



课 程 设 计 报 告

学院： 计算机与电子信息学院

专业： 信息安全

课程： 程序设计基础

任课教师： 张小萍

姓名： 陈 颖

学号： 1707170104

课题： 学生成绩管理系统

设计日期： 2018 年 11 月 1 日--2018 年 12 月 20 日

一. 题目与要求

设计一个简单的学生成绩管理系统，学生信息包括学号、姓名、成绩，要求实现如下功能：

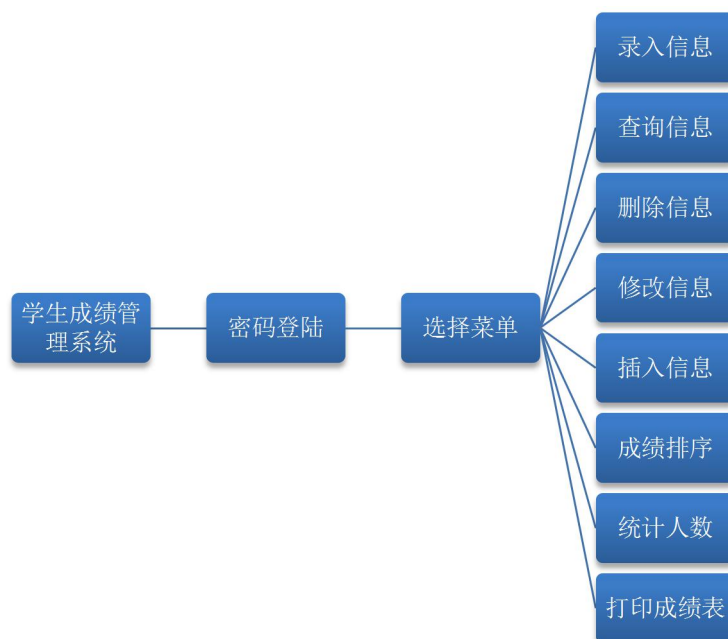
1. 录入功能：录入成绩到文件 `score.dat`，输入学生的学号、姓名、成绩，当输入学号为 0 时表示输入结束，输入的学号、姓名和成绩都存储到文件 `score.dat` 中。
2. 排序功能：按成绩从高到低的顺序从文件 `score.dat` 中在屏幕中打印出所有学生的信息。
3. 查询功能：给出学生学号，可以打印出该学生的所有信息，通过数字菜单选择所需功能，完成一个功能的显示又回到该数字菜单。

二. 设计目的

利用本学期程序设计基础课所学的 C 语言知识，依托本次课题任务，加深对数组、函数、循环结构、文件的保存等基本知识的理解，做到学以致用。同时，提高编码能力和算法设计能力。

三. 系统总体设计及说明

系统结构设计：



说明：在满足任务要求的基础上，本系统加入了删除、修改、插入、统计及打印成绩总表的功能，并在进入系统前设置了用密码登陆环节，使系统的功能更加齐全、完善，更符合用户要求，具有实用价值。

四. 功能描述

4.1 各项功能描述

4.1.1 密码登陆

假设用户密码已设定为“GXU19282018”，在进入该系统时，需要根据提示输入正确的密码才可以使用。输入错误时，提示“密码错误”，一共有三次输入机会，三次后，显示“NULLNULLNULL”，无法打开该系统使用。（为便于展示运行过程即结果，未将输入密码隐藏）

4.1.2 选择菜单

成功登陆后，出现菜单页，根据提示输入数字选择需要使用的功能项，输入对应数字进入对应功能进行下一步操作。

4.1.3 录入信息

输入数字 1，进入该项功能内页，根据提示，输入数字 1 表示继续操作，数字 0 表示结束操作，根据信息分项提示每完成一个学生的信息输入，出现“XX 同学信息保存成功！”提示，表示已将信息存入文件。

4.1.4 查询信息

输入数字 2，进入该项内页，根据提示，输入该学生的学号，若系统中存在此学生信息，则出现“已找到目标，是否显示信息？”提示，此时，输入“y”展示该生的所有成绩信息；若系统中不存在，则出现“error”。

4.1.5 删除信息

输入数字 3，进入该项内页，根据提示，输入所要删除的学生的学号，若系统中存在此学生信息，则出现“已找到目标，是否删除？”提示，此时，输入“y”删除该生的所有成绩信息；若系统中不存在，则出现“error”。

4.1.6 修改信息

输入数字 4，进入该项内页，根据提示，输入所要修改的学生的学号，若系统中存在此学生信息，则出现“已找到目标”提示，此时，开始修改该生的所有

成绩信息；若系统中不存在，则出现“error”。

4.1.7 插入信息

输入数字 5，进入该项内页，根据提示，输入所要插入的位置，随后，开始添加该生的所有成绩信息。

4.1.8 成绩排序

输入数字 6，进入该项内页，显示根据总分排序的成绩总表。

4.1.9 统计人数

输入数字 7，进入该项内页，显示当前系统中已经录入的学生人数。

4.1.10 打印成绩表

输入数字 8，进入该项内页，打印当前系统中所有已经录入的学生的所有成绩信息。

4.2 设计原理及流程图

4.2.1 总体

4.2.1.1 程序包含<stdio.h>, <conio.h>, <string.h>, <stdlib>, 四个头文件，其相关功能分别如下：

stdio.h: 在 C 或 C++中就是指 “standard input & output”，标准输入输出头文件，提供跨工程公共代码。

conio.h: conio.h 不是 C 标准库中的头文件，是 Console Input/Output (控制台输入输出)的简写，其中定义了通过控制台进行数据输入和数据输出的函数，主要是一些用户通过按键盘产生的对应操作，比如 getch() 函数等等。

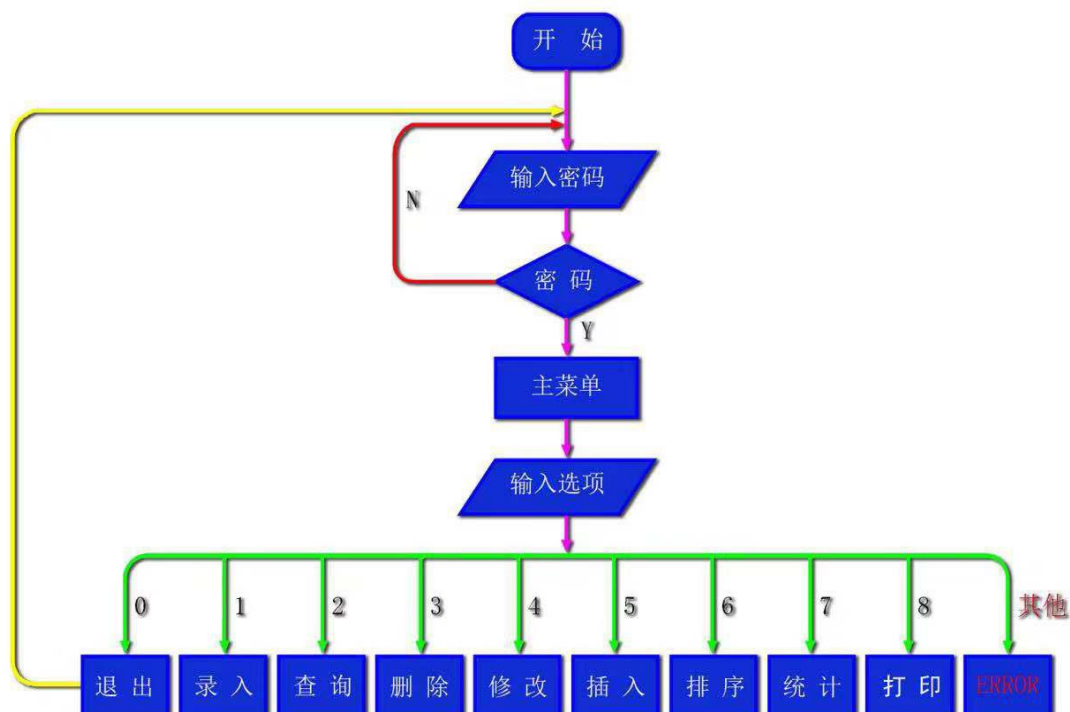
string.h: C 语言标准库中一个常用的头文件，在使用到字符数组时需要使用。常用函数有 strlen、strcmp、strcpy 等等。

stdlib.h: stdlib 头文件即 standard library 标准库头文件。stdlib.h 里面定义了五种类型、一些宏和通用工具函数。

4.2.1.2 定义关于学生学号、姓名、成绩（选修课、实验课、必修课、总成绩）信息的结构体，便于数据的整合与管理

4.2.1.3 引用文件类型指针，以“fopen” “wb” 打开（建立）二进制文件“score.dat”，以“fwrite” 指令将信息存入文件“score.dat”，以“fclose” 指令关闭文件。

4.2.1.4 简要总体流程图



4.2.2 各函数版块

4.2.2.1 密码登陆

利用 for 循环，实现对输入次数的控制。调用字符串比较函数 strcmp 判断密码是否正确。

4.2.2.2 菜单函数

上一步判断密码正确后，进入菜单。通过对打印格式的控制，输出菜单格式，并设计 switch 多项选择结构实现“多选一”，并进入不同的小功能函数项。

4.2.2.3 录入函数

在此处调用“fopen”、“wb”指令建立文件“score.dat”，以“fwrite”指令将学生信息写入磁盘。运用当型循环判断是否还要继续录入信息，并设置整型变量 m 作为计数变量，i 作为录入信息时的过程循环变量，将学生的信息逐项逐个输入，并保存到文件中。

4.2.2.4 打印函数

在此处调用“fopen”、“rb”打开文件“score.dat”，利用 for 循环打印显示相关信息。

4.2.2.5 查询函数

利用 for 循环语句对新输入的学号与文件中已有的学号进行逐一比对，利用 if 开关语句对比对结果进行判断，调用打印函数打印出所找到的学生全部信息。

4.2.2.6 删除函数

利用 for 循环，通过相邻数组元素替换，将删除目标之后的数组元素前移。之后再利用 for 循环，将新的成绩信息总表输入磁盘文件

4.2.2.7 修改函数

调用查询函数找到需要修改的学生信息并定位，之后利用 for 循环将新的成绩信息总表输入磁盘文件

4.2.2.8 插入函数

先定位需要插入在数组中的位置，与删除函数相反，通过相邻数组元素的替换，将插入之后的数组元素后移。之后再利用 for 循环将新的成绩信息总表输入到磁盘文件。

4.2.2.9 排序函数

利用经典算法“冒泡排序”，即双重 for 循环，对相邻两个元素（总分）进行大小比较，分高者在前，经过 $(m-1)$ 次 (m 为总人数) 比较，分数从高到低排列。

4.2.2.10 统计函数

在此处调用“fopen”、“rb”打开文件“score.dat”，设置计数变量 i ，利用 for 循环统计。

4.2.2.11 打印成绩总表

调用打印函数 show()。

五. 运行结果展示

5.1 登陆

登陆界面：

请输入密码:

输入错误，提示，3 次后登录失败

请输入密码:

11

密码错误，再次输入:

22

密码错误，再次输入:

33

NULLNULLNULL

输入正确的密码进入菜单页

请输入密码:

GXU19282018

5.2 菜单页

```
-----STUDENT-----
0. 退出系统 exit
1. 录入学生成绩信息input record
2. 查询学生信息search record
3. 删除学生成绩信息delete record
4. 修改生成成绩信息modify record
5. 插入记录生成成绩信息insert record
6. 成绩排序order
7. 统计录入总人数number
8. 打印成绩总表show
```

请选择您所需要的功能(输入数字0-8):

5.3 录入成绩

```
please input (1/0) (1为继续、0为结束录入):1
学号:101
姓名:Nini
选修课:92
实验课:95
必修课:97
Nini 同学的信息成功保存!
继续当前操作?(1/0) (1为继续、0为退出):1
学号:102
姓名:Nana
选修课:97
实验课:91
必修课:86
Nana 同学的信息成功保存!
继续当前操作?(1/0) (1为继续、0为退出):1
学号:103
姓名:Bob
选修课:77
实验课:85
必修课:67
Bob 同学的信息成功保存!
继续当前操作?(1/0) (1为继续、0为退出):1
学号:104
姓名:Dannie
选修课:99
实验课:63
必修课:87
Dannie 同学的信息成功保存!
```

以输入 0 结束

```
姓名:Ming
选修课:74
实验课:65
必修课:72
Ming 同学的信息成功保存!
继续当前操作?(1/0) (1为继续、0为退出):0
OK!
```

5.4 查询信息

输入学号，展示结果

```
请选择您所需要的功能(输入数字0-8):2
请输入学号:104
已找到目标，是否展示? (y/n)y
```

学号	姓名	选修课	实验课	必修课	总分
104	Dannie	99.0	63.0	87.0	249.0

5.5 删除信息

输入删除学号

学号	姓名	选修课	实验课	必修课	总分
101	Nini	92.0	95.0	97.0	284.0
102	Nana	97.0	91.0	86.0	274.0
103	Bob	77.0	85.0	67.0	229.0
104	Dannie	99.0	63.0	87.0	249.0
105	Ming	74.0	65.0	72.0	211.0

please input the number:104
find the student,delete?(y/n)y
删除成功!

5.6 修改信息

此时 104 号 Dannie 已经不在成绩表中，删除成功，输入修改学号

请选择您所需要的功能(输入数字0-8):4

学号	姓名	选修课	实验课	必修课	总分
101	Nini	92.0	95.0	97.0	284.0
102	Nana	97.0	91.0	86.0	274.0
103	Bob	77.0	85.0	67.0	229.0
105	Ming	74.0	65.0	72.0	211.0

请输入需要修改的学生信息的学号!
学号:102
已找到目标，开始修改!
姓名:Nana
选修课:97
实验课:91
必修课:96
修改成功!

5.7 插入信息

输入插入位置，添加

请选择您所需要的功能(输入数字0-8):5

请输入所需插入位置
7
现在，请输入新的信息
学号:108
姓名:Rose
选修课:91
实验课:77
必修课:89

5.8 统计人数

增加了 3 个学生信息项，有 8 人在系统中

请选择您所需要的功能(输入数字0-8):7

一共有 8 个学生!

5.9 打印成绩总表

请选择您所需要的功能(输入数字0-8):8					
学号	姓名	选修课	实验课	必修课	总分
101	Nini	92.0	95.0	97.0	284.0
102	Nana	97.0	91.0	96.0	284.0
108	Rose	91.0	77.0	89.0	257.0
106	Lily	85.0	86.0	79.0	250.0
107	Coco	87.0	69.0	92.0	248.0
103	Bob	77.0	85.0	67.0	229.0
105	Ming	74.0	65.0	72.0	211.0
104	Sam	51.0	77.0	65.0	193.0

如图，打印出系统中所有学生的所有信息，并已经排序

六. 系统设计的优缺点

6.1 优点

1. 加入密码登陆功能，对系统成绩信息有一定的加密性，使该系统更具实用性
2. 执行过程中多次调用了“clr”指令，清屏，使界面更加清晰、美观，可用性高，系统使用视觉效果好
3. 采用宏定义输出格式，简化后续代码，同时使输出格式更加规范、统一
4. 该系统采用汉化提示，可读性高

6.2 缺点

1. 该系统的程序算法设计主要利用数组，有一定的局限性，内存空间利用率有不必要浪费，时间复杂度高
2. 该系统不具有成绩统计（平均分、通过率等）功能，功能的多样化有待提高

参考文献

- [1]谭浩强.C 语言程序设计（第五版）[M]. 北京：清华大学出版社
- [2]谭浩强.C 语言程序设计（第五版）学习辅导[M]. 北京：清华大学出版社
- [3]明日科技. 零基础学C 语言[M]. 长春：吉林大学出版社
- [4]Stephen Prata in the United States,C++ Primer Plus[M]. 北京：人民邮电出版社
- [5]李春葆. 数据结构（第五版）[M]. 北京：清华大学出版社

附录：程序源代码

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>
#define LEN sizeof(struct student)
#define FORMAT "%-12d%-15s%-12.1lf%-12.1lf%-12.1lf%-12.1lf\n"
#define DATA stu[i].num,stu[i].name,stu[i].elec,stu[i].expe,stu[i].requ,stu[i].sum
```

struct student//学生成绩信息结构体

```
{ int num;//学号
  char name[15];//姓名
  double elec;//选修课
  double expe;//实验课
  double requ;//必修课
  double sum;//总分
};
```

struct student stu[50];/*定义结构体数组*/

```
void log();/*密码登陆*/
void input();/*录入学生成绩信息*/
void show();/*显示学生信息*/
void order();/*按总分排序*/
void del();/*删除学生成绩信息*/
void modify();/*修改学生成绩信息*/
void menu();/*菜单*/
void insert();/*插入学生信息*/
void total();/*统计总人数*/
void search();/*查找学生信息*/
```

```
void log();/*密码登陆*/
```

```

{
    int i;
    char password[20]={0};
    printf("请输入密码:\n");
    for(i=0;i<3;i++)
    {
        scanf("%s",password);
        if(i<2)
        {
            if(strcmp("GXU19282018",password)!=0)
            {
                do
                {
                    printf("密码错误, 再次输入:\n");
                }while(strcmp("GXU19282018",password)==0);
            }
            else
            if(strcmp("GXU19282018",password)==0)
            {
                printf("欢迎使用学生成绩管理系统!\n");
                printf("\n\n");
                menu(); break; /*进入菜单*/
            }
        }
        if(i==2)
            printf("NULLNULLNULL\n");
    }
}

```

```

void input()/*录入学生信息*/
{ int i,m=0; /*m 是记录的条数*/
  char ch[2];
  FILE *fp; /*定义文件指针*/
  if((fp=fopen("score.dat","a+"))==NULL)/*打开指定文件*/
  {
      printf("can not open\n");
      return;
  }
  while(!feof(fp))
  {
      if(fread(&stu[m],LEN,1,fp)==1)
          m++; /*统计当前记录条数*/
  }
}

```

```

fclose(fp);
if(m==0)
    printf("No record!\n");
else
{
    system("cls");
    show();/*调用 show 函数，显示原有信息*/
}

if((fp=fopen("score.dat","wb"))==NULL)
{
    printf("can not open\n");
    return;
}
for(i=0;i<m;i++)
{
    fwrite(&stu[i],LEN,1,fp);/*向指定的磁盘文件写入信息*/
}
system("cls");
printf("please input(1/0)(1 为继续、0 为结束录入):");
scanf("%s",ch);
while(strcmp(ch,"1")==0||strcmp(ch,"0")!=0)/*判断是否要录入新信息*/
{
    printf("学号:");
    scanf("%d",&stu[m].num);/*输入学生学号*/
    for(i=0;i<m;i++)
        if(stu[i].num==stu[m].num)
        {
            printf("该信息已经存在!");
            getch();
            fclose(fp);
            return;
        }

    printf("姓名:");
    scanf("%s",stu[m].name);/*输入学生姓名*/

    printf("选修课:");

    scanf("%lf",&stu[m].elec);/*输入选修课成绩*/
    printf("实验课:");
    scanf("%lf",&stu[m].expe);/*输入实验课成绩*/
    printf("必修课:");

```

```

scanf("%lf",&stu[m].requ);/*输入必修课成绩*/
stu[m].sum=stu[m].elec+stu[m].expe+stu[m].requ;/*计算出总成绩*/
if(fwrite(&stu[m],LEN,1,fp)!=1)/*将新录入的信息写入指定的磁盘文件*/
{
    printf("can not save!");
    getch();
}
else
{
    printf("%s 同学的信息成功保存!\n",stu[m].name);
    m++;
}
printf("继续当前操作?(1/0)(1 为继续、0 为退出):");/*询问是否继续*/
scanf("%s",ch);
}
fclose(fp);
printf("OK!\n");
}

```

```

void show()
{ FILE *fp;
  int i,m=0;
  fp=fopen("score.dat","rb");
  while(!feof(fp))
  {
    if(fread(&stu[m],LEN,1,fp)==1)
    m++;
  }
  fclose(fp);
  printf("学号      姓名      选修课      实验课      必修课      总分\t\n");
  for(i=0;i<m;i++)
  {
    printf(FORMAT,DATA);/*将信息按指定格式打印*/
  }
}

```

```

void menu()/*自定义函数实现菜单功能*/
{
    system("cls");
    printf("\n\n\n\n\n");
    printf("\t\t|-----STUDENT-----|\n");
}

```

```

printf("\t\t\t 0. 退出系统 exit                                \n");
printf("\t\t\t 1. 录入学生成绩信息 input record              \n");
printf("\t\t\t 2. 查询学生信息 search record                 \n");
printf("\t\t\t 3. 删除学生成绩信息 delete record              \n");
printf("\t\t\t 4. 修改成绩信息 modify record                  \n");
printf("\t\t\t 5. 插入记录成绩信息 insert record              \n");
printf("\t\t\t 6. 成绩排序 order                                \n");
printf("\t\t\t 7. 统计录入总人数 number                          \n");
printf("\t\t\t 8. 打印成绩总表 show                               \n");
printf("\t\t\t|-----|\n\n");
printf("\t\t\t 请选择您所需要的功能(输入数字 0-8):");
}

```

```

void order()/*自定义成绩排序 */
{ FILE *fp;
  struct student t;
  int i=0,j=0,m=0;
  if((fp=fopen("score.dat","r+"))==NULL)
  {
    printf("can not open!\n");
    return;
  }
  while(!feof(fp))
  if(fread(&stu[m] ,LEN,1,fp)==1)
    m++;
  fclose(fp);
  if(m==0)
  {
    printf("error!\n");
    return;
  }
  if((fp=fopen("score.dat","wb"))==NULL)
  {
    printf("can not open\n");
    return;}
  for(i=0;i<m-1;i++)
    for(j=i+1;j<m;j++)//利用“冒泡排序法”对成绩排序
      if(stu[i].sum<stu[j].sum)
        { t=stu[i];stu[i]=stu[j];stu[j]=t;}
  if((fp=fopen("score.dat","wb"))==NULL)
    { printf("can not open\n");return;}
  for(i=0;i<m;i++)/*将重新排好序的内容重新写入指定的磁盘文件中*/
    if(fwrite(&stu[i] ,LEN,1,fp)!=1)

```

```

        {
            printf("%s can not save!\n");
            getch();
        }
fclose(fp);
printf("保存成功\n");
}

```

void del()/*自定义删除函数*/

```

{
    system("cls");
    FILE *fp;
    int snum,i,j,m=0;
    char ch[2];
    if((fp=fopen("score.dat","r+"))==NULL)
    {
        printf("can not open\n");return;}
    while( !feof(fp) )
        if(fread(&stu[m],LEN,1,fp)==1) m++;
    fclose(fp);
    if(m==0)
    {
        printf("no record!\n");
        return;
    }
    show();
    printf("please input the number:");
    scanf("%d",&snum);

    for(i=0;i<m;i++)
    {
        if(snum==stu[i].num)
        {
            printf("find the student,delete?(y/n)");
            scanf("%s",ch);
            if(strcmp(ch,"Y")==0||strcmp(ch,"y")==0)/*判断是否要进行删除*/
            for(j=i;j<m;j++)
                stu[j]=stu[j+1];/*将后一个记录移到前一个记录的位置*/
            m--;/*记录的总个数减 1*/
            if((fp=fopen("score.dat","wb"))==NULL)
            { printf("can not open\n");return;}
            for(j=0;j<m;j++)/*将更改后的记录重新写入指定的磁盘文件中*/

```



```

        if(fwrite(&stu[j],LEN,1,fp)!=1)
        { printf("can not save!\n");
          getch();}
        fclose(fp);
        printf("删除成功!\n");
        return;
    }
}
printf("没有找到要删除的信息! \n");
}

```

```

void search()/*自定义查找函数*/
{ FILE *fp;
  int snum,i,m=0;
  char ch[2];
  if((fp=fopen("score.dat","rb"))==NULL)
  { printf("can not open\n");return;}
  while(!feof(fp)) if(fread(&stu[m],LEN,1,fp)==1) m++;
  fclose(fp);
  if(m==0) {printf("error!\n");return;}
  printf("请输入学号:");
  scanf("%d",&snum);
  for(i=0;i<m;i++)
  { if(snum==stu[i].num)/*查找输入的学号是否在记录中*/
    { printf("已找到目标, 是否展示? (y/n)");
      scanf("%s",ch);
      if(strcmp(ch,"Y")==0||strcmp(ch,"y")==0)
      {
        printf("学号      姓名      选修课      实验课      必修课      总分\n");
        printf(FORMAT,DATA);/*将查找出的结果按指定格式输出*/
        break;
      }
    }
  }
  if(i==m) printf("不存在!\n");/*未找到要查找的信息*/
}

```

```

void modify()/*自定义修改函数*/
{ FILE *fp;

```

```

int i,j,m=0,snum;
if((fp=fopen("score.dat","r+"))==NULL)
    { printf("can not open\n");return;}
while(!feof(fp))
if(fread(&stu[m],LEN,1,fp)==1) m++;
if(m==0)
{
    printf("no record!\n");
    fclose(fp);
    return;
}
show();
printf("请输入需要修改的学生信息的学号!\n");
printf("学号:");
scanf("%d",&snum);
for(i=0;i<m;i++)
{
    if(snum==stu[i].num)/*检索记录中是否有要修改的信息*/
    {
        printf("已找到目标，开始修改!\n");

        printf("姓名:");
        scanf("%s",stu[i].name);/*输入名字*/
        printf("选修课:");
        scanf("%lf",&stu[i].elec);/*输入选修课成绩*/
        printf("实验课:");
        scanf("%lf",&stu[i].expe);/*输入实验课成绩*/
        printf("必修课:");
        scanf("%lf",&stu[i].requ);/*输入必修课成绩*/
        printf("修改成功!");

        stu[i].sum=stu[i].elec+stu[i].expe+stu[i].requ;
        if((fp=fopen("score.dat","wb"))==NULL)
        { printf("can not open\n");return;}
        for(j=0;j<m;j++)/*将新修改的信息写入指定的磁盘文件中*/
        if(fwrite(&stu[j],LEN,1,fp)!=1)
            { printf("can not save!"); getch(); }
        fclose(fp);
        return ;
    }
}
printf("没有找到匹配信息! \n");
}

```

```

void insert()/*自定义插入函数*/
{ FILE *fp;
  int i,j,k,m=0,snum;
  if((fp=fopen("score.dat","r+"))==NULL)
    { printf("can not open\n");return;}
  while(!feof(fp))
  if(fread(&stu[m],LEN,1,fp)==1) m++;
  if(m==0) {printf("no record!\n");
  fclose(fp);
  return;
  }
  printf("请输入所需插入位置\n");
  scanf("%d",&snum);/*输入要插入的位置*/
  for(i=0;i<m;i++)
    if(snum==stu[i].num)
      break;
  for(j=m-1;j>i;j--)
    stu[j+1]=stu[j];/*从最后一条记录开始均向后移一位*/
  printf("现在, 请输入新的信息\n");
  printf("学号:");
  scanf("%d",&stu[i+1].num);
  for(k=0;k<m;k++)
    if(stu[k].num==stu[m].num)
    {
      printf("已存在!");
      getch();
      fclose(fp);
      return;
    }
  printf("姓名:");
  scanf("%s",stu[i+1].name);
  printf("选修课:");
  scanf("%lf",&stu[i+1].elec);
  printf("实验课:");
  scanf("%lf",&stu[i+1].expe);
  printf("必修课:");
  scanf("%lf",&stu[i+1].requ);
  stu[i+1].sum=stu[i+1].elec+stu[i+1].expe+stu[i+1].requ;
  if((fp=fopen("score.dat","wb"))==NULL)
  { printf("can not open\n");return;}
}

```

```

        for(k=0;k<=m;k++)
        if(fwrite(&stu[k],LEN,1,fp)!=1)/*将修改后的记录写入磁盘文件中*/
        { printf("can not save!"); getch(); }
fclose(fp);
}

```

```

void total()
{
    FILE *fp;
    int m=0;
    if((fp=fopen("score.dat","r+"))==NULL)
    {
        printf("can not open\n");
        return;
    }
    while(!feof(fp))
        if(fread(&stu[m],LEN,1,fp)==1)
            m++;/*统计记录个数即学生个数*/
    if(m==0)
    {
        printf("no record!\n");
        fclose(fp);
        return;
    }
    printf("一共有 %d 个学生!\n",m);/*将统计的个数输出*/
    fclose(fp);
}

```

```

int main()/*主函数*/
{
    log();
    int n;
    scanf("%d",&n);/*输入选择功能的编号*/
    while(n)
    {
        switch(n)
        { case 1: input();break;
          case 2: search();break;
          case 3: del();break;
          case 4: modify();break;

```

```
        case 5: insert();break;
        case 6: order();break;
        case 7: total();break;
        case 8: show();break;
        default:break;
    }
    getch();
    menu();/*执行完功能再次显示菜单界面*/
    scanf("%d",&n);
}

return 0;
}
```