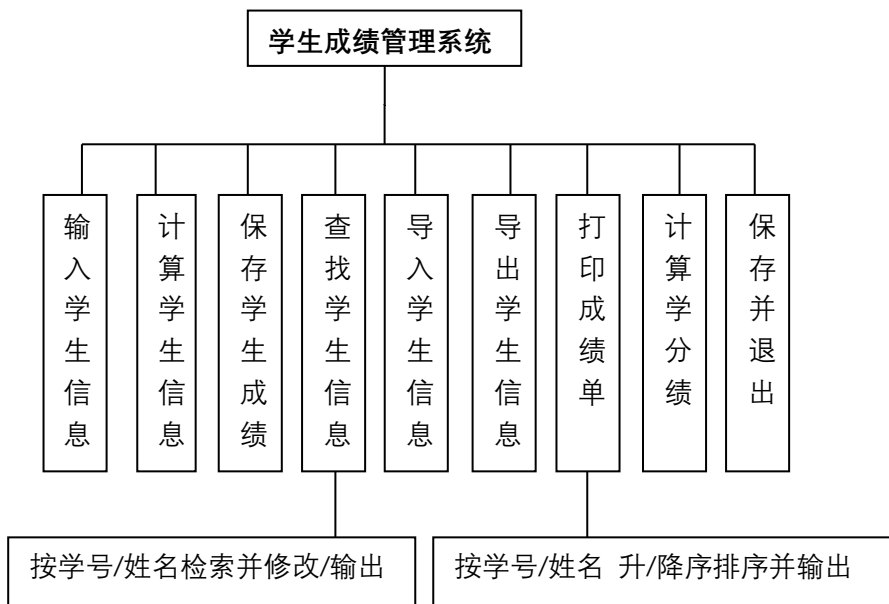
## (三) 实验环境

操作系统: Win10

编译器: Code:Blocks16.01

#### (四) 系统设计

系统功能模块结构图：



**算法设计：**（请写出程序中用到的算法，并说明所用的算法的思想）

在排序部分运用了选择法排序，其核心思想是：（按从小到大排列一组数字为例）第 1 趟，在待排序记录  $r[1] \sim r[n]$  中选出最小的记录，将它与  $r[1]$  交换；第 2 趟，在待排序记录  $r[2] \sim r[n]$  中选出最小的记录，将它与  $r[2]$  交换；以此类推，第  $i$  趟在待排序记录  $r[i] \sim r[n]$  中选出最小的记录，将它与  $r[i]$  交换，使有序序列不断增长直到全部排序完毕。

在查找学生信息部分运用了遍历算法，其核心思想是遍历每一个元素直到找到合乎条件的那一个。在数据没有进行排序以及经常变动的情况下，这种排序比二分排序要更适用。

**数据结构设计：**

（请说明采用数组、结构体还是链表存储学生个人信息，并列出所定义的数据各种信息，如：）

结构体 1-STU:

序号	字段名	字段含义	类型	长度	默认值	说明
1	num	学生学号	Char[15]	15 位	null	
2	name	学生姓名	Char [15]	15 位	null	
3	score	学生各科成绩	Int [6]	6*32 位	0	
4	sum	学生总成绩	float	32 位	0.0	
5	aver	学生平均成绩	float	32 位	0.0	
6	point	学生学分绩	float	32 位	0.0	

## 各模块（函数）的功能和接口设计：

**主函数：** 完成指令读取并按要求调用各函数。

(1) Menu**函数原型：** `int Menu(void);`

函数功能：显示菜单并获得用户键盘输入的选项

函数入口参数：

函数返回值：返回用户键盘输入的选项itemSelected

(2) AverSumofEveryStudent**函数原型：** `void AverSumofEveryStudent(STU stu[], int n, int m);`

函数功能：计算每个学生各门课程的总分和平均分

函数入口参数：结构体数组stu[], 整形变量n表示学生人数，整形变量m表示课程门数

函数返回值：无

(3) ReadScore**函数原型：** `void ReadScore(STU stu[], int n, int m);`

函数功能：输入n个学生的m门课程成绩

函数入口参数：结构体数组stu[], 整形变量n表示学生人数，整形变量m表示课程门数

函数返回值：无

(4) AverSumofEveryCourse**函数原型：** `void AverSumofEveryCourse(STU stu[], int n, int m);`

函数功能：计算每门课程的总分和平均分

函数入口参数：结构体数组stu[], 整形变量n表示学生人数，整形变量m表示课程门数

函数返回值：无

(5) Ascending**函数原型：** `int Ascending(float a, float b);`

函数功能：使数据升序排序

函数入口参数：需要比较的两个浮点数

函数返回值：返回比较结果的布尔值

(6) SortbyScore**函数原型：**

`void SortbyScore(STU stu[], int n, int m, int (*compare)(float a, float b));`

函数功能：按照选择法将数组sum的元素值排序

函数入口参数：结构体数组stu[], 整形变量n表示学生人数，整形变量m表示课程门数, 指向Ascending和Descending的函数指针\*compare

函数返回值：无

(7) Descending**函数原型：** `int Descending(float a, float b);`

函数功能：使数据降序排序

函数入口参数：需要比较的两个浮点数

函数返回值：返回比较结果的布尔值

(8) SwapFloat**函数原型：** `void SwapFloat(float *x, float *y);`

---

函数功能：交换两个浮点数内存位置

函数入口参数：需要交换的两个浮点数

函数返回值：无

(9) SwapLong函数原型：void SwapLong(long \*x, long \*y);

函数功能：交换两个长整数内存位置

函数入口参数：需要交换的两个长整型数

函数返回值：无

(10) AsSortbyNum函数原型：void AsSortbyNum(STU stu[], int n, int m);

函数功能：按选择法将字符串数组num的元素按顺序排列

函数入口参数：结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数

函数返回值：无

(11) SwapChar函数原型：void SwapChar(char x[], char y[]);

函数功能：交换两个字符串的内存位置

函数入口参数：需要交换的两个字符串

函数返回值：无

(12) SortbyName函数原型：void SortbyName(STU stu[], int n, int m);

函数功能：交换法实现字符串按字典顺序排序

函数入口参数：结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数

函数返回值：无

(13) SearchbyNum函数原型void SearchbyNum(STU stu[], int n, int m);

函数功能：按学号查找学生成绩并显示查找结果

函数入口参数：结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数

函数返回值：无

(14) SearchbyName函数原型：void SearchbyName(STU stu[], int n, int m);

函数功能：按姓名查找学生成绩并显示查找结果

函数入口参数：结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数

函数返回值：无

(15) PrintScore函数原型：void PrintScore(STU stu[], int n, int m);

函数功能：打印学生成绩

函数入口参数：学生数据数组stu[], 学生人数n, 课程数m

函数返回值：无

(16) StatisticAnalysis函数原型：void StatisticAnalysis(STU stu[], int n, int m);

函数功能：统计各分数段的学生人数及所占的百分比

函数入口参数：结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数

函数返回值：无

(17) GradePoint函数原型: `void GradePoint(STU stu[], int n, int m);`

函数功能: 计算平均学分绩

函数入口参数: 学生数据数组stu[], 学生人数n, 课程数m

函数返回值: 无

(18) ReadfromFile函数原型: `void ReadfromFile(STU stu[], int *n, int *m);`

函数功能: 从文件中读取学生的学号、姓名及成绩等信息写入到结构体数组stu中

函数入口参数: 学生数据数组stu[], 学生人数n, 课程数m

函数返回值: 无

(19) WritetoFile函数原型: `void WritetoFile(STU stu[], int n, int m);`

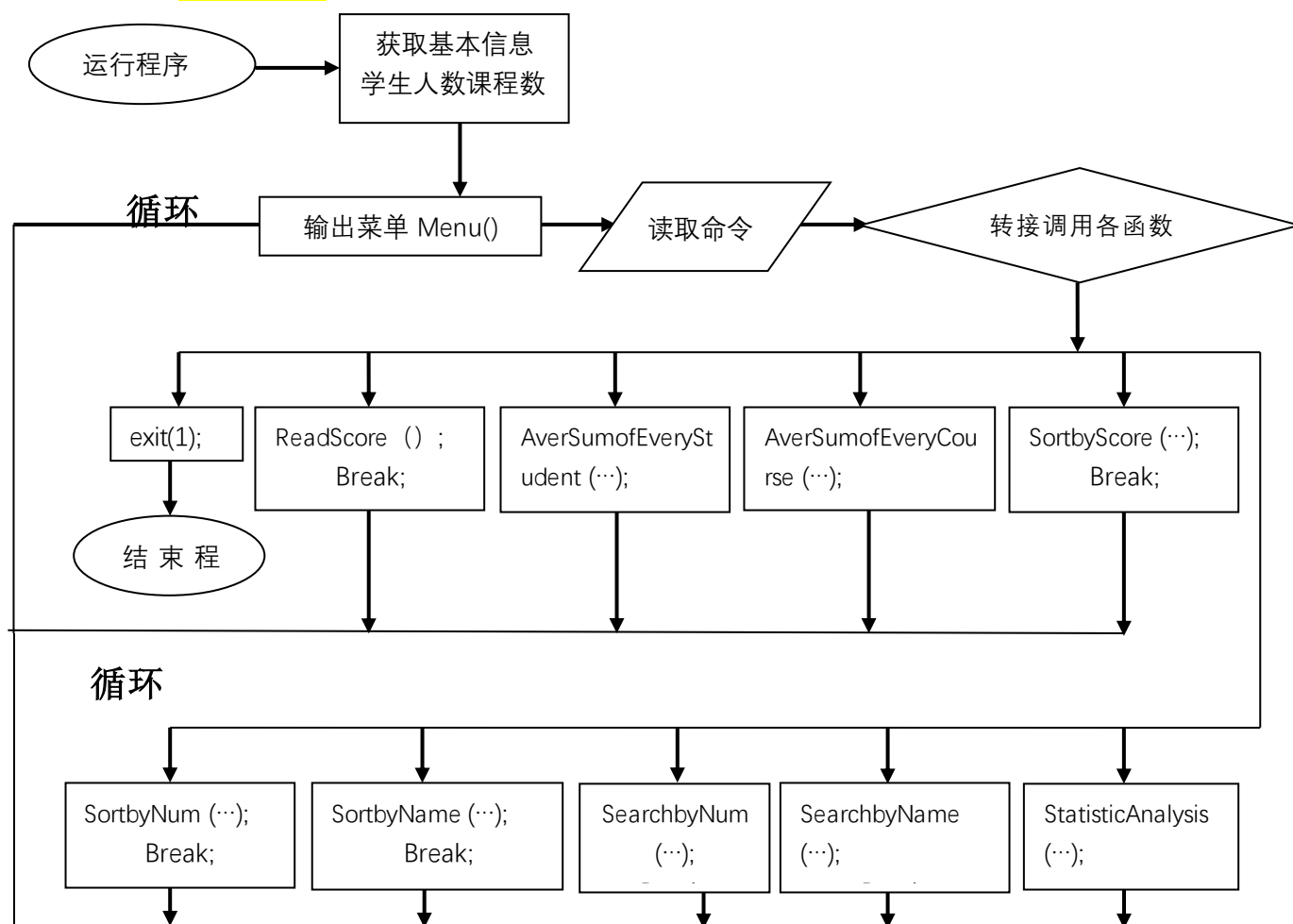
函数功能: 输出n个学生的学号、姓名及m门课程的成绩到文件student.txt中

函数入口参数: 学生数据数组stu[], 学生人数n, 课程数m

函数返回值: 无

## 系统实现

### 主要模块的程序流程图:



## 源程序的全部代码：

文件：源.c

```
001 #include <stdio.h>
002 #include <stdlib.h>
003 #include <string.h>
004 #include <windows.h>
005 #define MAX_LEN 15 //字符串最大长度
006 #define STU_NUM 50 //最多的学生人数
007 #define COURSE_NUM 8 //最多的考试科目数

008 typedef struct student
009 {
010     char num[MAX_LEN]; //每个学生的学号
011     char name[MAX_LEN]; //每个学生的姓名
012     float score[COURSE_NUM]; //每个学生COURSE_LEN门功课的成绩
013     float sum; //每个学生的总成绩
014     float aver; //每个学生的平均成绩
015     float point; //每个学生的平均学分绩
016 }STU;

017 int Menu(void);
018 void ReadScore(STU stu[], int n, int m);
019 void AverSumofEveryStudent(STU stu[], int n, int m);
020 void AverSumofEveryCourse(STU stu[], int n, int m);
021 void SortbyScore(STU stu[], int n, int m, int(*compare)(float a, float b));
022 int Ascending(float a, float b);
023 int Descending(float a, float b);
024 void SwapFloat(float *x, float *y);
025 void SwapLong(long *x, long *y);
026 void SwapChar(char x[], char y[]);
027 void AsSortbyNum(STU stu[], int n, int m);
028 void SortbyName(STU stu[], int n, int m);
029 void SearchbyNum(STU stu[], int n, int m);
030 void SearchbyName(STU stu[], int n, int m);
031 void StatisticAnalysis(STU stu[], int n, int m);
032 void GradePoint(STU stu[], int n, int m);
033 void PrintScore(STU stu[], int n, int m);
034 void WritetoFile(STU record[], int n, int m);
035 void ReadfromFile(STU record[], int *n, int *m);

036 /* 函数功能：主函数，提示用户输入各项数据并按需求调用各项函数
037 函数入口参数：无
038 函数返回值：程序正常结束返回0
```

```
039 */
040 int main()
041 {
042     system("color f1");
043     char ch;
044     int n = 0, m = 0; //学生人数为n, 课程数m
045     int ret1, ret2; //两个返回值用来判断输入是否符合要求
046     STU stu[STU_NUM];

047     printf("\n                                欢迎进入学生成绩管理系统\n\n");
048     printf("请输入学生人数（50人之内）");
049     ret1 = scanf("%d", &n);
050     while (ret1 != 1 || n > 50 || n <= 0)
051     {
052         while (getchar() != '\n');
053         printf("请重新输入学生人数（50人之内）");
054         ret1 = scanf("%d", &n);
055     }
056     printf("请输入课程数目（8门之内）");
057     ret2 = scanf("%d", &m);
058     while (ret2 != 1 || m > 8 || m <= 0)
059     {
060         while (getchar() != '\n');
061         printf("请重新输入课程数目（8门之内）");
062         ret2 = scanf("%d", &m);
063     }
064     printf("\n");

065     while (1)
066     {
067         ch = Menu(); //显示菜单，并读取用户输入
068         switch (ch)
069         {
070             case 1: ReadScore(stu, n, m);
071             break;
072             case 2: AverSumofEveryCourse(stu, n, m);
073             break;
074             case 3: AverSumofEveryStudent(stu, n, m);
075             break;
076             case 4: SortbyScore(stu, n, m, Descending);
077             printf("\n按总分从高到低排序: \n");
078             PrintScore(stu, n, m);
079             break;
```



```
080 case 5: SortbyScore(stu, n, m, Ascending);
081 printf("\n按总分从低到高排序: \n");
082 PrintScore(stu, n, m);
083 break;
084 case 6: AsSortbyNum(stu, n, m);
085 printf("\n按学号从小到大排序: \n");
086 PrintScore(stu, n, m);
087 break;
088 case 7: SortbyName(stu, n, m);
089 printf("\n按姓名的字典顺序排序: \n");
090 PrintScore(stu, n, m);
091 break;
092 case 8: SearchbyNum(stu, n, m);
093 break;
094 case 9: SearchbyName(stu, n, m);
095 break;
096 case 10: StatisticAnalysis(stu, n, m);
097 break;
098 case 11: PrintScore(stu, n, m);
099 break;
100 case 12: GradePoint(stu, n, m);
101 break;
102 case 13: WritetoFile(stu, n, m);
103 break;
104 case 14: ReadfromFile(stu, &n, &m);
105 break;
106 case 0: printf("\n                                     欢迎您下次使用!\n");
107 exit(0);
108 default: printf("输入错误! ");
109 }
110 }
111 return 0;
112 }

113 /* 函数功能: 显示菜单并获得用户键盘输入的选项
114 函数入口参数:
115 函数返回值: 返回用户键盘输入的选项itemSelected
116 */
117 int Menu(void)
118 {
119     int itemSelected;

120     printf("1. 录入每个学生的学号、姓名和各科考试成绩\n");
```

```
121 printf("2. 计算每门课程的总分和平均分\n");
122 printf("3. 计算每个学生的总分和平均分\n");
123 printf("4. 按每个学生的总分从高到低排出成绩表\n");
124 printf("5. 按每个学生的总分从低到高排出成绩表\n");
125 printf("6. 按学号由小到大排出成绩表\n");
126 printf("7. 按姓名的字典顺序排出成绩表\n");
127 printf("8. 按学号查询学生姓名及其考试成绩\n");
128 printf("9. 按姓名查询学生学号及其考试成绩\n");
129 printf("10. 成绩数据分析\n");
130 printf("11. 输出每个学生的学号、姓名、各科考试成绩以及每门课程的总分和平均分\n");
131 printf("12. 计算每位学生的平均学分绩\n");
132 printf("13. 将每个学生的记录信息写入文件\n");
133 printf("14. 从文件读取每个学生的记录信息\n");
134 printf("0. 退出系统\n");
135 printf("\n请输入您的选择:");
136 scanf("%d", &itemSelected);    //读入用户输入
137 return itemSelected;
138 }

139 /* 函数功能: 输入n个学生的m门课程成绩
140 函数入口参数: 结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数
141 函数返回值: 无
142 */
143 void ReadScore(STU stu[], int n, int m)
144 {
145     int i, j;

146     printf("请输入学生的学号、姓名和成绩\n");
147     for (i = 0; i < n; i++)
148     {
149         scanf("%s%s", stu[i].num, stu[i].name);
150         for (j = 0; j < m; j++)
151         {
152             scanf("%f", &stu[i].score[j]);
153         }
154     }
155     printf("\n");
156 }

157 /* 函数功能: 计算每个学生各门课程的总分和平均分
158 函数入口参数: 结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数
159 函数返回值: 无
160 */
161 void AverSumofEveryStudent(STU stu[], int n, int m)
```

```
162 {
163     int i, j;

164     for (i = 0; i < n; i++)
165     {
166         stu[i].sum = 0;
167         for (j = 0; j < m; j++)
168         {
169             stu[i].sum = stu[i].sum + stu[i].score[j];
170         }
171         stu[i].aver = m > 0 ? stu[i].sum / m : -1;
172         printf("学生 %d: 总成绩: %.0f, 平均分: %.0f\n", i + 1, stu[i].sum,
            stu[i].aver);
173     }
174 }

175 /* 函数功能: 计算每门课程的总分和平均分
176 函数入口参数: 结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数
177 函数返回值: 无
178 */
179 void AverSumofEveryCourse(STU stu[], int n, int m)
180 {
181     int i, j;
182     float sum[COURSE_NUM], aver[COURSE_NUM];

183     for (j = 0; j < m; j++)
184     {
185         sum[j] = 0;
186         for (i = 0; i < n; i++)
187         {
188             sum[j] = sum[j] + stu[i].score[j];
189         }
190         aver[j] = n > 0 ? sum[j] / n : -1;
191         printf("课程 %d: 总分: %.0f, 平均分: %.0f\n", j + 1, sum[j], aver[j]);
192     }
193 }

194 /* 函数功能: 按照选择法将数组sum的元素值排序
195 函数入口参数: 结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数, 指向
    Ascending和Descending的函数指针*compare
196 函数返回值: 无
197 */
198 void SortbyScore(STU stu[], int n, int m, int (*compare)(float a, float b))
199 {
```

---

```
200 int i, j, k, t;

201 for (i = 0; i < n - 1; i++)
202 {
203     k = i;
204     for (j = i + 1; j < n; j++)
205     {
206         if ((*compare)(stu[j].sum, stu[k].sum))
207             k = j;
208     }
209     if (k != i)
210     {
211         for (t = 0; t < m; t++)    //交换m门课程的成绩
212         {
213             SwapFloat(&stu[k].score[t], &stu[i].score[t]);
214         }
215         SwapFloat(&stu[k].sum, &stu[i].sum);    //交换总分
216         SwapFloat(&stu[k].aver, &stu[i].aver);    //交换平均分
217         SwapChar(stu[k].num, stu[i].num);    //交换学号
218         SwapChar(stu[k].name, stu[i].name);    //交换姓名
219     }
220 }
221 }

222 /* 函数功能：使数据升序排序
223 函数入口参数：需要比较的两个浮点数
224 函数返回值：返回比较结果的布尔值
225 */
226 int Ascending(float a, float b)
227 {
228     return a < b;    //这样比较决定了按升序排序，如果a<b，则交换
229 }

230 /* 函数功能：使数据降序排序
231 函数入口参数：需要比较的两个浮点数
232 函数返回值：返回比较结果的布尔值
233 */
234 int Descending(float a, float b)
235 {
236     return a > b;    //这样比较决定了按降序排序，如果a>b，则交换
237 }

238 /* 函数功能：交换两个浮点数内存位置
239 函数入口参数：需要交换的两个浮点数
```

---

```
240 函数返回值: 无
241 */
242 void SwapFloat(float *x, float *y)
243 {
244     float temp;
245     temp = *x;
246     *x = *y;
247     *y = temp;
248 }

249 /* 函数功能: 交换两个长整形数内存位置
250 函数入口参数: 需要交换的两个长整型数
251 函数返回值: 无
252 */
253 void SwapLong(long *x, long *y)
254 {
255     long temp;
256     temp = *x;
257     *x = *y;
258     *y = temp;
259 }

260 /* 函数功能: 交换两个字符串的内存位置
261 函数入口参数: 需要交换的两个字符串
262 函数返回值: 无
263 */
264 void SwapChar(char x[], char y[])
265 {
266     char temp[MAX_LEN];
267     strcpy(temp, x);
268     strcpy(x, y);
269     strcpy(y, temp);
270 }

271 /* 函数功能: 按选择法将字符数组num的元素按顺序排列
272 函数入口参数: 结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数
273 函数返回值: 无
274 */
275 void AsSortbyNum(STU stu[], int n, int m)
276 {
277     int i, j, t;

278     for (i = 0; i < n - 1; i++)
279     {
```

---

```
280 for (j = i + 1; j < n; j++)
281 {
282     if (strcmp(stu[j].num, stu[i].num) < 0)
283     {
284         for (t = 0; t < m; t++)    //交换m门课程的成绩
285         {
286             SwapFloat(&stu[i].score[t], &stu[j].score[t]);
287         }
288         SwapFloat(&stu[i].sum, &stu[j].sum);    //交换总分
289         SwapFloat(&stu[i].aver, &stu[j].aver);    //交换平均分
290         SwapChar(stu[i].num, stu[j].num);    //交换学号
291         SwapChar(stu[i].name, stu[j].name);    //交换姓名
292     }
293 }
294 }
295 }

296 /*  函数功能: 交换法实现字符串按字典顺序排序
297  函数入口参数: 结构体数组stu[], 整形变量n表示学生人数, 整形变量m表示课程门数
298  函数返回值: 无
299  */
300 void SortbyName(STU stu[], int n, int m)
301 {
302     int i, j, t;

303     for (i = 0; i < n - 1; i++)
304     {
305         for (j = i + 1; j < n; j++)
306         {
307             if (strcmp(stu[j].name, stu[i].name) < 0)
308             {
309                 for (t = 0; t < m; t++)    //交换m门课程的成绩
310                 {
311                     SwapFloat(&stu[i].score[t], &stu[j].score[t]);
312                 }
313                 SwapFloat(&stu[i].sum, &stu[j].sum);    //交换总分
314                 SwapFloat(&stu[i].aver, &stu[j].aver);    //交换平均分
315                 SwapChar(stu[i].num, stu[j].num);    //交换学号
316                 SwapChar(stu[i].name, stu[j].name);    //交换姓名
317             }
318         }
319     }
320 }
```

---

```
321 /*  函数功能：按学号查找学生成绩并显示查找结果
322  函数入口参数：结构体数组stu[], 整形变量n表示学生人数，整形变量m表示课程门数
323  函数返回值：无
324 */
325 void SearchbyNum(STU stu[], int n, int m)
326 {
327     char number[MAX_LEN];
328     int i, j;

329     printf("请输入想要查找的学号：");
330     scanf("%s", number);    //输入想要查找的学号
331     for (i = 0; i < n; i++)
332     {
333         if (strcmp(stu[i].num, number) == 0)    //查到
334         {
335             printf("%s\t%s\t", stu[i].num, stu[i].name);    //输出该学号以及对应姓名
336             for (j = 0; j < m; j++)
337             {
338                 printf("%.0f\t", stu[i].score[j]);    //输出该学号对应的m门课程的成绩
339             }
340             printf("%.0f\t%.0f\n", stu[i].sum, stu[i].aver);    //输出该学号对应的总分和平均分
341             return;
342         }
343     }
344     printf("未发现!\n");
345 }

346 /*  函数功能：按姓名查找学生成绩并显示查找结果
347  函数入口参数：结构体数组stu[], 整形变量n表示学生人数，整形变量m表示课程门数
348  函数返回值：无
349 */
350 void SearchbyName(STU stu[], int n, int m)
351 {
352     char x[MAX_LEN];
353     int i, j;

354     printf("请输入想要查找的姓名：");
355     scanf("%s", x);    //输入想要查找的姓名
356     for (i = 0; i < n; i++)
357     {
358         if (strcmp(stu[i].name, x) == 0)    //查到
359         {
360             printf("%s\t%s\t", stu[i].num, stu[i].name);    //输出该姓名以及对应学号
361             for (j = 0; j < m; j++)
```

```
362 {
363     printf("%.0f\t", stu[i].score[j]);    //输出该姓名对应的m门课程的成绩
364 }
365 printf("%.0f\t%.0f\n", stu[i].sum, stu[i].aver);    //输出该姓名对应的总分和平均分
366 return;
367 }
368 }
369 printf("未发现!\n");
370 }

371 /*  函数功能：统计各分数段的学生人数及所占的百分比
372  函数入口参数：结构体数组stu[], 整形变量n表示学生人数，整形变量m表示课程门数
373  函数返回值：无
374 */
375 void StatisticAnalysis(STU stu[], int n, int m)
376 {
377     int i, j, t[6];

378     for (j = 0; j < m; j++)
379     {
380         printf("对于课程%d:\n", j + 1);
381         memset(t, 0, sizeof(t));    //将数组t的全部元素初始化为零
382         for (i = 0; i < n; i++)
383         {
384             if (stu[i].score[j] >= 0 && stu[i].score[j] < 60) t[0]++;
385             else if (stu[i].score[j] < 70) t[1]++;
386             else if (stu[i].score[j] < 80) t[2]++;
387             else if (stu[i].score[j] < 90) t[3]++;
388             else if (stu[i].score[j] < 100) t[4]++;
389             else if (stu[i].score[j] == 100) t[5]++;
390         }
391         for (i = 0; i <= 5; i++)
392         {
393             if (i == 0)
394                 printf("<60\t%d\t%.2f%\n", t[i], (float)t[i] / n * 100);    //不及格人数及百分比
395             else if (i == 5)
396                 printf("%d\t%d\t%.2f%\n", (i + 5) * 10, t[i], (float)t[i] / n * 100);    //满分人
                 数及百分比
397             else printf("%d-%d\t%d\t%.2f%\n", (i + 5) * 10, (i + 5) * 10 + 9, t[i],
                 (float)t[i] / n * 100);    //其他分数段人数及百分比
398         }
399     }
400 }
```



---

```
401 /* 函数功能: 打印学生成绩
402 函数入口参数: 学生数据数组stu[], 学生人数n, 课程数m
403 函数返回值: 无
404 */
405 void PrintScore(STU stu[], int n, int m)
406 {
407     int i, j;

408     for (i = 0; i < n; i++)
409     {
410         printf("%s\t%s\t", stu[i].num, stu[i].name);    //打印学号和姓名
411         for (j = 0; j < m; j++)
412         {
413             printf("%.0f\t", stu[i].score[j]);    //打印m门课程的成绩
414         }
415         printf("%.0f\t%.0f\n", stu[i].sum, stu[i].aver);    //打印总分和平均分
416     }
417 }

418 /* 函数功能: 计算平均学分绩
419 函数入口参数: 学生数据数组stu[], 学生人数n, 课程数m
420 函数返回值: 无
421 */
422 void GradePoint(STU stu[], int n, int m)
423 {
424     float grade[COURSE_NUM];
425     int i, j, sum = 0;

426     printf("请输入每门课程的学分: ");
427     for (i = 0; i < m; i++)
428     {
429         scanf("%f", &grade[i]);
430         sum += grade[i];
431     }
432     for (i = 0; i < n; i++)
433     {
434         stu[i].point = 0;
435         for (j = 0; j < m; j++)
436         {
437             stu[i].point += (stu[i].score[j])*grade[j];
438         }
439         printf("学生 %d: 平均学分绩: %.0f\n", i + 1, (stu[i].point) / sum);
440     }
```

---

```
441 }
```

442 /\* 函数功能：输出n个学生的学号、姓名及m门课程的成绩到文件student.txt中  
443 函数入口参数：学生数据数组stu[]，学生人数n，课程数m  
444 函数返回值：无  
445 \*/

```
446 void WritetoFile(STU stu[], int n, int m)
447 {
448     FILE *fp;
449     int i, j;
450     if ((fp = fopen("student.txt", "w")) == NULL)
451     {
452         printf("打开文件失败！\n");
453         exit(0);
454     }
455     fprintf(fp, "%d\t%d\n", n, m); //将学生人数和课程门数写入文件
456     for (i = 0; i < n; i++)
457     {
458         fprintf(fp, "%10s%10s", stu[i].num, stu[i].name);
459         for (j = 0; j < m; j++)
460         {
461             fprintf(fp, "%10.0f", stu[i].score[j]);
462         }
463         fprintf(fp, "%10.0f%10.0f\n", stu[i].sum, stu[i].aver);
464     }
465     fclose(fp);
466 }
```

467 /\* 函数功能：从文件中读取学生的学号、姓名及成绩等信息写入到结构体数组stu中  
468 函数入口参数：学生数据数组stu[]，学生人数n，课程数m  
469 函数返回值：无  
470 \*/

```
471 void ReadfromFile(STU stu[], int *n, int *m)
472 {
473     FILE *fp;
474     int i, j;
475     if ((fp = fopen("student.txt", "r")) == NULL)
476     {
477         printf("打开文件失败！\n");
478         exit(0);
479     }
480     fscanf(fp, "%d\t%d", n, m); //从文件中读出学生人数和课程门数
```

```
481 for (i = 0; i < *n; i++) //学生人数保存在n指向的存储单元
482 {
483     fscanf(fp, "%10s", stu[i].num);
484     fscanf(fp, "%10s", stu[i].name);
485     for (j = 0; j < *m; j++) //课程门数保存在n指向的存储单元
486     {
487         fscanf(fp, "%10f", &stu[i].score[j]);
488     }
489     fscanf(fp, "%10f%10f", &stu[i].sum, &stu[i].aver);
490 }
491 fclose(fp);
492 }
```

## (五) 实验步骤及调试过程

### 遇到的问题有：

1、输入非法数据时程序崩溃退出；

解决方法：对输入数据进行合法性检查和防溢出检查，当输入错误数据时，提示输入有误，并要求用户重新输入；

2、出现少数逻辑错误使得排序结果不符合要求；

解决方法：重新检查代码，谨慎选择排序方法，逐条运行代码检查错误步骤并修改；

3、输出学生信息时，输出结果不整齐，不美观；

解决方法：调整输出格式，使用'\t'使输出网格对齐，再次运行程序观察是否合适美观。

## (六) 系统测试结果

正常打开：

欢迎进入学生成绩管理系统

获取本学期考试基本信息（科目数，学生人数）：

假设为 2

请输入学生人数（50人之内） 2  
请输入课程数目（8门之内） 2

菜单界面：

```

*****
1. 录入每个学生的学号、姓名和各科考试成绩
2. 计算每门课程的总分和平均分
3. 计算每个学生的总分和平均分
4. 按每个学生的总分从高到低排出成绩表
5. 按每个学生的总分从低到高排出成绩表
6. 按学号由小到大排出成绩表
7. 按姓名的字典顺序排出成绩表
8. 按学号查询学生姓名及其考试成绩
9. 按姓名查询学生学号及其考试成绩
10. 成绩数据分析
11. 输出每个学生的学号、姓名、各科考试成绩以及每门课程的总分和平均分
12. 计算每位学生的平均学分绩
13. 将每个学生的记录信息写入文件
14. 从文件读取每个学生的记录信息
0. 退出系统
*****

```

录入学生信息:

```

请输入您的选择:1
请输入学生的学号、姓名和成绩
1 aaa 99 98
2 bbb 97 95

```

计算每门课程的总分和平均分:

```

请输入您的选择:2
课程 1: 总分 : 196, 平均分 : 98
课程 2: 总分 : 193, 平均分 : 97

```

计算每个学生的总分和平均分:

```

请输入您的选择:3
学生 1: 总成绩 : 197, 平均分 : 99
学生 2: 总成绩 : 192, 平均分 : 96

```

按每个学生的总分从高到低排出成绩表:

请输入您的选择:4

按总分从高到低排序:

1	aaa	99	98	197	99
2	bbb	97	95	192	96

按每个学生的总分从低到高排出成绩表:

请输入您的选择:5

按总分从低到高排序:

2	bbb	97	95	192	96
1	aaa	99	98	197	99

按学号由小到大排出成绩表:

按学号从小到大排序:

1	aaa	99	98	197	99
2	bbb	97	95	192	96

按姓名的字典顺序排出成绩表:

按姓名的字典顺序排序:

1	aaa	99	98	197	99
2	bbb	97	95	192	96

按学号查询学生姓名及其考试成绩:

请输入您的选择:8

请输入想要查找的学号: 1

1	aaa	99	98	197	99
---	-----	----	----	-----	----

按姓名查询学生学号及其考试成绩:

请输入您的选择:9

请输入想要查找的姓名: aaa

1	aaa	99	98	197	99
---	-----	----	----	-----	----

成绩数据分析:

请输入您的选择:10

对于课程1:

<60	0	0.00%
60-69	0	0.00%
70-79	0	0.00%
80-89	0	0.00%
90-99	2	100.00%
100	0	0.00%

对于课程2:

<60	0	0.00%
60-69	0	0.00%
70-79	0	0.00%
80-89	0	0.00%
90-99	2	100.00%
100	0	0.00%

输出每个学生的学号、姓名、各科考试成绩以及每门课程的总分和平均分:

请输入您的选择:11

1	aaa	99	98	197	99
2	bbb	97	95	192	96

计算每位学生的平均学分绩:

请输入您的选择:12

请输入每门课程的学分: 3 4

学生 1: 平均学分绩: 98

学生 2: 平均学分绩: 96

将每个学生的记录信息写入文件:

请输入您的选择:13

保存文件成功

从文件读取每个学生的记录信息:

请输入您的选择:14

读取文件成功

退出程序:

请输入您的选择: 0

按任意键结束程序……

## (七) 分析总结

**我对题目的理解是：**开发管理系统，是对熟练选择使用合适的数据结构，和程序性思维，对程序整体框架、流程的掌控能力，以及语法正确性、函数封装、程序调试修改等基础综合能力的考察。

这要求我们不仅要追求程序运行的正确性，更要使代码风格便于阅读理解，以助于团队合作中他人进行更改和升级。

**我的思路和我的代码的特点是：**我阅读了用户需求，充分理解用户的需要，在草稿纸上模拟程序运行的流程，然后设想好了本程序的框架，并依据框架分块完成各项函数功能，并在主函数中转接调用。

得益于充分的思考构想，我的代码简洁，高利用率，充分合并同类功能，简短而有力，同时拥有美观易读的代码风格，合理缩进、注释，简单易懂的变量名，符合规范的函数名，注释适当，充分而不繁琐。整体上结构紧凑，使得 Debug 时能够迅速分析错因，找到错语句并且易于修改。

**亮点与缺点（可改进的地方）：**亮增加了学分绩和成绩分级分析的功能，更加具有实用性。代码风格严谨有序，简洁易懂，便于后期的维护和升级。

可改进的地方是，还没有进行充分的实践测试，提供的帮助信息、失败报错的功能还不够完善，有待补充和改进。

## 参考文献

- [1] 苏小红, 王宇颖, 孙志岗.C 语言程序设计 (第 3 版) [M].北京: 高等教育出版社.2015
- [2] 苏小红, 王宇颖, 孙志岗.C 语言程序设计学习指导 (第 3 版) [M].北京: 高等教育出版社.2015