



HARBIN INSTITUTE OF TECHNOLOGY

# 哈尔滨工业大学

## 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 V7.0。某班有最多不超过 30 人（具体人数由键盘输入）参加某门课程的考试，用**链表结构体**编程实现对学生成绩的管理。实现对学生信息的增删查改、排序等基本管理功能。

### 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

- (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。



(14) **删除学生（涉及链表）**



(15) **计算学分绩**



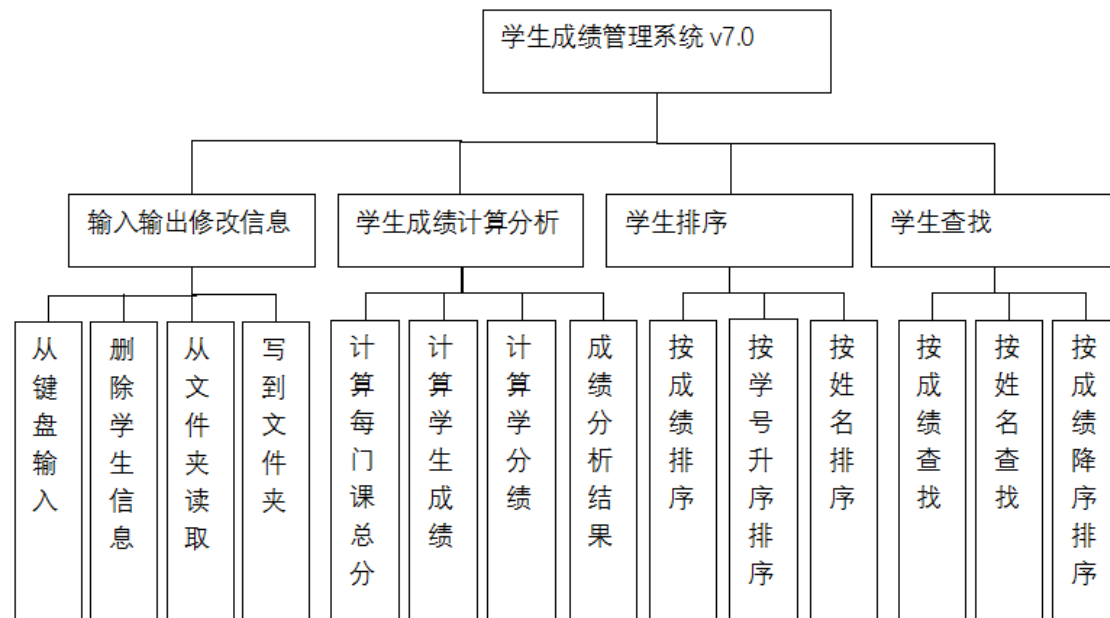
(16) **查找分数区间上的同学**

### 4.实验环境

Win10、dev c++

### (三) 系统设计

#### 1. 系统功能模块划分



#### 2. 算法设计

**直接选择排序：**第一次从  $R[0] \sim R[n-1]$  中选取最小值，与  $R[0]$  交换，第二次从  $R[1] \sim R[n-1]$  中选取最小值，与  $R[1]$  交换，……，第  $i$  次从  $R[i-1] \sim R[n-1]$  中选取最小值，与  $R[i-1]$  交换，……，第  $n-1$  次从  $R[n-2] \sim R[n-1]$  中选取最小值，与  $R[n-2]$  交换，总共通过  $n-1$  次，得到一个按排序码从小到大排列的有序序列。

#### 3. 数据结构设计(顶层数据字典)：

**结构体 student (学生数据)**

**student 结构体组成的链表 (储存所有学生信息)**

变量类型	变量名
long	num

char	name[MAX_LEN]
float	score[COURSE_NUM]
float	sum
float	aver
float	creditgrade
struct student *	next

### 结构体 Class

变量类型	变量名
float	point

## 4.各模块（函数）的功能和接口设计：

### 4.1 各个模块之间的逻辑关系图

#### 主函数：完成学生成绩功能

序号	函数名	函数功能	函数参数	函数返回值
1	swap	交换两个节点数据域信息	(STU *stu1, STU *stu2)	void
2	distribute	分配节点	(STU **stu, int n)	void
3	ReadScore	用于从键盘读取成绩	(STU *stu, int *m)	void
4	AverSumofEveryStudent	计算每名学生的总分	(STU *stu, int m)	void
5	AverSumofEveryCourse	计算每门课的总分	(STU *stu, int m, int n)	void
6	SortbyScore	按成绩排序（降序/升序）	(STU *stu, int (*compare)(float a, float b))	void
7	Ascending	比两数大小（用于函数 6）	(float a, float b)	int
8	Descending	比两数大小（用于函数 6）	(float a, float b)	int
9	AsSortbyNum	按学号升序排序	(STU *stu)	void
10	SortbyName	按姓名字典顺序排序	(STU *stu)	void
11	SearchbyNum	按学号查找	(STU *stu, int m)	void

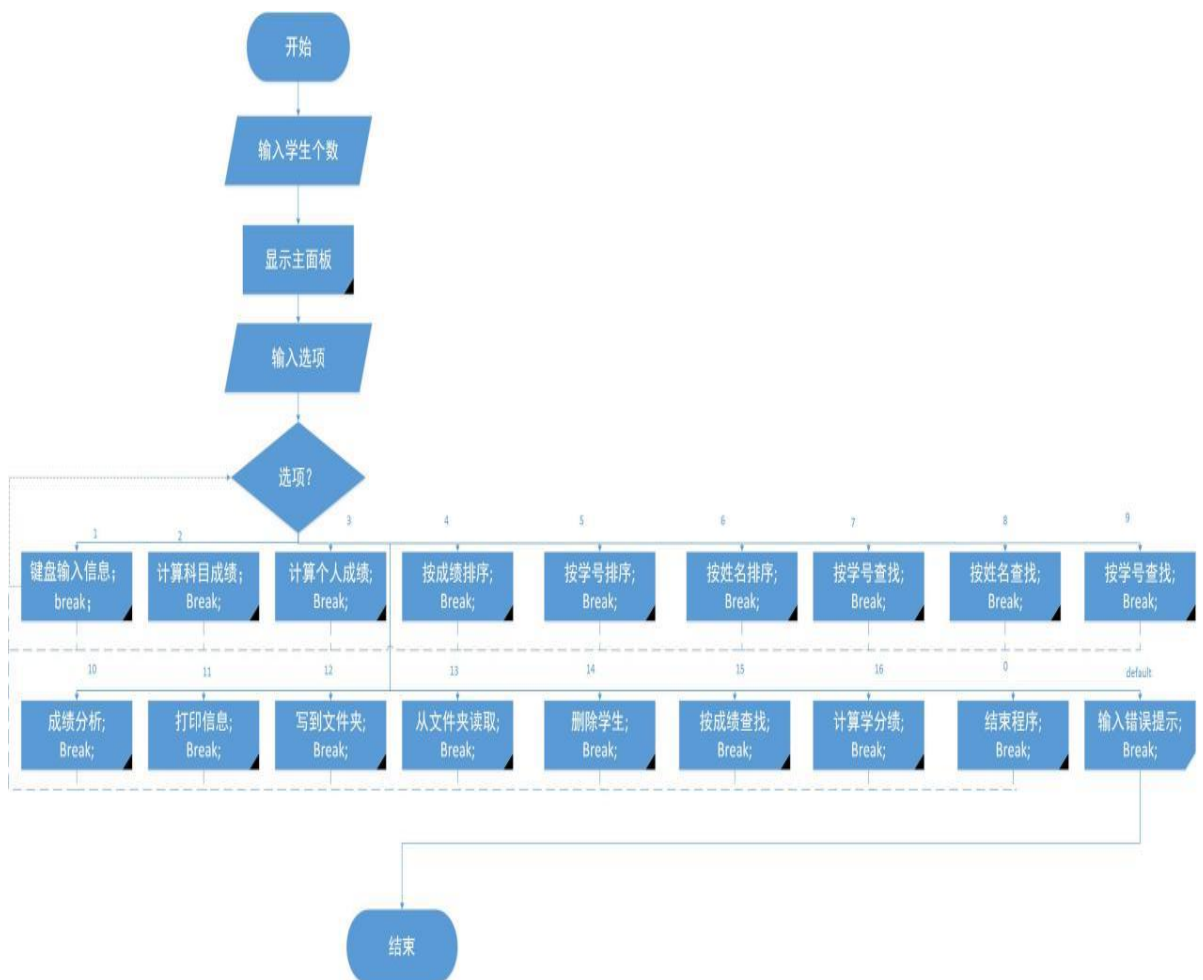
12	SearchbyName	按姓名查找	(STU *stu,int m)	void
13	StatisticAnalysis	成绩分析	(STU *stu,int m)	void
14	PrintScore	打印学生信息	(STU *stu,int m)	void
15	ReadfromFile	从文件读取信息	(STU *stu,int *n, int *m)	void
16	WritetoFile	将信息写到文件	(STU *stu,int n,int m)	void
17	Delete	删除学生信息	(STU *head)	STU *
18	SearchInInterval	按成绩查找	(STU *stu,int m)	void
19	CreditGrade	学分绩计算、排名	(STU *stu,int m)	void

#### (四) 系统实现

##### 1.系统的总流程图:

菜单驱动的系统的主流程:

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



### 3. 程序的全部源代码:

#### 学生成绩管理系统.Cpp

```
001 /*本程序特色：算学分并按学分排序，查找成绩区间上的所有同学，删除学生*/
002 #include<stdio.h>
003 #include<stdlib.h>
004 #include<string.h>
005 #define MAX_LEN 10 /* 字符串最大长度 */
006 #define STU_NUM 30 /* 最多的学生人数 */
007 #define COURSE_NUM 6 /* 最多的考试科目数 */
008 typedef struct Class
009 {
010 float point;
011 }CLASS;
012 typedef struct student
013 {
014 long num; /* 每个学生的学号 */
015 char name[MAX_LEN]; /* 每个学生的姓名 */
016 float score[COURSE_NUM]; /* 每个学生 COURSE_NUM 门功课的成绩 */
017 float sum; /* 每个学生的总成绩 */
018 float aver;
019 float creditgrade;
020 struct student *next; /* 每个学生的平均成绩 */
021 }STU;
022 int Menu(void);
023 void swap(STU *stu1, STU *stu2);
024 void ReadScore(STU *stu, int *m);
025 void AverSumofEveryStudent(STU *stu, int m);
026 void AverSumofEveryCourse(STU *stu, int m,int n);
027 void SortbyScore(STU *stu,int (*compare)(float a,float b));
028 int Ascending(float a, float b);
029 int Descending(float a, float b);
030 void AsSortbyNum(STU *stu);
031 void SortbyName(STU *stu);
032 void SearchbyNum(STU *stu,int m);
033 void SearchbyName(STU *stu,int m);
034 void StatisticAnalysis(STU *stu,int m);
035 void PrintScore(STU *stu, int m);
```

---

```
036 void WritetoFile(STU *stu,int n,int m);
037 void ReadfromFile(STU *stu,int *n,int *m);
038 void distribute(STU **stu, int n);
039 STU *Delete(STU *head);
040 void SearchInaInterval(STU *stu,int m);
041 void CreditGrade(STU *stu,int m);

042 /*主函数*/
043 int main()
044 {
045     int ch;
046     int n=0,m=0;
047     STU *stu;
048     do
049     {
050         printf("Input student number(n<30):\n");
051         scanf("%d",&n);
052         fflush(stdin);
053     }while(!(n<30&& n>0));
054     distribute(&stu,n);
055     while(1)
056     {
057         ch=Menu();
058         switch(ch)
059         {
060             case 1:
061                 ReadScore(stu,&m);
062                 break;
063             case 2:
064                 AverSumofEveryCourse(stu,m,n);
065                 break;
066             case 3:
067                 AverSumofEveryStudent(stu, m);
068                 break;
069             case 4:
070                 SortbyScore(stu,Descending);
071                 printf("Sort in descending order by score:\n");
072                 PrintScore(stu,m);
073                 break;
074             case 5:
075                 SortbyScore(stu,Ascending);
076                 printf("Sort in ascending order by score:\n");
077                 PrintScore(stu,m);
078                 break;
```

---

```
079 case 6:
080 AsSortbyNum(stu);
081 printf("Sort in ascending order by number:\n");
082 PrintScore(stu,m);
083 break;
084 case 7:
085 SortbyName(stu);
086 printf("Sort in dictionary order by name:\n");
087 PrintScore(stu,m);
088 break;
089 case 8:
090 SearchbyNum(stu,m);
091 break;
092 case 9:
093 SearchbyName(stu,m);
094 break;
095 case 10:
096 StatisticAnalysis(stu,m);
097 break;
098 case 11:
099 PrintScore(stu,m);
100 break;
101 case 12:
102 WritetoFile(stu,n,m);
103 break;
104 case 13:
105 ReadfromFile(stu, &n, &m);
106 break;
107 case 14:
108 stu=Delete(stu);
109 PrintScore(stu,m);
110 break;
111 case 15:
112 SearchInaInterval(stu,m);
113 break;
114 case 16:
115 CreditGrade(stu,m);
116 break;
117 case 0:
118 printf("End of program!\n");
119 exit(0);
120 default:
121 printf("Input error!\n");
122 }
```



---

```
123 }
124 return 0;
125 }
126 /*打印菜单*/
127 int Menu(void)
128 {
129 int itemSelected;
130 printf("Management for Students' scores\n");
131 printf("1.Input record\n");
132 printf("2.Caculate total and average score and credit grade of every
    course\n");
133 printf("3.Caculate total and average score of every student\n");
134 printf("4.Sort in descending order by score\n");
135 printf("5.Sort in ascending order by score\n");
136 printf("6.Sort in ascending order by number\n");
137 printf("7.Sort in dictionary order by name\n");
138 printf("8.Search by number\n");
139 printf("9.Search by name\n");
140 printf("10.Statistic analysis\n");
141 printf("11.List record\n");
142 printf("12.Write to a file\n");
143 printf("13.Read from a file\n");
144 printf("14.Delete a student\n");
145 printf("15.Search student in a interval of score\n");
146 printf("16.Calculate the credit grade of every student\n");
147 printf("0.Exit\n");
148 printf("Please Input your choice:\n");
149 scanf("%d",&itemSelected);
150 return itemSelected;
151 }
152 /*交换数据*/
153 void swap(STU *stu1, STU *stu2)
154 {
155 STU temp;
156 temp = *stu1; /*交换节点*/
157 *stu1 = *stu2;
158 *stu2 = temp;
159 temp.next=stu1->next; /*把指针域换回来*/
160 stu1->next=stu2->next;
161 stu2->next=temp.next;
162 }
163 /*申请内存*/
164 void distribute(STU **stu, int n)
165 {
```

---

```
166 int i;
167 *stu=(STU *)malloc(sizeof(STU));
168 STU *stutail=*stu;
169 for(i=1;i<n;i++)/*根据学生个数申请内存*/
170 {
171 stutail->next=(STU *)malloc(sizeof(STU));
172 stutail=stutail->next;
173 }
174 stutail->next=NULL;
175 }
176 /*读入数据*/
177 void ReadScore(STU *stu,int *m)
178 {
179 int j;
180 do{
181 printf("Input course number (m<=%d):\n",COURSE_NUM);
182 scanf("%d",m);
183 fflush(stdin);
184 }while(!(*m<=6&&*m>0));/*确保输入数据正确*/
185 printf("Input student's ID, name and score:\n");
186 for( ;stu!=NULL;stu=stu->next)/*读入数据*/
187 {
188 scanf("%ld%s",&stu->num,stu->name);
189 for(j=0;j<*m;j++)
190 scanf("%f",&stu->score[j]);
191 fflush(stdin);
192 }
193 }
194 /*计算每个学生得平均成绩*/
195 void AverSumofEveryStudent(STU *stu, int m)
196 {
197 int j;
198 for( ;stu!=NULL;stu=stu->next)
199 {
200 stu->sum=0;
201 for(j=0;j<m;j++)/*对每个学生成绩求和*/
202 {
203 stu->sum+=stu->score[j];
204 }
205 stu->aver=m>0?stu->sum/m:-1;
206 printf("student %d: sum=%.0f,aver=%.0f\n",stu->num,stu->sum,stu->aver);
207 }
208 }
209 /*计算每门课得平均成绩*/
```

---

```
210 void AverSumofEveryCourse(STU *stu, int m,int n)
211 {
212 int j;
213 STU *blank;
214 float sum[COURSE_NUM],aver[COURSE_NUM];
215 for(j=0;j<m;j++)/*对每门课成绩求和*/
216 {
217 blank=stu;
218 sum[j]=0;
219 for(;blank!=NULL;blank=blank->next)
220 {
221 sum[j]=sum[j]+blank->score[j];
222 }
223 aver[j]= n>0 ? sum[j]/n: -1;
224 printf("course %d:sum=%.0f,aver=%.0f\n",j+1,sum[j],aver[j]);
225 }
226 }
227 /*按成绩排序*/
228 void SortbyScore(STU *stu,int (*compare)(float a,float b))
229 {
230 STU *i,*j,*temp;
231 for(i=stu;i->next!=NULL;i=i->next)
232 {
233 temp=i;
234 for(j=i->next;j!=NULL;j=j->next)/*寻找最大值*/
235 {
236 if(compare(j->sum,temp->sum))
237 temp=j;
238 }
239 if(temp!=i)/*判断是否需要交换*/
240 {
241 swap(i,temp);
242 }
243 }
244 }
245 /*升序*/
246 int Ascending(float a, float b)
247 {
248 return a<b;
249 }
250 /*降序*/
251 int Descending(float a, float b)
252 {
253 return a>b;
```

---

```
254 }
255 /*按学号升序排序*/
256 void AsSortbyNum(STU *stu)
257 {
258     STU *i,*j,*temp;
259     for(i=stu;i->next!=NULL;i=i->next)/*按学号进行排序*/
260     {
261         temp=i;
262         for(j=i->next;j!=NULL;j=j->next)/*寻找最大值*/
263             if(j->num<temp->num)
264                 temp=j;
265         if(i!=temp)/*判断是否需要交换*/
266             swap(i,temp);
267     }
268 }
269 /*按姓名排序*/
270 void SortbyName(STU *stu)
271 {
272     STU *i,*j,*temp;
273     for(i=stu;i->next!=NULL;i=i->next)/*按姓名排序*/
274     {
275         temp=i;
276         for(j=i->next;j!=NULL;j=j->next)/*寻找最大值*/
277             if(strcmp(j->name,i->name)<0)
278                 temp=j;
279         if(temp!=i)/*判断是否需要交换*/
280             swap(i,temp);
281     }
282 }
283 /*按学号查找*/
284 void SearchbyNum(STU *stu, int m)
285 {
286     long number;
287     STU *i;
288     int j;
289     printf("INput the number you want to search:");
290     scanf("%ld",&number);
291     for(i=stu;i!=NULL;i=i->next)
292     {
293         if(i->num==number)/*判断是否为目标*/
294         {
295             printf("%ld\t%s\t",i->num,i->name);
296             for(j=0;j<m;j++)
297             {
```

---

```
298 printf("%0.f\t",i->score[j]);
299 }
300 printf("%0.f\t%0.f\n",i->sum,i->aver);
301 return;
302 }
303 }
304 printf("Not found!\n");
305 }
306 /*按姓名查找*/
307 void SearchbyName(STU *stu, int m)
308 {
309 char x[MAX_LEN];
310 int j;
311 printf("Input the name you want to search:\n");
312 scanf("%s",x);
313 for( ;stu!=NULL;stu=stu->next)
314 {
315 if(strcmp(stu->name,x)==0) /*判断是否是目标*/
316 {
317 printf("%ld\t%s\t",stu->num,stu->name);
318 for(j=0;j<m;j++)
319 {
320 printf("%0.f\t",stu->score[j]);
321 }
322 printf("%0.f\t%0.f\n",stu->sum,stu->aver);
323 return;
324 }
325 }
326 printf("Not found!\n");
327 }
328 /*数据分析*/
329 void StatisticAnalysis(STU *stu,int m)
330 {
331 int j,total,k,t[6];
332 STU *i;
333 for(j=0;j<m;j++)
334 {
335 printf("For course %d:\n",j+1);
336 memset(t,0,sizeof(t));
337 for(i=stu;i!=NULL;i=i->next)
338 {
339 if(i->score[j]>=0&&i->score[j]<=60) t[0]++;
340 else if(i->score[j]<70) t[1]++;
341 else if(i->score[j]<80) t[2]++;
```

---

```
342 else if (i->score[j]<90) t[3]++;
343 else if (i->score[j]<100) t[4]++;
344 else if (i->score[j]==100) t[5]++;
345 }
346 for(total=0,k=0;k<6;k++)
347 {
348 total+=t[k];
349 }
350 for(k=0;k<=5;k++)
351 {
352 if(k==0)
353 printf("<60\t%d\t%.2f%%\n",t[0],t[0]*1.0/total*100);
354 else if(k==5)
355 printf("%d\t%d\t%.2f%%\n",100,t[4],t[4]*1.0/total*100);
356 else
357 printf("%d-%d\t%d\t%.2f%%\n",50+k*10,60+k*10,t[k],t[k]*1.0/total*100);
358 }
359 }
360 }
361 /*打印函数*/
362 void PrintScore(STU *stu,int m)
363 {
364 int j;
365 STU *i;
366 for(i=stu;i!=NULL;i=i->next)
367 {
368 printf("%ld\t%s\t",i->num,i->name);
369 for(j=0;j<m;j++)
370 {
371 printf("%.0f\t",i->score[j]);
372 }
373 printf("%.0f\t%.0f\n",i->sum,i->aver);
374 }
375 }
376 /*从文件中输入*/
377 void ReadfromFile(STU *stu,int *n, int *m)
378 {
379 FILE *fp;
380 int i,j;
381 int temp = *n;
382 if((fp = fopen("student.txt", "r")) == NULL)
383 {
384 printf(" Fail to open student.txt \n");
385 return ;
```

---

```
386 }
387 else
388 {

389 fscanf(fp, "%d\t%d\n", n, m);
390 if(temp<*n)
391 {
392 free(stu);
393 distribute(&stu, *n);
394 }

395 for(i=0;i<*n;i++)
396 {
397 fscanf(fp, "%ld%s\t", &stu->num, stu->name);
398 for(j=0;j<*m;j++)
399 {
400 fscanf(fp,"%f\t", &stu->score[j]);
401 }
402 fscanf(fp, "%f\t%f\n", &stu->sum, &stu->aver);
403 stu=stu->next;
404 }
405 printf("Import Successfully!\n");
406 }
407 }
408 /*输出到文件*/
409 void WritetoFile(STU *stu,int n,int m)
410 {
411 FILE *fp;
412 int i,j;
413 if((fp = fopen("student.txt", "w"))==NULL)
414 {
415 printf("Fail to open student.txt ");
416 return;
417 }
418 else
419 {
420 fprintf(fp, "%d\t%d\n", n, m);
421 for(i=0;i<n;i++)
422 {
423 fprintf(fp,"%12ld%12s", stu->num, stu->name);
424 for(j=0;j<m;j++)
425 {
426 fprintf(fp,"%12.0f", stu->score[j]);
427 }
```

---

```
428 fprintf(fp,"%12.0f%12.0f\n", stu->sum, stu->aver );
429 stu=stu->next;
430 }
431 printf("Export Successfully!\n");
432 }
433 fclose(fp);
434 }
435 /*删除学生信息*/
436 STU *Delete(STU *head)
437 {
438     long num;
439     char a[5],name[10],c;
440     STU *node1=head;
441     STU *node2=NULL; /*用于存储目标节点上一个节点*/
442     scanf("%c",&c);
443     while(c!='\n') /*清空缓存区*/
444     {
445         scanf("%c",&c);
446     }
447     printf("Do you want to enter name or num\n");
448     gets(a);
449     if(strcmp(a,"name")==0)
450     {
451         printf("Please enter the name of the student you want to delete.\n");
452         gets(name); /*获得删除信息*/
453         if (head==NULL)
454         {
455             return NULL; /*如果链表为空，返回 NULL*/
456         }
457         else /*如果链表不为空*/
458         {
459             if (strcmp(node1->name,name)==0) /*如果头节点为目标*/
460             {
461                 head=head->next;
462                 free(node1);
463                 return head;
464             }
465             else /*如果头结点不是目标*/
466             {
467                 while (node1!=NULL)
468                 {
469                     node2=node1;
470                     node2=node2->next;
471                     if (strcmp(node2->name,name)==0)
```



---

```
472 {
473 node1->next=node2->next;
474 free(node2);
475 break;
476 }
477 node1=node1->next;
478 }
479 return head;
480 }
481 }

482 }
483 else if(strcmp(a,"num")==0)/*如果为学号删除*/
484 {
485 printf("Please enter the num of the student you want to delete.");
486 scanf("%ld",&num);
487 if (head==NULL)
488 {
489 return NULL;
490 }
491 else
492 {
493 if (node1->num==num)
494 {
495 head=head->next;
496 free(node1);
497 return head;
498 }
499 else
500 {
501 while (node1!=NULL)
502 {
503 node2=node1;
504 node2=node2->next;
505 if (node2->num==num)
506 {
507 node1->next=node2->next;
508 free(node2);
509 break;
510 }
511 node1=node1->next;
512 }
513 return head;
514 }
```

---

```
515 }
516 }
517 }
518 /*根据总成绩进行区间查找*/
519 void SearchInaInterval(STU *stu,int m)
520 {
521 STU *head=stu;
522 int lower=0,upper=-1,j,i=0;/*lower,upper 分别为删除成绩的上下线*/
523 SortbyScore(stu,Descending);
524 for( ;head!=NULL;head=head->next)/*先求的总成绩*/
525 {
526 head->sum=0;
527 for(j=0;j<m;j++)
528 {
529 head->sum+=head->score[j];
530 }
531 stu->aver=m>0?stu->sum/m:-1;
532 }
533 while(upper<lower)/*保证上限比下限大, 数据输入正确*/
534 {
535 printf("Please enter the Interval upper limit and Interval lower limit(lower
    can be equal to upper)(sum score is %d)\n",100*m);
536 scanf("%d%d",&upper,&lower);
537 }
538 for( ;stu!=NULL;stu=stu->next)
539 {
540 if(stu->sum<=upper&&stu->sum>=lower)
541 {
542 i++;
543 printf("%10ld%10s\t", stu->num, stu->name);
544 for(j=0;j<m;j++)
545 printf("%.0f\t", stu->score[j]);
546 printf("%.0f\t%.0f\n", stu->sum, stu->aver);
547 }
548 }
549 if(i==0)
550 printf("Not Found\n");
551 else
552 printf("%d students' score are between %d and %d",i,lower,upper);
553 }
554 /*用于计算学分绩*/
555 void CreditGrade(STU *stu,int m)
556 {
557 int i,count=1,countc=0;/*countc 用于记录考试课门数*/
```

---

```
558 float sumcredit=0;
559 STU *pi,*j,*k,*b;
560 pi=j=b=stu;
561 char a[MAX_LEN],c;
562 CLASS cla[COURSE_NUM];
563 for(i=0;i<m;i++)
564 {
565 strcpy(a,"a");/*用于判断是否为考试课*/
566 while(!(strcmp(a,"y")==0||strcmp(a,"Y")==0||strcmp(a,"n")==0||strcmp(a,
    "N")==0))
567 {
568 printf("Is the %d course a compulsory course?(input 'y'or'n')",i+1);
569 scanf(" %s",a);
570 scanf("%c",&c);
571 while(c!='\n')
572 {
573 scanf("%c",&c);
574 }
575 }
576 if(strcmp(a,"y")==0||strcmp(a,"Y")==0)/*读入考试课对应学分*/
577 {
578 countc++;
579 printf("Please input the credit");
580 scanf("%d",&cla[i].point);
581 sumcredit+=cla[i].point;
582 }
583 else
584 cla[i].point=0;
585 }
586 for( ;stu!=NULL;stu=stu->next)/*计算每个学生的学分绩*/
587 {
588 stu->creditgrade=0;
589 for(i=0;i<m;i++)
590 {
591 stu->creditgrade+=cla[i].point*stu->score[i];
592 }
593 stu->creditgrade/=sumcredit;
594 }
595 for( ;pi!=NULL;pi=pi->next)/*对学生学分绩排名*/
596 {
597 k=pi;
598 for(j=pi->next;j!=NULL;j=j->next)
599 if(j->creditgrade>pi->creditgrade)
600 k=j;
```

```
601 if(k!=pi)
602 swap(pi,k);
603 }
604 if(countc!=0)
605 for( ;b!=NULL;b=b->next)/*打印排名成绩等信息*/
606 {
607 printf("num=%ld          name=%s          creditgrade=%.2f\n",b->num,b->name,b->creditgrade,count);
608 count++;
609 }
610 else if(countc==0)/*没有考试课的情况*/
611 printf("There is no compulsory course\n");
612 }
```

## (五) 系统测试

### 1. 程序调试中遇到的问题和解决方法及思路

- 1.大段代码难以管理和修改，**解决办法**：使用更强大的代码托管工具，增加注释。
- 2.链表申请数量错误，**解决办法**：重新审计代码，认真修改。
- 3.从文件输入会将学生名字读错，**解决办法**：修改了读取方式。
4. 输入错误系统会崩溃，**解决办法**：用 while 语句清空缓存区。

### 2. 实验结果

菜单界面：

```
Input student number(n<30):
2
Management for Students' scores
1.Input record
2.Caculate total and average score and credit grade 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
14.Delete a student
15.Search student in a interval of score
16.Calculate the credit grade of every student
0.Exit
Please Input your choice:
```

## 1、键盘输入学生信息

```

Please Input your choice:
1
Input course number(m<=6):
2
Input student's ID, name and score:
100 wang 100 70
101 li 100 80

```

## 2、计算学科总分

```

Please Input your choice:
2
course 1:sum=473,aver=79
course 2:sum=466,aver=78
course 3:sum=492,aver=82

```

## 3、计算个人总分

```

Please Input your choice:
3
student 11003006: sum=289,aver=96
student 11003005: sum=280,aver=93
student 11003001: sum=258,aver=86
student 11003003: sum=233,aver=78
student 11003004: sum=206,aver=69
student 11003002: sum=165,aver=55

```

## 4、按成绩排名（降）

```

Please Input your choice:
4
Sort in descending order by score:
11003006      suyu      100      95      94      289      96
11003005      heli       98       92      90      280      93
11003001      lisi       87       82      89      258      86
11003003      ludi       75       78      80      233      78
11003004      zuma       65       69      72      206      69
11003002      dumo       48       50      67      165      55

```

## 5、按成绩排名（升）

```

Please Input your choice:
5
Sort in ascending order by score:
11003002      dumo       48       50      67      165      55
11003004      zuma       65       69      72      206      69
11003003      ludi       75       78      80      233      78
11003001      lisi       87       82      89      258      86
11003005      heli       98       92      90      280      93
11003006      suyu      100      95      94      289      96

```

## 6、按学号排序

```

Please Input your choice:
6
Sort in ascending order by number:
11003001      lisi      87      82      89      258      86
11003002      dumo      48      50      67      165      55
11003003      ludi      75      78      80      233      78
11003004      zuma      65      69      72      206      69
11003005      heli      98      92      90      280      93
11003006      suyu      100     95      94      289      96

```

## 7、按姓名排序

```

Please Input your choice:
7
Sort in dictionary order by name:
11003005      heli      98      92      90      280      93
11003002      dumo      48      50      67      165      55
11003001      lisi      87      82      89      258      86
11003006      suyu      100     95      94      289      96
11003003      ludi      75      78      80      233      78
11003004      zuma      65      69      72      206      69

```

## 8、按学号查找

```

Please Input your choice:
8
Input the number you want to search:11003006
11003006      suyu      100     95      94      289      96

```

## 9、按姓名查找

```

Please Input your choice:
9
Input the name you want to search:
zuma
11003004      zuma      65      69      72      206      69

```

## 10、成绩分析

```

Please Input your choice:
10
For course 1:
<60      1      16.67%
60-70    1      16.67%
70-80    1      16.67%
80-90    1      16.67%
90-100   1      16.67%
100      1      16.67%

```

```

For course 2:
<60      1      16.67%
60-70    1      16.67%
70-80    1      16.67%
80-90    1      16.67%
90-100   2      33.33%
100      2      33.33%

```

```
For course 3:
<60    0    0.00%
60-70   1    16.67%
70-80   1    16.67%
80-90   2    33.33%
90-100  2    33.33%
100     2    33.33%
```

## 11、显示成绩

```
Please Input your choice:
11
11003006      suyu      100      95      94      289      96
11003005      heli      98       92      90      280      93
11003001      lisi      87       82      89      258      86
11003003      ludi      75       78      80      233      78
11003004      zuma      65       69      72      206      69
11003002      dumo      48       50      67      165      55
```

## 12、输出到文件

```
Please Input your choice:
12
Export Successfully!
```

## 13、从文件输入

```
Please Input your choice:
13
Import Successfully!
```

## 14、删除学生

按姓名

```
Please Input your choice:
14
Do you want to enter name or num
name
Please enter the name of the student you want to delete.
zuma
11003006      suyu      100      95      94      289      96
11003005      heli      98       92      90      280      93
11003001      lisi      87       82      89      258      86
11003003      ludi      75       78      80      233      78
11003002      dumo      48       50      67      165      55
```

按学号

```
Please Input your choice:
14
Do you want to enter name or num
num
Please enter the num of the student you want to delete.11003004
11003006      suyu      100      95      94      289      96
11003005      heli      98       92      90      280      93
11003001      lisi      87       82      89      258      86
11003003      ludi      75       78      80      233      78
11003002      dumo      48       50      67      165      55
```

## 15、成绩查找

```
Please Input your choice:
15
Please enter the Interval upper limit and Interval lower limit(lower can be equal to upper)(sum score is 300)
250 300
Please enter the Interval upper limit and Interval lower limit(lower can be equal to upper)(sum score is 300)
300 250
11003006      suyu   100    95    94    289    96
11003005      heli   98     92    90    280    93
11003001      lisi   87     82    89    258    86
```

16、

## 16、学分绩计算

```
Please Input your choice:
16
Is the 1 course a compulsory course?(input 'y' or 'n')y
Please input the credit4
Is the 2 course a compulsory course?(input 'y' or 'n')y
Please input the credit6
Is the 3 course a compulsory course?(input 'y' or 'n')y
Please input the credit3
num=11003006 name=suyu creditgrade=96.31 rank=1
num=11003005 name=heli creditgrade=93.38 rank=2
num=11003001 name=lisi creditgrade=85.15 rank=3
num=11003003 name=ludi creditgrade=77.54 rank=4
num=11003004 name=zuma creditgrade=68.46 rank=5
num=11003002 name=dumo creditgrade=53.31 rank=6
```

## (六) 分析总结、收获和体会

**题目理解：**学生管理系统的作用就是更加方便的管理学生成绩，所以应该有更详细的注释，方便使用的功能，具有增删查改四个基本功能，从文件夹输入输出的功能尤为重要，该功能可以方便的进行成绩管理。管理系统的用户对象应当是老师，为了因材施教教导学生，还应当增加学生学分绩查询，按分数段查看学生等模块以增加实用性、便捷性。

**感受：**较大的程序首先要分块构思，将主程序，函数的功能想好，分好，然后一个个写，逐个击破。最好将尽可能多的功能写到函数中，而不是主函数。在写代码的时候要用好 define，写完一个函数 debug 一次。如果在写完程序后一起 debug，容易因为程序过多，不好找到。同时，我自行提高了题目要求，增加了删除学生信息和学分绩计算查询、按分数段查看学生的模块，极大提高了该学生成绩管理系统的实用性、便捷性。

## 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