**程序设计与实践**

**实验报告**

**实验六：学生成绩管理**

**实验时间：2022.04.27 ~ 2022.05.05**

**目录**

**实验名称………………………………………………………………P3**

**实验环境………………………………………………………………P3**

**实验目的………………………………………………………………P3**

**实验步骤………………………………………………………………P3**

**1. 头文件…………….…………………………..………..………..…………………..P3**

**2. 主函数…………..………………….…………………….………..………………...P4**

**3. 成绩记录与处理…………………….………..………..………….………………...P5**

**4. 成绩管理排序…………………….………..………..……………………….……...P7**

**5. 转换时间…………………….………..………………………...…………………...P7**

**6. 程序运行截图 …………………….………..…………...……..…………………...P8**

**实验总结………………………………………………………………P12**

**参考资料………………………………………………………………P12**

**一：****实验名称：学生成绩管理**

**二：实验环境: 一台安装有C语言编译器的电脑。**

**三：实验目的**

（1）培养和锻炼对具有一定复杂度和规模的问题的分析与求解能力。

（2）掌握程序设计的综合方法，能综合应用各种数据类型实现较复杂数据的存储。

（3）培养良好的C程序设计风格与代码规范意识。

**四：实验内容**

设计一个菜单驱动的学生成绩管理程序，

管理n个学生的m门考试科目成绩，实现以下基本功能。

(1)能够新增学生信息，并计算总分和平均分。

(2)能够根据学号修改和删除某学生信息。

(3)能够显示所有学生的成绩信息。

(4)能够分别按总分和学号进行排序。

(5)能够根据学号查询该学生的基本信息。

(6)学生成绩数据最终保存在文件**scores.txt**中，能够对文件读、写学生数据。

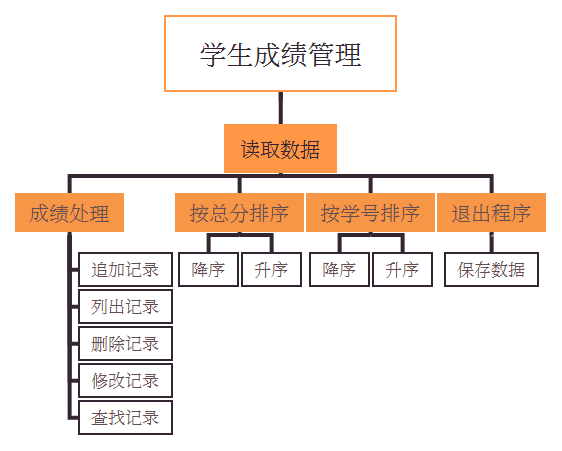


图4.1 学生成绩管理的功能图

**五：实验步骤**

**本次实验的基本要求是完成****学生成绩管理，其中包括****追加记录、列出记录、删除记录、修改记录、查找记录，分别根据总分和学号按降序、升序进行排序。同时对数据进行读取、保存等功能。按照不同功能需求分别写了4个C、C++文件和1个.h头文件，并完成合并编译、运行。**

1、头文件 student.h

student.h中引用多个头文件，并定义结构体数组。

*struct student //结构体*

*{*

*char num[11];*

*char \*name[20];*

*int score[SUBJECT];*

*int m;*

*float sum;*

*float avg;*

*}s1[MAX];*

2、主函数 config.cpp

主函数config.cpp中，首先导入目录内的.h头文件和C文件，并定义变量count为全局变量进行计数。使用do-while和switch函数对不同功能进行分类处理。还有保证数据txt文件正常导入、正常使用、保存数据、另存数据、删除空文件和关闭后台文件等文件处理。同时在导入、导出，使用时间戳转换为本地时间。

*#include "student.h" //引用头文件*

*#include "recorder.c" //引用其他C、C++文件*

*#include "sorts.c"*

*#include "time.cpp"*

*int count=0; //全局变量*

*int main()*

*{*

*FILE \*fp; //文件指针 - fp*

*FILE \*backup; //文件指针 - backup(备份)*

*char ch; //从原文件将数据一个个备份到新的文件变量ch*

*/\*另存一份备份文件\*/*

*fp=fopen("scores.txt","r"); //打开文件，文件必存在，否则会跳过*

*backup=fopen("scores\_backup.txt","w+"); //创建并打开新文件，可以进行读写*

*fscanf(fp,"%d",&count); //读取count当前的数*

*if(count==0) //如果count为0，则关闭文件；反之备份文件并关闭两个文件*

*{*

*fclose(fp);*

*}*

*else*

*{*

*while ((ch = fgetc(fp)) != EOF)*

*fputc(ch, backup);*

*fclose(fp);*

*fclose(backup);*

*}*

*/\*数据导入程序\*/*

*fp=fopen("scores.txt","a+"); //打开读写文件*

*fscanf(fp,"%d",&count); //读取数据到程序中*

*for(int t=1;t<=count;t++)*

*{*

*fscanf(fp,"%d%s%s%f%f",&s1[t].m,s1[t].num,&s1[t].name,&s1[t].sum,&s1[t].avg);*

*for(int s=0;s<s1[t].m;s++)*

*{*

*fscanf(fp,"%d",&s1[t].score[s]);*

*}*

*}*

*printf("\n####数据导入成功！####\n"); //提示用户完成正常读取，并输出当前时间和关闭文件*

*time();*

*fclose(fp);*

*/\*菜单框架\*/*

*/\*数据处理\*/*

*int choice; //do-while+switch分类处理各环节*

*do*

*{*

*printf("\n请输入：");*

*if(scanf("%d",&choice)!=1)*

*{*

*printf("\n输入错误！自动保存数据并退出。\n");*

*break;*

*}*

*switch (choice)*

*{*

*case 0: //退出程序*

*{*

*printf("是否删除备份？删除请输入‘1’。\n请输入："); //删除备份*

*char q;*

*getchar(); //吸收一个回车字符*

*scanf("%c",&q);*

*if(q=='1')*

*{*

*fclose(backup);*

*if (remove("scores\_backup.txt") == 0) //删除备份，并提示用户*

*{*

*printf("####已删除备份！####\n\n");*

*}*

*}*

*printf("####成功退出程序！####\n"); //完成退出*

*break;*

*}*

*}*

*}*

*while(choice!=0);*

*/\*保存数据到文件\*/*

*fp=fopen("scores.txt","w+");*

*fprintf(fp,"%d\n",count);*

*for(int t=1;t<=count;t++)*

*{*

*fprintf(fp,"%d %s %s %.2f %.2f ",s1[t].m,s1[t].num,&s1[t].name,s1[t].sum,s1[t].avg);*

*for(int s=0;s<s1[t].m;s++)*

*{*

*fprintf(fp,"%d ",s1[t].score[s]);*

*}*

*fprintf(fp,"\n");*

*}*

*fclose(fp);*

*printf("\n####数据导出成功！####\n");*

*time();*

*/\*检查文件是否有意义\*/*

*int test;*

*fp=fopen("scores.txt","r"); //打开存在文件，读文件*

*fscanf(fp,"%d",&test);*

*if(test==0) //文件没有记录成绩，自动删除文件*

*{*

*fclose(fp); //要先关闭已打开的文件*

*remove("scores.txt");*

*}*

*else*

*{*

*fclose(fp);*

*}*

*system("pause");*

*return 0;*

*}*

3、成绩记录与处理 recorder.c

recorder.c中实现追加记录、列出记录、删除记录、修改记录和查找记录等功能。

(1)追加记录

*void Append() //追加记录*

*{*

*int i;*

*count++;*

*printf("\n##请输入学号：");*

*scanf("%s",s1[count].num);*

*printf("\n##请输入姓名：");*

*scanf("%s",&s1[count].name);*

*printf("\n##请输入科数：");*

*scanf("%d",&s1[count].m);*

*if(s1[count].m>20) //限制科目数*

*{*

*printf("####追加失败，超出范围，请重新输入。####\n");*

*count--;*

*}*

*else*

*{*

*printf("\n##请输入各科成绩(空格隔开)：");*

*for(i=0;i<s1[count].m;i++){ //导入分科成绩*

*float ss;*

*scanf("%f",&ss);*

*if(ss-(int)ss<0.5) s1[count].score[i]=int(ss); //四舍五入*

*else s1[count].score[i]=int(ss)+1;*

*s1[count].sum+=s1[count].score[i]; //计算总分*

*}*

*s1[count].avg=s1[count].sum/s1[count].m; //计算平均分*

*}*

*}*

(2)列出记录

*void List() //列出记录*

*{*

*int i,j;*

*printf("\n #学号# #姓名# #总分# #平均分#\n");*

*for(i=1;i<=count;i++){*

*printf("%8s %-8s%8.2f%10.2f ",s1[i].num,&s1[i].name,s1[i].sum,s1[i].avg);*

*for(j=0;j<s1[i].m;j++){*

*printf("##科目%-2d：%3d分 ",j+1,s1[i].score[j]);*

*}*

*printf("\n");*

*}*

*printf("\n####列出完成！####\n");*

*}*

(3)删除记录，清空记录

*void Delete() //删除记录*

*{*

*printf("是否全部删除？全部删除请输入‘1’。\n请输入：");*

*getchar();*

*char b;*

*scanf("%c",&b);*

*if(b=='1')*

*{*

*count=0; //清空记录*

*}*

*else*

*{*

*char a[11];*

*int i,flag=0,j;*

*printf("\n##请输入需要删除的学号：");*

*scanf("%s",a);*

*for(i=1;i<=count;i++)*

*{*

*if(strcmp(a,s1[i].num)==0) //查找对应学号*

*{*

*printf("\n####查找成功！");*

*flag=i;*

*break;*

*}*

*}*

*if(flag!=0)*

*{*

*for(i=flag;i<=count;i++)*

*{*

*s1[i]=s1[i+1]; //删除记录*

*}*

*count--;*

*printf("删除成功！####\n");*

*List();*

*}*

*if(flag==0)*

*{*

*printf("\n####查找失败！删除失败！####\n");*

*}*

*}*

*}*

(4)修改记录

*void Modify() //修改记录*

*{*

*char a[11];*

*int i,flag=0,j;*

*printf("\n##请输入需要修改的学号：");*

*scanf("%s",a);*

*for(i=1;i<=count;i++)*

*{*

*if(strcmp(a,s1[i].num)==0)*

*{*

*printf("\n####查找成功！####\n"); //查找记录*

*flag=i;*

*break;*

*}*

*}*

*if(flag!=0) //重新填入数据*

*if(flag==0){*

*printf("\n####查找失败！修改失败！####\n");*

*}*

*}*

(5)查找记录

*void Search() //查找记录*

*{*

*char a[11];*

*int i,flag=0,j;*

*printf("\n##请输入需要查找的学号：");*

*scanf("%s",a);*

*for(i=1;i<=count;i++)*

*{*

*if(strcmp(a,s1[i].num)==0)*

*{*

*printf("\n####查找成功！####\n");*

*flag=i;*

*break;*

*}*

*}*

*if(flag!=0)*

*{*

*printf("\n #学号# #姓名# #总分# #平均分#\n");*

*printf("%8s %-8s%8.2f%10.2f ",s1[flag].num,&s1[flag].name,s1[flag].sum,s1[flag].avg);*

*for(j=0;j<s1[i].m;j++){*

*printf("##科目%-2d：%3d分 ",j+1,s1[flag].score[j]);*

*}*

*printf("\n");*

*}*

*if(flag==0){*

*printf("\n####查找失败！####\n");*

*}*

*}*

4、成绩管理排序 sort.c

sort.c中对总分和学号分别进行升序和降序排序。在降序排序中使用冒泡排序，在升序排序中使用选择排序。由于基本类似，本文仅提供其中2个代码片段。

(1)总分降序排序

*void descending\_sum()*

*{*

*int i,j;*

*for(i=1;i<=count;i++){*

*for(j=1;j<=count-i;j++){*

*if(s1[j].sum<s1[j+1].sum){*

*s1[0]=s1[j];*

*s1[j]=s1[j+1];*

*s1[j+1]=s1[0];*

*}*

*}*

*}*

*List();*

*printf("\n####按总分降序排序完成####\n");*

*}*

(2)学号升序排序

*void ascending\_num()*

*{*

*int q,p;*

*for(q=1;q<=count-1;q++){*

*for(p=q+1;p<=count;p++){*

*if(strcmp(s1[q].num,s1[p].num)>0){*

*s1[0]=s1[q];*

*s1[q]=s1[p];*

*s1[p]=s1[0];*

*}*

*}*

*}*

*List();*

*printf("\n####按学号升序排序完成####\n");*

*}*

5、转换时间 time.cpp

导入数据和保存数据成功后，获取时间戳并输出本地时间[2]。

*int time()*

*{*

*time\_t now=time(0);*

*tm \*ltm=localtime(&now);*

*printf("%02d.%02d.%02d %02d:%02d:%02d\n",1900+ltm->tm\_year,1+ltm->tm\_mon,ltm->tm\_mday,ltm->tm\_hour,ltm->tm\_min,ltm->tm\_sec);*

*return 0;*

*}*

6、程序运行截图

(1)开始



图5.6.1 导入数据、开启备份、显示当前操作时间和菜单预览

(2)数据处理

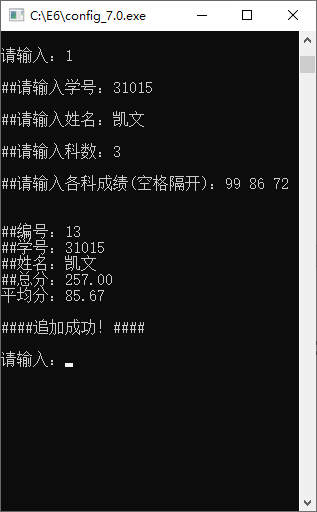


图5.6.2 追加记录

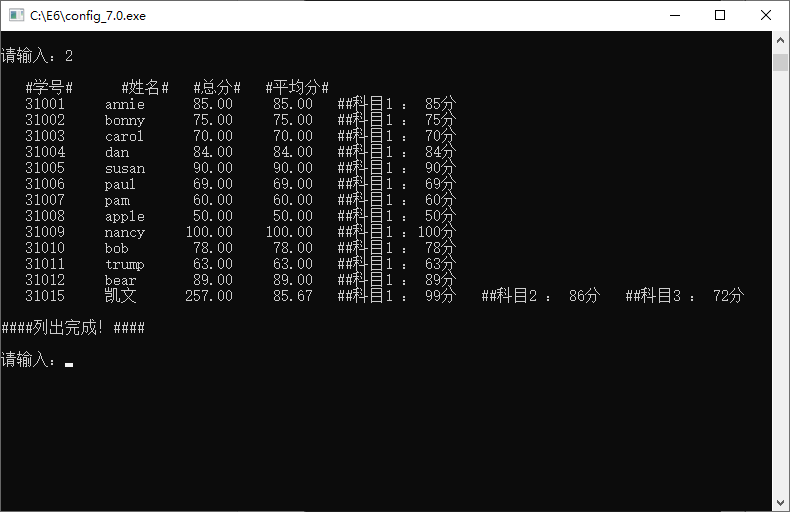


图5.6.3 列出记录

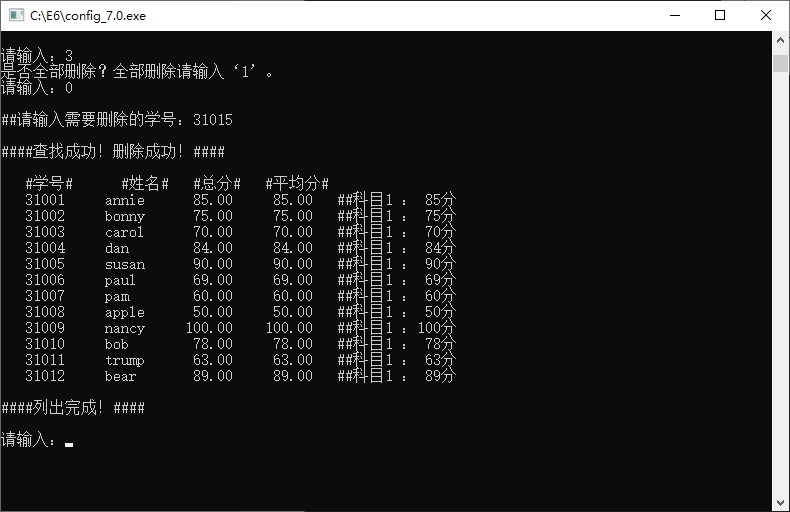


图5.6.4 删除记录

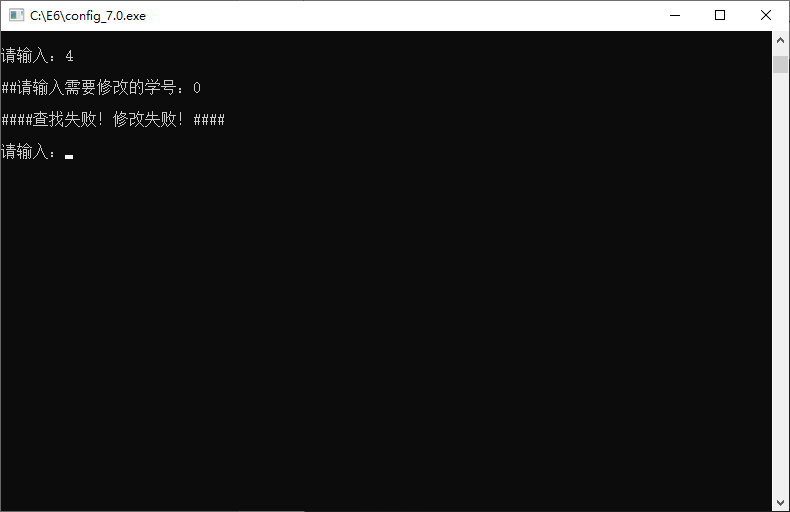


图5.6.5 查找记录——失败



图5.6.6 修改记录——成功

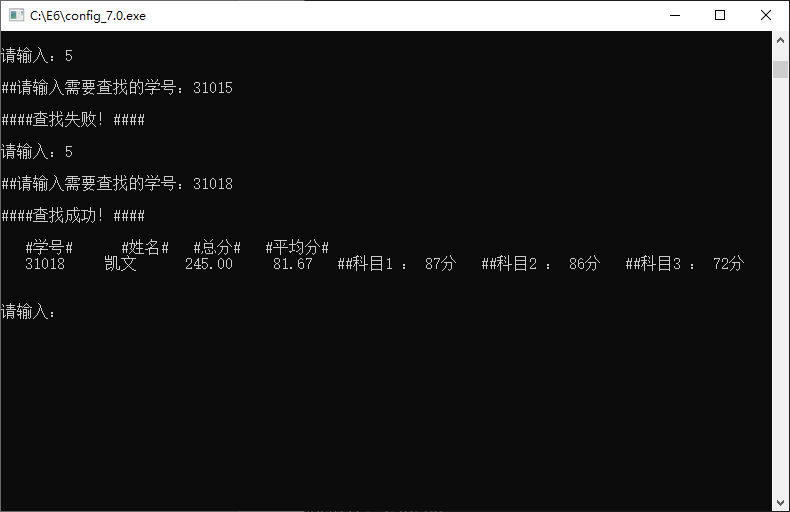


图5.6.6 查找记录（成功&失败）



图5.6.7 删除记录——一键清空

(3)成绩排序

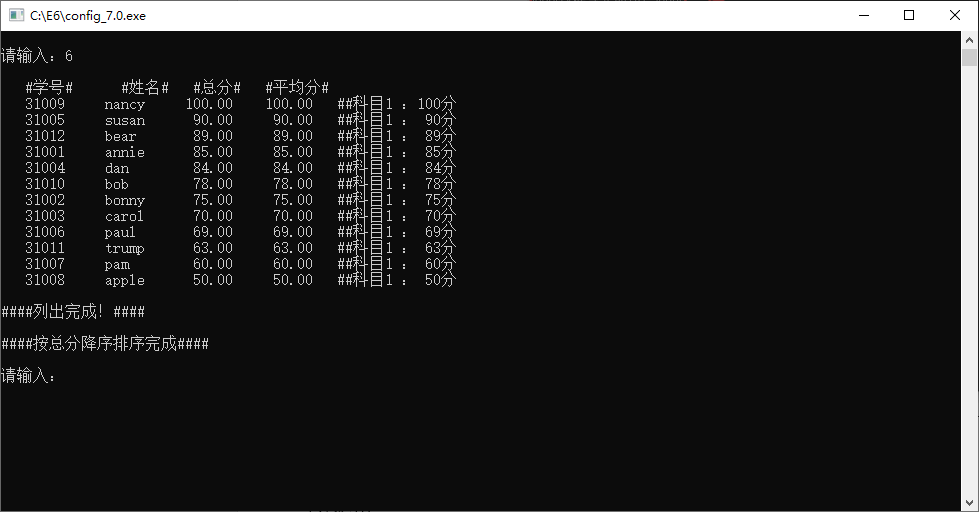
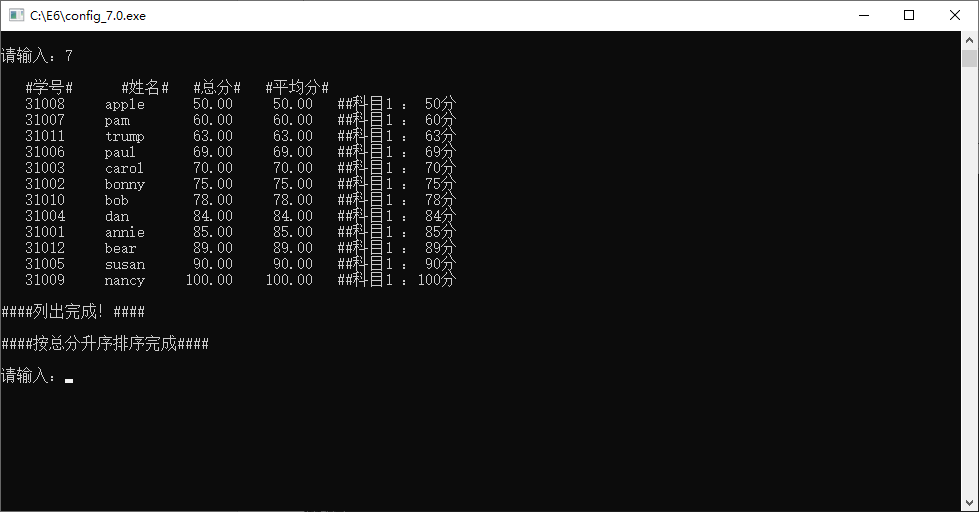
 

图5.6.8 按总分降序排序 图5.6.9 按总分升序排序

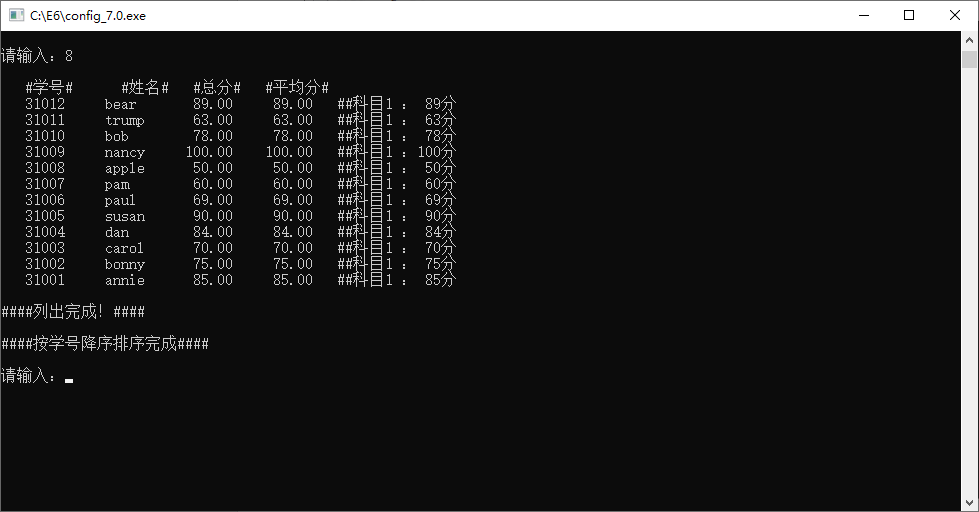
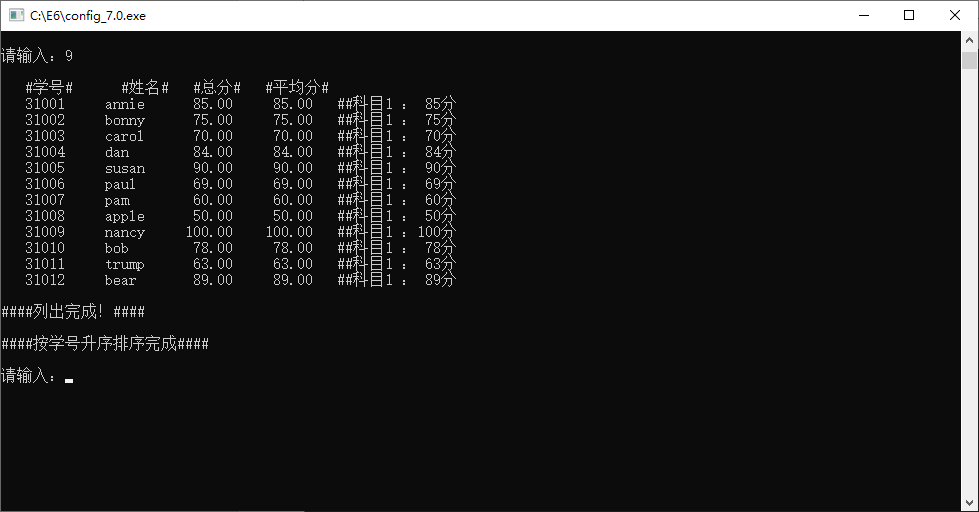
 

图5.6.10 按学号降序排序 图5.6.11 按学号升序排序

(4)退出

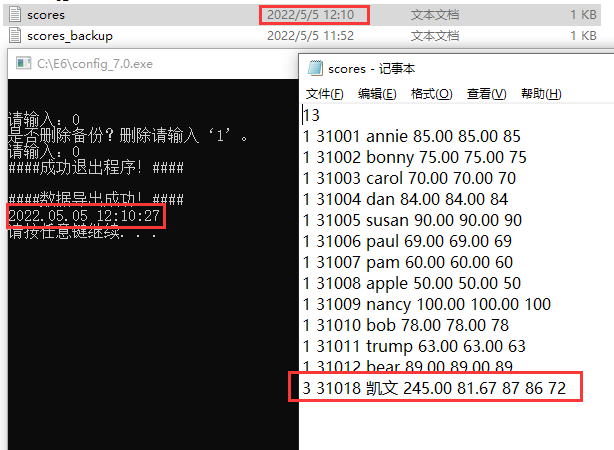


图5.6.12 退出程序、导出数据

**六：实验小结**

完成情况：基本完成实验学生成绩管理的要求，其中包括追加记录、列出记录、删除记录、修改记录、查找记录，分别根据总分和学号按降序、升序进行排序。同时对数据进行读取、保存等功能。并追加部分小功能，备份数据、显示时间、一键清空等。

需改进处： 增加科目名称的输入，备份文件名后缀时间，尝试导入备份文件。

难处： 大文件构成，多文件模块组合。数据与数据之间会相互影响，所以要注意数据的变化。

收获： 加强了C语言文件这一块薄弱的地方。

**七：参考资料**：

[1] C语言程序设计（第4版），何钦铭，颜晖主编。

[2] C++ 日期 & 时间，菜鸟教程，<https://www.runoob.com/cplusplus/cpp-date-time.html>。