



HARBIN INSTITUTE OF TECHNOLOGY

哈尔滨工业大学

实验大作业

实验项目名称： 学生成绩管理系统 V6.0 成绩评定_____

实验项目类型： 综合设计型 实验时间： _____

学生姓名： _____ 班级： _____ 学号： _____

学院： _____ 专业： _____ 任课教师： _____

(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思想以及增量测试方法。

(二) 实验内容和要求

1.实验内容

学生成绩管理系统 V6.0。某班有最多不超过 30 人（具体人数由键盘输入）参加期末考试，最多不超过 6 门（具体门数由键盘输入）。学生成绩管理系统是一个非常实用的程序，如果能够把用户输入的数据存盘，下次运行时读出，就更有用了。参考学生成绩管理系统 V5.0，增加文件读写的功能，定义结构体类型，用结构体数组作函数参数，编程实现对学生成绩的管理。

2.实验要求

采用自顶向下、逐步求精的模块化设计思想设计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，本设计的亮点和难点在哪里，实验结果如何，有哪些收获和学习体会。

3.系统功能需求分析

本系统主要实现下面的功能 V6.0:

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. 按每个学生的总分由高到低排出名次表；
5. 按每个学生的总分由低到高排出名次表；
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. 按学号查询学生排名及其考试成绩；
9. 按姓名查询学生排名及其考试成绩；
10. 按优秀（90~100）、良好（80~89）、中等（70~79）、及格（60~69）、不及格（0~59）5 个类别，对每门课程分别统计每个类别的人数以及所占的百分比；
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. 将每个学生的纪录信息写入文件。
13. 从文件中读出每个学生的纪录信息并显示

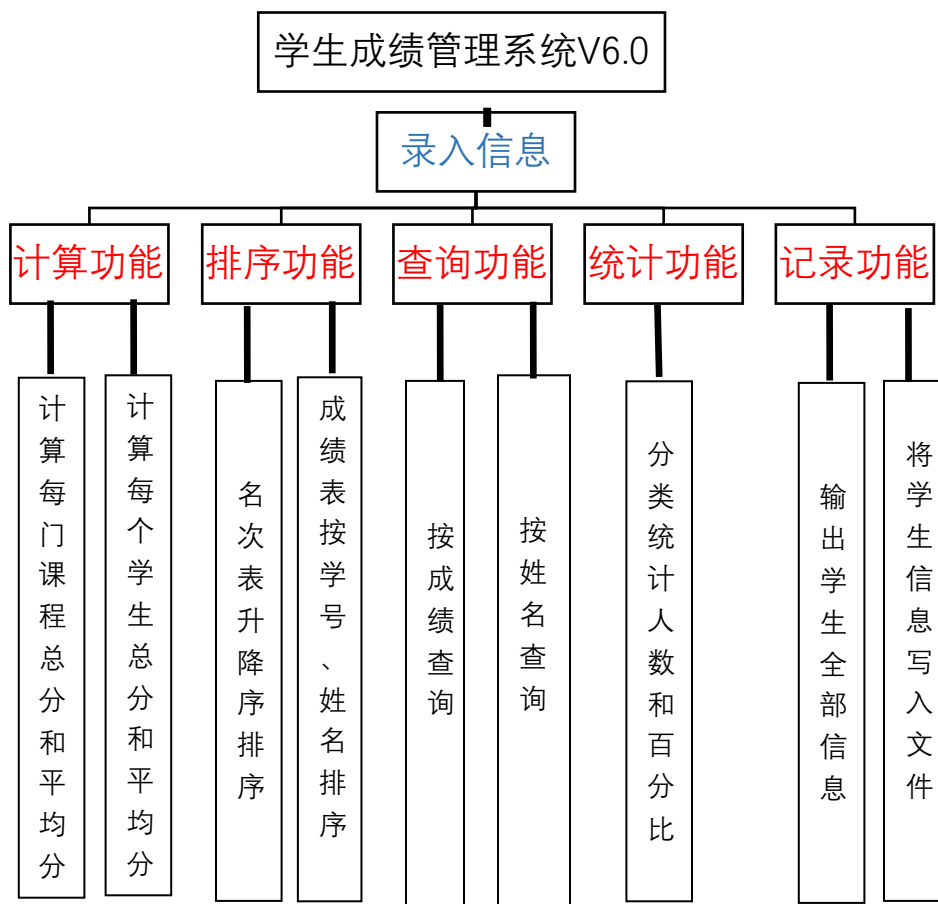
4.实验环境

(三) Win7、Codeblocks16.01

(四) 系统设计

1. 系统功能模块划分

2.



2. 算法设计

选择排序法（逐个查找），交换法。

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。

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。

数据结构设计

序号	字段名	字段含义	类型	长度
1	Stu num	学生学号	Long	10
2	Stu name	姓名	Char	10
3	Stu score	每门课成绩	float	6
4	Stu sum	每个学生总成绩	float	无
5	Stu aver	每个学生平均成绩	float	无

各模块（函数）的功能和接口设计：

（1）主函数：完成读入学生数，科目数功能

参数：

Stu_ID[]:存放学生的学号信息形参数组

Score[]: 存放学生成绩形参数组

n: 学生人数的形参

m: 学生科目的形参

（2）定义函数

序号	函数名	函数功能	函数参数	函数返回值
1	Menu()	显示菜单并获得用户键盘输入的选项	void	int
2	ReadScore()	输入 n 个学生的 m 门课程成绩	STUstu[], int n, int m	void
3	AverSumofEveryStudent	计算每个学生各门成绩的总分和平均分	STUstu[], int n, int m	void
4	AverSumofEveryCourse	计算每门课程的总分和平均分	STUstu[], int n, int m	void
5	SortbyScore	按选择法将数组 sum 的元素值排序	STU stu[], int n, Intm, int (*compare) (float a, float b	void
6	Ascending	使数据按升序排序	float a, float b	int
7	Descending	使数据按降序排序	float a, float b	int
8	SwapFloat	交换两个单精度浮点型数据	float *x, float *y	void
9	SwapLong	交换两个长整性数据	long *x, long *y	void

10	SwapChar	交换两个字符串	char x[], char y[]	void
11	AsSortbyNum	按选择法将数组 num 的元素值按从低到高排序	STU stu[], int n, int m	void
12	SortbyName	交换法实现字符串按字典顺序排序	STU stu[], int n, int m	void
13	SearchbyNum	按学号查找学生成绩并显示查找结果	STU stu[], int n, int m	void
14	SearchbyName	按姓名查找学生成绩并显示查找结果	STU stu[], int n, int m	void
15	StatisticAnalysis	统计各分数段的学生人数及所占的百分比	STU stu[], int n, int m	void
16	PrintScore	打印学生成绩	STUstu[], int n, int m	void
17	WriteToFile	将信息写入文件	STU stu[], int n, int m	void
18	ReadFromFile	从文件中读出信息	STU stu[], int *n, int *m	void

(五) 系统实现

1.系统的总流程图:

菜单驱动的系统的主流程一般为:

显示菜单——>输入用户的选择——>根据用户的输入选择不同的操作, 可以采用如下 case 语句形式的简单画法代替多分支选择结构的画法

显示菜单——>输入用户的选择——>

Case1 录入每个学生的学号、姓名和各科考试成绩;

Case2 计算每门课程的总分和平均分;

Case3 计算每个学生的总分和平均分;

Case4 按每个学生的总分由高到低排出名次表;

Case5 按每个学生的总分由低到高排出名次表;

-
- Case6** 按学号由小到大排出成绩表;
- Case7** 按姓名的字典顺序排出成绩表;
- Case8** 按学号查询学生排名及其考试成绩;
- Case9** 按姓名查询学生排名及其考试成绩;
- Case10** 按优秀(90~100)、良好(80~89)、中等(70~79)、及格(60~69)、不及格(0~59) 5个类别,对每门课程分别统计每个类别的人数以及所占的百分比;
- Case11** 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。
- Case12** 将每个学生的纪录信息写入文件。
- Case13** 从文件中读出每个学生的纪录信息并显示

3. 程序的全部源代码:

文件: 源.c

```
001 #include <stdio.h>
002 #include <stdlib.h>
003 #define MAX_LEN 10      /*字符串最大长度*/
004 #define STU_NUM 30      /*最多学生数*/
005 #define COURSE_NUM 6    /*最多科目数*/

006 typedef struct student
007 {
008     long num;            /*学号*/
009     char name[MAX_LEN];  /*姓名*/
010     float score[COURSE_NUM]; /*每门课成绩*/
011     float sum;           /*每个学生总成绩*/
012     float aver;         /*每个学生平均成绩*/
013 }STU;

014 int Init(STU stu[], int *n, int *m);
015 int Menu(void);
016 int Ascending(float a, float b);
017 int Descending(float a, float b);
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 void SwapFloat(float *x, float *y);
023 void SwapLong(long *x, long *y);
024 void SwapChar(char x[], char y[]);
025 void AsSortbyNum(STU stu[], int n, int m);
026 void SortbyName(STU stu[], int n, int m);
027 void SearchbyNum(STU stu[], int n, int m);
028 void SearchbyName(STU stu[], int n, int m);
029 void StatisticAnalysis(STU stu[], int n, int m);
030 void PrintScore(STU stu[], int n, int m);
031 void WriteToFile(STU stu[], int n, int m);
032 void ReadFromFile(STU stu[], int *n, int *m);

033 /*功能：主函数，按用户需求调用各个功能函数
034 参数：无
035 返回值：程序正常结束，返回0
036 */
037 int main(void)
038 {
039     char ch;
040     char k;
041     int n=0, m=0;      /*学生人数为n, 课程门数为m */
042     STU stu[STU_NUM];
043     if(Init(stu, &n, &m)) ;
044     else{
045         printf("Input student number(n<=%d)", STU_NUM);
046         scanf("%d", &n);
047         printf("Input course number(m<=%d)", COURSE_NUM);
048         scanf("%d", &m);
049     }
050     do
051     {
052         ch = Menu();    /*显示菜单，并读取用户输入*/
053         switch (ch)
054         {
055             case 1: ReadScore(stu, n, m);
056                 printf("\nPress Enter to continue...");
057                 getchar();
058                 getchar();
059                 break;
060             case 2: AverSumofEveryCourse(stu, n, m);
061                 printf("\nPress Enter to continue...");
062                 getchar();
063                 getchar();
```

```
064 break;
065 case 3:AverSumofEveryStudent(stu,n,m);
066 printf("\nPress Enter to continue...");
067 getchar();
068 getchar();
069 break;
070 case 4:SortbyScore(stu,n,m,Descending);
071 printf("\nSort in descending order by score:\n");
072 PrintScore(stu,n,m);
073 printf("\nPress Enter to continue...");
074 getchar();
075 getchar();
076 break;
077 case 5:SortbyScore(stu,n,m,Ascending);
078 printf("\nSort in ascending order by score:\n");
079 PrintScore(stu,n,m);
080 printf("\nPress Enter to continue...");
081 getchar();
082 getchar();
083 break;
084 case 6:AsSortbyNum(stu,n,m);
085 printf("\nSort in ascending order by number:\n");
086 PrintScore(stu,n,m);
087 printf("\nPress Enter to continue...");
088 getchar();
089 getchar();
090 break;
091 case 7:SortbyName(stu,n,m);
092 printf("\nSort in ascending order by name:\n");
093 PrintScore(stu,n,m);
094 printf("\nPress Enter to continue...");
095 getchar();
096 getchar();
097 break;
098 case 8:SearchbyNum(stu,n,m);
099 printf("\nPress Enter to continue...");
100 scanf("%c",&k);
101 break;
102 case 9:SearchbyName(stu,n,m);
103 printf("\nPress Enter to continue...");
104 getchar();
105 getchar();
106 break;
107 case 10:StatisticAnalysis(stu,n,m);
```

```
108 printf("\nPress Enter to continue...");
109 getchar();
110 getchar();
111 break;
112 case 11:PrintScore(stu, n, m);
113 break;
114 case 12:
115 WriteToFile(stu, n, m);
116 printf("\nPress Enter to continue...");
117 getchar();
118 getchar();
119 break;
120 case 13:
121 ReadFromFile(stu, &n, &m);
122 printf("\nPress Enter to continue...");
123 getchar();
124 getchar();
125 break;
126 case 0:printf("End of program!");
127 exit(0);
128 default:printf("Input error!");
129 }
130 }while(1);
131 return 0;
132 }

133 /*功能：显示菜单并获得用户键盘输入的选项
134 参数：无
135 返回值：用户的键入值
136 */
137 int Menu(void)
138 {
139 int itemSelected;
140 printf("Management For Students' Score\n");
141 printf("(YOU MUST CHOOSE 1 AND 3!!!) \n");
142 printf("1.Input record\n");
143 printf("2.Caculate total and average score of every course\n");
144 printf("3.Caculate total and average score of every student\n");
145 printf("4.Sort in descending order by score\n");
146 printf("5.Sort in ascending order by score\n");
147 printf("6.Sort in ascending order by number\n");
148 printf("7.Sort in dictionary order by name\n");
149 printf("8.Search by number\n");
150 printf("9.Search by name\n");
```

```
151 printf("10. Statistic analysis\n");
152 printf("11. List record\n");
153 printf("12. Write to a file\n");
154 printf("13. Read from a file\n");
155 printf("0. Exit\n");
156 printf("Please Input your choice:\n");
157 scanf("%d", &itemSelected);      /*读入用户输入*/
158 return itemSelected;
159 }
```

```
160 /*功能：输入n个学生的m门课程成绩
161 参数：学生信息结构体数组stu，人数n，课程数m
162 返回值：无
163 */
164 void ReadScore (STU stu[], int n, int m)
165 {
166     int i, j;
167     printf("Input student's ID, name and score:\n");
168     for(i=0; i<n; i++)
169     {
170         scanf("%ld%s", &stu[i].num, stu[i].name);
171         for(j=0; j<m; j++)
172         {
173             scanf("%f", &stu[i].score[j]);
174         }
175     }
176 }
```

```
177 /*功能：计算每个学生各门成绩的总分和平均分
178 参数：学生信息结构体数组stu，人数n，课程数m
179 返回值：无
180 */
181 void AverSumofEveryStudent (STU stu[], int n, int m)
182 {
183     int i, j;
184     for(i=0; i<n; i++)
185     {
186         stu[i].sum=0;
187         for(j=0; j<m; j++)
188         {
189             stu[i].sum=stu[i].sum+stu[i].score[j];
190         }
191         stu[i].aver=m>0?stu[i].sum/m:-1;
```

```
192 printf("student %d: sum = %.0f, aver = %.0f\n", i+1, stu[i].sum, stu[i].aver);
193 }
194 }

195 /*功能: 计算每门课程的总分和平均分
196 参数: 学生信息结构体数组stu, 人数n, 课程数m
197 返回值:
198 */
199 void AverSumofEveryCourse(STU stu[], int n, int m)
200 {
201     int i, j;
202     float sum[COURSE_NUM], aver[COURSE_NUM];
203     for(j=0; j<m; j++)
204     {
205         sum[j]=0;
206         for(i=0; i<n; i++)
207         {
208             sum[j]+=stu[i].score[j];
209         }
210         aver[j]=n>0?sum[j]/n:-1;
211     }
212     printf("course %d: sum = %.0f, aver = %.0f\n", j+1, sum[j], aver[j]);
213 }

214 /*功能: 按选择法将数组sum的元素值排序
215 参数: 学生信息结构体数组stu, 人数n, 课程数m, 比较方法函数指针
216 返回值: 无
217 */
218 void SortbyScore(STU stu[], int n, int m, int(*compare)(float a, float b))
219 {
220     int i, j, k, t;
221     for(i=0; i<n-1; i++)
222     {
223         k=i;
224         for(j=i+1; j<n; j++)
225         {
226             if((*compare)(stu[j].sum, stu[k].sum)) k=j;
227         }
228         if(k!=i)
229         {
230             for(t=0; t<m; t++) /*交换m门课程成绩*/
231             {
232                 SwapFloat(&stu[k].score[t], &stu[i].score[t]);
233             }
234         }
235     }
236 }
```

```
234 SwapFloat(&stu[k].sum,&stu[i].sum); /*交换总分*/
235 SwapFloat(&stu[k].aver,&stu[i].aver); /*交换平均分*/
236 SwapLong(&stu[k].num,&stu[i].num); /*交换学分*/
237 SwapChar(stu[k].name,stu[i].name); /*交换姓名*/
238 }
239 }
240 }
```

```
241 /*功能：使数据按升序排序
242 参数：待比较的浮点数a, b
243 返回值：比较结果的布尔值
244 */
245 int Ascending(float a, float b)
246 {
247     return a<b;
248 }
```

```
249 /*功能：使数据按降序排序
250 参数：待比较的浮点数a, b
251 返回值：比较结果的布尔值
252 */
253 int Descending(float a, float b)
254 {
255     return a>b;
256 }
```

```
257 /*功能：交换两个单精度浮点型数据
258 参数：待交换的浮点数x, y
259 返回值：无
260 */
261 void SwapFloat(float *x, float *y)
262 {
263     float t;
264     t=*x;
265     *x=*y;
266     *y=t;
267 }
```

```
268 /*功能：交换两个长整型数据
269 参数：待交换的长整型数x, y
270 返回值：无
271 */
```

```
272 void SwapLong(long *x, long *y)
273 {
274     long t;
275     t=*x;
276     *x=*y;
277     *y=t;
278 }

279 /*功能：交换两个字符串
280 参数：待交换的字符串x, y
281 返回值：无
282 */
283 void SwapChar(char x[], char y[])
284 {
285     char t[MAX_LEN];
286     strcpy(t, x);
287     strcpy(x, y);
288     strcpy(y, t);
289 }

290 /*功能：按选择法将数组num的元素值按从低到高排序
291 参数：学生信息结构体数组stu, 人数n, 课程数m
292 返回值：无
293 */
294 void AsSortbyNum(STU stu[], int n, int m)
295 {
296     int i, j, k, t;
297     for(i=0; i<n-1; i++)
298     {
299         k=i;
300         for(j=i+1; j<n; j++)
301         {
302             if(stu[j].num<stu[k].num) k=j;
303         }
304         if(k!=i)
305         {
306             for(t=0; t<m; t++)    /*交换m门课程成绩*/
307             {
308                 SwapFloat(&stu[k].score[t], &stu[i].score[t]);
309             }
310             SwapFloat(&stu[k].sum, &stu[i].sum);    /*交换总分*/
311             SwapFloat(&stu[k].aver, &stu[i].aver); /*交换平均分*/
312             SwapLong(&stu[k].num, &stu[i].num);    /*交换学分*/
313             SwapChar(stu[k].name, stu[i].name);    /*交换姓名*/
```

```
314 }
315 }
316 }

317 /*功能：交换法实现字符串按字典顺序排序
318 参数：学生信息结构体数组stu，人数n，课程数m
319 返回值：无
320 */
321 void SortbyName(STU stu[], int n, int m)
322 {
323     int i, j, t;
324     for(i=0; i<n-1; i++)
325     {
326         t=i;
327         for(j=i+1; j<n; j++)
328         {
329             if(strcmp(stu[j].name, stu[i].name)<0)
330             {
331                 for(t=0; t<m; t++)    /*交换m门课程成绩*/
332                 {
333                     SwapFloat(&stu[i].score[t], &stu[j].score[t]);
334                 }
335                 SwapFloat(&stu[i].sum, &stu[j].sum);    /*交换总分*/
336                 SwapFloat(&stu[i].aver, &stu[j].aver); /*交换平均分*/
337                 SwapLong(&stu[i].num, &stu[j].num);    /*交换学分*/
338                 SwapChar(stu[i].name, stu[j].name);    /*交换姓名*/
339             }
340         }
341     }
342 }

343 /*功能：按学号查找学生成绩并显示查找结果
344 参数：学生信息结构体数组stu，人数n，课程数m
345 返回值：无
346 */
347 void SearchbyNum(STU stu[], int n, int m)
348 {
349     long number;
350     int i, j;
351     printf("Input the number you want to search:");
352     scanf("%ld", &number);
353     for(i=0; i<n; i++)
```

```
354 {
355     if(stu[i].num==number)
356     {
357         printf("%ld\t%s\t", stu[i].num, stu[i].name);
358         for(j=0; j<m; j++)
359         {
360             printf("%.0f\t", stu[i].score[j]);
361         }
362         printf("%.0f\t%.0f\n", stu[i].sum, stu[i].aver);
363         return;
364     }
365 }
366 printf("\nNot found! \n");
367 }
```

368 /*功能：按姓名查找学生成绩并显示查找结果

369 参数：学生信息结构体数组stu，人数n，课程数m

370 返回值：无

371 */

```
372 void SearchbyName(STU stu[], int n, int m)
```

```
373 {
374     char x[MAX_LEN];
375     int i, j;
376     printf("Input the name you want to search:");
377     scanf("%s", x);
378     for(i=0; i<n; i++)
379     {
380         if(strcmp(stu[i].name, x)==0)
381         {
382             printf("%ld\t%s\t", stu[i].num, stu[i].name);
383             for(j=0; j<m; j++)
384             {
385                 printf("%.0f\t", stu[i].score[j]);
386             }
387             printf("%.0f\t%.0f\n", stu[i].sum, stu[i].aver);
388             return;
389         }
390     }
391     printf("\nNot found! \n");
392 }
```

393 /*功能：统计各分数段的学生人数及所占的百分比

```
394 参数：学生信息结构体数组stu，人数n，课程数m
395 返回值：无
396 */
397 void StatisticAnalysis(STU stu[], int n, int m)
398 {
399     int i, j, total, t[6], g;
400     for(j=0; j<m; j++)
401     {
402         printf("For course %d:\n", j+1);
403         for(g=0; g<6; g++)    /*将数组t的元素初始化为0*/
404         {
405             t[g]=0;
406         }
407         for(i=0; i<n; i++)
408         {
409             if(stu[i].score[j]>=0&&stu[i].score[j]<60) t[0]++;
410             else if(stu[i].score[j]<70) t[1]++;
411             else if(stu[i].score[j]<80) t[2]++;
412             else if(stu[i].score[j]<90) t[3]++;
413             else if(stu[i].score[j]<100) t[4]++;
414             else if(stu[i].score[j]==100) t[5]++;
415         }
416         for(total=0, i=0; i<=5; i++)
417         {
418             total=total+t[i];
419         }
420         for(i=0; i<=5; i++)
421         {
422             if(i==0) printf("<60\t%d\t%.2f%\n", t[i], (float)t[i]/n*100);
423             else if(i==5) printf("%d\t%d\t%.2f\t%\n", (i+5)*10, t[i], (float)t[i]/n*100);
424             else printf("%d-%d\t%d\t%.2f%\n", (i+5)*10, (i+5)*10+9, t[i], (float)t[i]/n*100);
425         }
426     }
427 }

428 /*功能：打印学生成绩
429 参数：学生信息结构体数组stu，人数n，课程数m
430 返回值：无
431 */
432 void PrintScore(STU stu[], int n, int m)
433 {
434     int i, j;
435     for(i=0; i<n; i++)
```

```
436 {
437 printf("%ld\t%s\t", stu[i]. num, stu[i]. name);
438 for(j=0; j<m; j++)
439 {
440 printf("%.0f\t", stu[i]. score[j]);
441 }
442 printf("%.0f\t%.0f\n", stu[i]. sum, stu[i]. aver);
443 }
444 }

445 /*功能：输入学生信息到文件中
446 参数：学生信息结构体数组stu，人数n，课程数m
447 返回值：无
448 */
449 void WriteToFile(STU stu[], int n, int m)
450 {
451 FILE *ClassInfo;
452 if( ( ClassInfo = fopen("grade. date", "wt+") ) == NULL ) printf("Can not open the
file!\n");
453 fprintf(ClassInfo, "%d %d\n", n, m);
454 fwrite(stu, sizeof(STU), n, ClassInfo);
455 fclose(ClassInfo);
456 printf("\nWrite successfully!");
457 return ;
458 }

459 /*功能：读取包含学生信息的文件
460 参数：学生信息结构体数组stu，人数n指针，课程数m指针
461 返回值：无
462 */
463 void ReadFromFile(STU stu[], int *n, int *m)
464 {
465 FILE *ClassInfo;
466 if( ( ClassInfo = fopen("grade. date", "rt") ) == NULL ) printf("Can not open the
file!\n");
467 fscanf(ClassInfo, "%d %d", n, m);
468 fgetc(ClassInfo);
469 fread(stu, sizeof(STU), *n, ClassInfo);
470 fclose(ClassInfo);
471 printf("\nRead successfully. The student number is %d, and the course number
is %d. \n", *n, *m);
472 return ;
473 }
```

```
474 /*功能：读取文件中的课程信息并引导用户决定
475 参数：学生信息结构体数组stu，人数n的指针，课程数m指针
476 返回值：用户选择结果布尔值
477 */
478 int Init(STU stu[], int *n, int *m)
479 {
480 FILE *ClassInfo;
481 char c;
482 if( ( ClassInfo = fopen("grade.date", "rt") ) != NULL ) printf("Some grade data has
    been found, do you want to load?(Y/N):");
483 else return 0;
484 scanf("%c", &c);
485 if(c=='n' || c=='N') return 0;
486 fscanf(ClassInfo, "%d %d", n, m);
487 fgetc(ClassInfo);
488 fread(stu, sizeof(STU), *n, ClassInfo);
489 fclose(ClassInfo);
490 printf("The student number is %d, and the course number is %d. \n", *n, *m);
491 return 1;
492 }
```

系统测试

1. 程序调试中遇到的问题和解决方法及思路

困难一：定义了非常多函数，使用和管理混乱，使得程序的阅读和修改难度增加。

解决方法：添加注释，使用 Git 等工具进行代码托管。

困难二：进行数据库的设计首先必须了解与分析用户需求。需求分析是整个设计过程的基础，是最困难、最耗时的一部分。需求分析做的不好，甚至会导致整个数据库设计返工重做。需求分析的任务是通过详细的调查现实世界要处理的对象，充分了解用户的各种需求，然后在此基础上确定系统功能。系统还必须充分考虑今后可能的扩充和改变。

解决方法：自上而下的结构，紧随要求一步步完成，对各个函数逐个击破，不求花哨但求稳健。参考课本的范例，踏实的完成了各项功能。

2. 实验结果

主菜单

```
Input student number(n<=30) 5
Input course number(m<=6) 3
Management For Students' Score
(YOU MUST CHOOSE 1 AND 3!!!)
1. Input record
2. Caculate total and average score of every course
3. Caculate total and average score of every student
4. Sort in descending order by score
5. Sort in ascending order by score
6. Sort in ascending order by number
7. Sort in dictionary order by name
8. Search by number
```

```
9. Search by name
10. Statistic analysis
11. List record
12. Write to a file
13. Read from a file
0. Exit
Please Input your choice:
```

1. 录入每个学生的学号、姓名和各科考试成绩

```
Please Input your choice:
1
Input student's ID,name and score:
12301 qiqi 89 78 85
12302 wiwi 78 95 82
12303 roro 79 83 89
12304 cici 76 82 94
12305 yuyu 91 79 84
```

2. 计算每门课程的总分和平均分

```
Please Input your choice:
2
course 1:sum = 413,aver = 83
course 2:sum = 417,aver = 83
course 3:sum = 434,aver = 87
```

3. 计算每个学生的总分和平均分

```
Please Input your choice:
3
student 1:sum = 252,aver = 84
student 2:sum = 255,aver = 85
student 3:sum = 251,aver = 84
student 4:sum = 252,aver = 84
student 5:sum = 254,aver = 85
```

4. 按每个学生的总分由高到低排出名次表

```
Please Input your choice:
4

Sort in descending order by score:
12302  wiwi  78    95    82    255    85
12305  yuyu  91    79    84    254    85
12304  cici  76    82    94    252    84
12301  qiqi  89    78    85    252    84
12303  roro  79    83    89    251    84
```

5. 按每个学生的总分由低到高排出名次表

```
Please Input your choice:
5

Sort in ascending order by score:
12303  roro  79    83    89    251    84
12304  cici  76    82    94    252    84
12301  qiqi  89    78    85    252    84
12305  yuyu  91    79    84    254    85
12302  wiwi  78    95    82    255    85
```

6. 按学号由小到大排出成绩表

```
Please Input your choice:
6

Sort in ascending order by number:
12301  qiqi  89    78    85    252    84
12302  wiwi  78    95    82    255    85
12303  roro  79    83    89    251    84
12304  cici  76    82    94    252    84
12305  yuyu  91    79    84    254    85
```

7. 按姓名字典顺序排出成绩表

```
Please Input your choice:
7

Sort in ascending order by name:
12304  cici  76    82    94    252    84
12301  qiqi  89    78    85    252    84
12303  roro  79    83    89    251    84
12302  wiwi  78    95    82    255    85
12305  yuyu  91    79    84    254    85
```

8. 按学号查询学生排名及其考试成绩

```
Please Input your choice:
8
Input the number you want to search:12304
12304  cici  76      82      94      252      84
```

9. 按姓名查询学生排名及其考试成绩

```
Please Input your choice:
9
Input the name you want to search:qiqi
12301  qiqi  89      78      85      252      84
```

10. 按优秀（90~100）、良好（80~89）、中等（70~79）、及格（60~69）、不及格（0~59）5个类别，对每门课程分别统计每个类别的人数以及所占的百分比

```
Please Input your choice:
10
For course 1:
<60      0      0.00%
60-69    0      0.00%
70-79    3      60.00%
80-89    1      20.00%
90-99    1      20.00%
100      0      0.00 %
For course 2:
<60      0      0.00%
60-69    0      0.00%
70-79    2      40.00%
80-89    2      40.00%
90-99    1      20.00%
100      0      0.00 %
For course 3:
<60      0      0.00%
60-69    0      0.00%
70-79    0      0.00%
80-89    4      80.00%
90-99    1      20.00%
100      0      0.00 %
```

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。

```
Please Input your choice:
11
12304  cici  76      82      94      252      84
12301  qiqi  89      78      85      252      84
12303  roro  79      83      89      251      84
12302  wiwi  78      95      82      255      85
12305  yuyu  91      79      84      254      85
```

12. 将每个学生的纪录信息写入文件

```
Please Input your choice:
12

Write successfully!
Press Enter to continue...
```

13. 从文件中读出每个学生的纪录信息并显示

```
Please Input your choice:
13

Read successfully. The student number is 5, and the course number is 3.
```

(六) 分析总结、收获和体会

通过将近一个星期的设计，在课设中也遇到了很多问题，经过向老师询问跟同学们交流，认真的思考，积极解决问题，终于解决了那些问题。在解决问题的过程中，我也学到了很多知识，更是让我把课本的知识应用到实际之中，让我了解了我们学习的知识有什么用，增强我的自信心和学习的动力。同时感谢老师和同学在课程设计过程中的帮助。通过这次的系统编程，我学会了如何调用多个函数的使用，尽量简化代码，使程序优化。

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