**《程序设计基础实训》设计报告**

**设计题目**

**学生姓名：李堂烽**

**学 号：202058304066**

**班 级：计算机类2班**

**同组成员：周方未**

**日 期: 2021.06.01**

目录

**[1 题目与要求](#_Toc74992703)** [3](#_Toc74992703)

**[1.1问题提出](#_Toc74992704)** [3](#_Toc74992704)

**[1.2 本系统涉及的知识点](#_Toc74992705)** [3](#_Toc74992705)

**[1.3 功能要求](#_Toc74992706)** [3](#_Toc74992706)

**[2 功能设计](#_Toc74992707)** [3](#_Toc74992707)

**[2.1 主函数模块图](#_Toc74992708)** [3](#_Toc74992708)

**[.2 部分模块流程图](#_Toc74992709)** [4](#_Toc74992709)

**[3 程序代码设计](#_Toc74992710)** [5](#_Toc74992710)

**[3.1主函数模块](#_Toc74992711)** [6](#_Toc74992711)

**[3.2查询函数](#_Toc74992712)** [7](#_Toc74992712)

**[3.3添加学生信息模块](#_Toc74992713)** [9](#_Toc74992713)

**[3.4删除函数](#_Toc74992714)** [10](#_Toc74992714)

**[3.5班级情况函数](#_Toc74992715)** [12](#_Toc74992715)

**[3.6年级情况模块](#_Toc74992716)** [15](#_Toc74992716)

**[3.7备份函数模块](#_Toc74992717)** [16](#_Toc74992717)

**[3.8退出函数模块](#_Toc74992718)** [17](#_Toc74992718)

**[3.9修改密码函数模块](#_Toc74992719)** [17](#_Toc74992719)

**[3.10挂科函数模块](#_Toc74992720)** [18](#_Toc74992720)

**[3.11匹配函数模块](#_Toc74992721)** [19](#_Toc74992721)

**[3.12排序函数模块](#_Toc74992722)** [20](#_Toc74992722)

**[3.13密码确定函数模块](#_Toc74992723)** [22](#_Toc74992723)

**[3.14修改密码函数模块](#_Toc74992725)** [23](#_Toc74992725)

**[3.15密码写入文件函数模块](#_Toc74992726)** [23](#_Toc74992726)

**[3.16备份函数模块](#_Toc74992727)** [23](#_Toc74992727)

**[3.17写入文件函数模块](#_Toc74992728)** [24](#_Toc74992728)

**[3.18总成绩计算函数模块](#_Toc74992729)** [25](#_Toc74992729)

**[3.19输出函数模块](#_Toc74992730)** [25](#_Toc74992730)

**[3.20分配到班级函数模块](#_Toc74992731)** [26](#_Toc74992731)

**[3.21各种重复利用的输出函数模块](#_Toc74992732)** [27](#_Toc74992732)

**[4 总结](#_Toc74992733)** [27](#_Toc74992733)

**[4.1程序调试情况](#_Toc74992734)** [27](#_Toc74992734)

**[4.2本人在程序设计中感想](#_Toc74992735)** [37](#_Toc74992735)

**1 题目与要求**

**1.1问题提出**

建立一个顺序表，顺序表数组包括学生信息，姓名，学号，成绩等，然后对信息进行处理，算出总分，算出各科平均分，得到排序表。也可以把信息分配到班级里，从而得到各个班级的情况，得知一个班级的最高分，最低分，个科目平均分等。考虑到操作的简便性，条理性，方便性，所以使用while（）函数建立功能菜单，方便使用。为了美观，可以使用system（“color xx”）函数改变字体颜色和背景板颜色，还有可以使用GetStdHandle(STD\_OUTPUT\_HANDLE)函数获取句柄，改变颜色;美化界面。还可以考虑到信息安全，建立安全模块，使用while（）函数循环的跳出条件设置密码，可以模拟密码功能，只有输出正确的密码才能跳出while（）函数循环，进入程序。

**1.2 本系统涉及的知识点**

结构、数组、循环、函数、分支、指针、顺序表、

**1.3 功能要求**

1、建立学生信息顺序表，包括：学号、班级、姓名、性别、专业、语文成绩、数学成绩、外语成绩、语文平均成绩

2、算出语文平均成绩，数学平均成绩，外语平均成绩，总成绩；

3、排序：分别按总分排序、按学号排序；

4、根据学号，姓名查询成绩；

5、输出各个科目挂科名单，各级科目平均成绩，最高成绩，最低成绩等；

6、修改学生的各个信心；

7、删除某个学生信息；

8、备份文件；

9、从为文件中读取；

10、设置密码，修改密码

**2 功能设计**

**2.1 主函数模块图**

**开始**

**打开文件**

**从文件读数据**

**关闭文件**

**显示系统菜单**

**响应用户选择**

**各功能处理**

**打开文件**

**是否退出**

**Y**

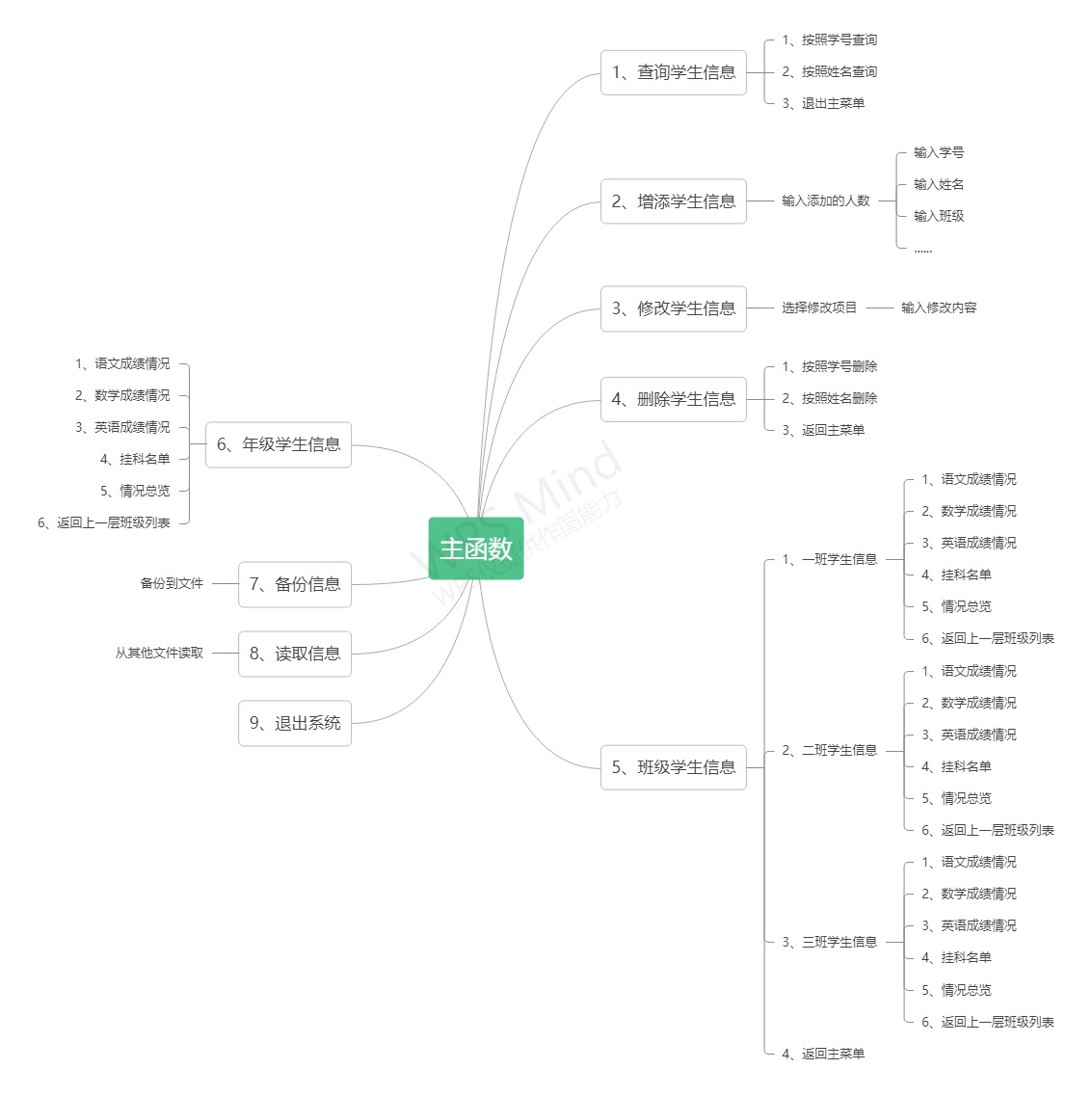
**N**

**结束**

**关闭文件**

**将数据写入相应文件**

**.2 部分模块流程图**

****

**3 程序代码设计**

**3.1主函数模块**

1. 函数原形：

int main()

{

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE); //获取输出设备的句柄

CONSOLE\_SCREEN\_BUFFER\_INFO csbiInfo;

GetConsoleScreenBufferInfo(handle, &csbiInfo); //获取窗口缓冲区信息

WORD wOldColorAttrs = csbiInfo.wAttributes; //获取初始定义的文本属性，即默认属性

int flog;

char key[20];

list peo, peo1, peo2, peo3;

get\_key\_func(key);

key\_show(key);

welcome\_show();

read\_func(peo);

sort\_func(peo, 5);

all\_score\_func(peo);

distribute\_func(peo, peo1, peo2, peo3);

system("cls");

while (1)

{

system("color B4");

SetConsoleTextAttribute(handle, BACKGROUND\_RED);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*欢迎进入第二学期学生信息管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_RED | BACKGROUND\_GREEN | BACKGROUND\_INTENSITY);

printf("-------------- 1、查询学生信息 ----------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_RED | BACKGROUND\_BLUE | BACKGROUND\_INTENSITY);

printf("-------------- 2、增添学生信息 ----------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_GREEN | BACKGROUND\_BLUE | BACKGROUND\_INTENSITY);

printf("-------------- 3、修改学生信息 ----------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_GREEN | BACKGROUND\_RED | BACKGROUND\_BLUE | BACKGROUND\_INTENSITY);

printf("-------------- 4、删除学生信息 ----------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_RED | BACKGROUND\_GREEN | BACKGROUND\_INTENSITY);

printf("-------------- 5、班级学生信息 ----------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_RED | BACKGROUND\_BLUE | BACKGROUND\_INTENSITY);

printf("-------------- 6、年级学生信息 ----------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_GREEN | BACKGROUND\_BLUE | BACKGROUND\_INTENSITY);

printf("-------------- 7、备份信息 --------------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_GREEN | BACKGROUND\_RED | BACKGROUND\_BLUE | BACKGROUND\_INTENSITY);

printf("-------------- 8、读取信息 --------------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_RED | BACKGROUND\_GREEN | BACKGROUND\_INTENSITY);

printf("-------------- 9、退出系统 --------------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_RED | BACKGROUND\_BLUE | BACKGROUND\_INTENSITY);

printf("---------------10、修改密码--------------\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

SetConsoleTextAttribute(handle, BACKGROUND\_RED | BACKGROUND\_INTENSITY);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

SetConsoleTextAttribute(handle, wOldColorAttrs);

b\_func();

scanf("%d", &flog);

// fflush(stdin);

system("cls");

switch (flog)

{

case 1:

inquire\_func(peo); break;

case 2:

add\_func(peo, peo1, peo2, peo3); break;

case 3:

change\_func(peo, peo1, peo2, peo3); break;

case 4:

delete\_func(peo, peo1, peo2, peo3); break;

case 5:

class\_func(peo, peo1, peo2, peo3); break;

case 6:

grade\_func(peo);

break;

case 7:

backups\_func(peo);

break;

case 8:

read\_other\_func(peo, peo1, peo2, peo3); break;

case 9:

quit\_func(peo);

break;

case 10:

change\_key\_func(key);

break;

default:

error\_func();

break;

}

if (flog == 9)

break;

}

system("color E4");

printf("感谢使用此系统，按任意键以退出\n");

getch();

}

2）功能：主菜单

3）变量：

int flog：作为主菜单switch（）函数的参数

char key[20]：存储密码

list peo：定义一个全体学生信息顺序表

List peo1：定义一个一班学生信息顺序表

List peo2：定义一个二班学生信息顺序表

List peo3：定义一个三班学生信息顺序表

**3.2查询函数**

1）函数原形：//查询函数

void inquire\_func(list& peo)

{

int flog, flog1;

int num, i;

char name[15];

while (1)

{

printf("选择查询方式\n");

printf(" 1、按照学号查询\n");

printf(" 2、按照线姓名查询\n");

printf(" 3、返回主菜单\n");

b\_func();

scanf("%d", &flog1);

switch (flog1)

{

case 1:

printf("输入学号：");

scanf("%d", &num);

flog = match1\_func(peo, num);

if (flog != -1)

a\_output\_func(peo, flog);

else

printf("无此人，查询失败\n");

break;

case 2:

printf("输入姓名：");

scanf("%s", name);

flog = match2\_func(peo, name);

if (flog != -1)

a\_output\_func(peo, flog);

else

printf("无此人，查询失败\n");

break;

case 3:

break;

default:

error\_func();

break;

}

if (flog1 != 3)

{

printf("按任意键以继续查询......");

getch();

system("cls");

}

if (flog1 == 3)

{

system("cls");

break;

}

}

}

2）功能：利用while循环创建循环菜单，选择查询学生信息的方式

3）变量及类型：

list& peo ：总学生信息结构体数组，将学生信息传输进来。

int num ：输出匹配的学号。

Int name ：输入需要匹配的姓名

int flog1 ：选择匹配方式。

int flog ： 接收match函数返回的值。

**3.3添加学生信息模块**

1）函数原形：//添加函数

void add\_func(list& peo, list& peo1, list& peo2, list& peo3)

{

int n, i, flog;

char c;

printf("输入添加学生的人数：");

scanf("%d", &n);

while ((c = getchar()) != '\n' && c != EOF);

for (i = 0, flog = peo.length; i < n; i++, flog++)

{

printf("学号:");

scanf("%d" ,&peo.all\_message[flog].num);

printf("班级:");

scanf("%d", &peo.all\_message[flog].cla);

printf("姓名:");

scanf("%s", peo.all\_message[flog].name);

printf("性别:");

scanf("%s", peo.all\_message[flog].sex);

printf("专业:");

scanf("%s", peo.all\_message[flog].profession);

printf("语文成绩:");

scanf("%f", &peo.all\_message[flog].chinese\_score);

printf("数学成绩:");

scanf("%f", &peo.all\_message[flog].math\_score);

printf("英语成绩:");

scanf("%f", &peo.all\_message[flog].english\_score);

if (n != 1)

system("cls");

}

while ((c = getchar()) != '\n' && c != EOF);

peo.length += n;

all\_score\_func(peo);

sort\_func(peo, 5);

distribute\_func(peo, peo1, peo2, peo3);

printf("添加%d个人成功，按任意键以返回主菜单...", n);

getch();

system("cls");

}

2）功能：添加任意个数的学生信息。

3）变量：

list& peo ：总学生信息结构体数组，将学生信息传输进来。

list& peo1：一班学生信息结构体数组，将一班学生信息传输进来。

list& peo2：二班学生信息结构体数组，将二班学生信息传输进来。

list& peo3：二班学生信息结构体数组，将二班学生信息传输进来。

int n ：作为添加学生的个数的参数

Int i ： 辅助添n个学生，作为循环参数

Char c ：接收可能多余的缓冲区符号，防止读取出错。

**3.4删除函数**

1）函数原形：//删除函数

void delete\_func(list& peo, list& peo1, list& peo2, list& peo3)

{

int i, num, flog, flog1;

char name[15];

while (1)

{

printf("选择删除方式\n");

printf(" 1、学号匹配删除\n");

printf(" 2、姓名匹配删除\n");

printf(" 3、返回主菜单\n");

printf("你的选择为：");

scanf("%d", &flog1);

switch (flog1)

{

case 1:

printf("输入学号：");

scanf("%d", &num);

flog = match1\_func(peo, num);

if (flog != -1)

{

for (i = flog; i < peo.length; i++)

{

peo.all\_message[i] = peo.all\_message[i + 1];

}

peo.length--;

distribute\_func(peo, peo1, peo2, peo3);

sort\_func(peo, 5);

printf("删除成功,");

printf("按任意键以继续......");

}

else

{

printf("删除失败，查无此人\n");

printf("按任意键以继续......");

}

getch();

system("cls");

break;

case 2:

printf("输入姓名：");

scanf("%s", name);

flog = match2\_func(peo, name);

if (flog != -1)

{

for (i = flog; i < peo.length; i++)

{

peo.all\_message[i] = peo.all\_message[i + 1];

}

peo.length--;

distribute\_func(peo, peo1, peo2, peo3);

sort\_func(peo, 5);

printf("删除成功,");

printf("按任意键以继续......");

}

else

{

printf("删除失败，查无此人\n");

printf("按任意键以继续......");

}

getch();

system("cls");

break;

case 3:

break;

}

if (flog1 == 3)

{

system("cls");

break;

}

}

}

2）功能：删除某个学生信息。

3）变量及类型：

list& peo ：总学生信息结构体数组，将学生信息传输进来。

list& peo1：一班学生信息结构体数组，将一班学生信息传输进来。

list& peo2：二班学生信息结构体数组，将二班学生信息传输进来。

list& peo3：二班学生信息结构体数组，将二班学生信息传输进来。

int n ：作为添加学生的个数的参数

Int i ： 辅助添n个学生，作为循环参数

Char c ：接收可能多余的缓冲区符号，防止读取出错。

**3.5班级情况函数**

1）函数原形：//班级函数

void class\_func(list& peo, list& peo1, list& peo2, list& peo3)

{

void class1\_func(list & all, list & peo, int cla);

int cla;

while (1)

{

printf("选择查看的班级\n");

printf(" 1.一班成绩情况\n");

printf(" 2.二班成绩情况\n");

printf(" 3.三班成绩情况\n");

printf(" 4.返回主菜单\n");

b\_func();

scanf("%d", &cla);

switch (cla)

{

case 1:

class1\_func(peo,peo1, cla);

break;

case 2:

class1\_func(peo,peo2, cla);

break;

case 3:

class1\_func(peo,peo3, cla);

break;

case 4:

return\_func();

break;

default:

error\_func();

break;

}

if (cla == 4)

break;

}

}

//班级副菜单

void class1\_func(list&all, list&peo, int cla)

{

float ave\_chinese, ave\_math, ave\_english;

float count\_chinese = 0, count\_math = 0, count\_english = 0;

int i,n;

for (i = 0; i < peo.length; i++)

{

count\_chinese += peo.all\_message[i].chinese\_score;

count\_math += peo.all\_message[i].math\_score;

count\_english += peo.all\_message[i].english\_score;

}

ave\_chinese = count\_chinese / peo.length;

ave\_math = count\_math / peo.length;

ave\_english = count\_english / peo.length;

system("cls");

while (1)

{

printf("%d班成绩情况\n",cla);

printf(" 1、语文成绩情况:\n");

printf(" 2、数学成绩情况:\n");

printf(" 3、英语成绩情况:\n");

printf(" 4、挂科名单:\n");

printf(" 5、情况总览:\n");

printf(" 6、返回上一层班级列表\n");

printf("你的选择:");

scanf("%d", &n);

system("cls");

switch (n)

{

case 1:

printf(" %d班语文平均成绩:%-8.2f最高语文成绩:%-8.2f最低语文成绩:%-8.2f\n", cla, ave\_chinese, peo.all\_message[0].chinese\_score, peo.all\_message[peo.length - 1].chinese\_score);

break;

case 2:

printf(" %d班数学平均成绩:%-8.2f最高数学成绩:%-8.2f最低数学成绩:%-8.2f\n", cla, ave\_math, peo.all\_message[0].math\_score, peo.all\_message[peo.length - 1].math\_score);

break;

case 3:

printf(" %d班英语平均成绩:%-8.2f最高英语成绩:%-8.2f最低英语成绩:%-8.2f\n", cla, ave\_english, peo.all\_message[0].english\_score, peo.all\_message[peo.length - 1].english\_score);

break;

case 4:

do\_not\_pass\_func(peo, cla, 2); break;

case 5:

class\_condition\_func(peo,cla);

case 6:

break;

default:

error\_func();

continue;

}

if (n != 6)

{

printf("按任意键以继续......");

getch();

system("cls");

}

if (n == 6)

break;

}

}

2）功能：查看班级学生成绩情况。

3）变量及类型：

list& peo ：总学生信息结构体数组，将学生信息传输进来。

list& peo1：一班学生信息结构体数组，将一班学生信息传输进来。

list& peo2：二班学生信息结构体数组，将二班学生信息传输进来。

list& peo3：二班学生信息结构体数组，将二班学生信息传输进来。

int n ：作为添加学生的个数的参数

Int i ： 辅助添n个学生，作为循环参数

Char c ：接收可能多余的缓冲区符号，防止读取出错。

int cla ：代表某个班级。

float ave\_chinese：某个班级的平均语文成绩

Float ave\_math：某个班级平均数徐成绩

Float ave\_english 某个班级平均英语成绩

float count\_chinese ：某个班级总成绩

Float count\_math ： 某个班数学总成绩

Float count\_english ：某个班英语总成绩

Int n ：作为switch（）的参数

**3.6年级情况模块**

1）函数原形：//年级函数

void grade\_func(list& peo)

{

int i;

float ave\_chinese, ave\_math,ave\_english;

float count\_chinese=0, count\_math=0, count\_english=0;

for (i = 0; i < peo.length; i++)

{

count\_chinese += peo.all\_message[i].chinese\_score;

count\_math += peo.all\_message[i].math\_score;

count\_english += peo.all\_message[i].english\_score;

}

ave\_chinese = count\_chinese / peo.length;

ave\_math = count\_math / peo.length;

ave\_english = count\_english / peo.length;

printf("一、年级成绩情况:\n");

printf(" 1.年级语文平均成绩:%-8.2f最高语文成绩:%-8.2f最低语文成绩:%-8.2f\n", ave\_chinese, peo.all\_message[0].chinese\_score, peo.all\_message[peo.length - 1].chinese\_score);

printf(" 2.年级数学平均成绩:%-8.2f最高数学成绩:%-8.2f最低数学成绩:%-8.2f\n", ave\_math, peo.all\_message[0].chinese\_score, peo.all\_message[peo.length - 1].chinese\_score );

printf(" 3.年级英语平均成绩:%-8.2f最高英语成绩:%-8.2f最低英语成绩:%-8.2f\n", ave\_english, peo.all\_message[0].english\_score, peo.all\_message[peo.length-1].english\_score);

a\_func();

printf("\n\n");

printf("二、年级挂科名单:\n");

do\_not\_pass\_func(peo, 0, 1);

a\_func();

printf("\n\n");

printf("三、年级成绩详细情况:\n");

for (i = 0; i < peo.length; i++)

a\_output\_func(peo,i);

a\_func();

return\_func();

}

2）功能：查看年级成绩情况

3）变量及类型：

list& peo ：总学生信息结构体数组，将学生信息传输进来。

float ave\_chinese：年级的平均语文成绩

Float ave\_math：年级平均数学成绩

Float ave\_english 年级平均英语成绩

float count\_chinese ：年级总成绩

Float count\_math ： 年级数学总成绩

Float count\_english ：年级英语总成绩

**3.7备份函数模块**

1）函数原形：//备份到其他文件

void read\_other\_func(list& peo, list& peo1, list& peo2, list& peo3)

{

FILE\* fp;

char str[1000];

int N = 1000;

int i, flog = 0;

char filename[20];

printf("输入读取的文件路径与文件名:");

scanf("%s", filename);

if ((fp = fopen(filename, "r")) == NULL)

{

printf("学生信息文件打开失败\n");

exit(0);

}

fgets(str, N, fp);

while (fscanf(fp, "%d%d%s%s%s%f%f%f%f", &peo.all\_message[flog].num, &peo.all\_message[flog].cla, peo.all\_message[flog].name, peo.all\_message[flog].sex, peo.all\_message[flog].profession, &peo.all\_message[flog].chinese\_score, &peo.all\_message[flog].math\_score, &peo.all\_message[flog].english\_score, &peo.all\_message[flog].all\_score) != EOF)

flog++;

peo.length = flog;

fclose(fp);

sort\_func(peo, 5);

distribute\_func(peo, peo1, peo2, peo3);

printf("读取成功......\n");

Sleep(2000);

system("cls");

}

2）功能：将内存中的学生信息备份到制定路径

3）变量：

list& peo ：总学生信息结构体数组，将学生信息传输进来。

list& peo1：一班学生信息结构体数组，将一班学生信息传输进来。

list& peo2：二班学生信息结构体数组，将二班学生信息传输进来。

list& peo3：二班学生信息结构体数组，将二班学生信息传输进来。

FILE\* fp：文件指针，与文件关联起来。

char filename[20]：备份文件的文件名。

**3.8退出函数模块**

1)函数原形：//退出函数

void quit\_func(list& peo)

{

cover\_func(peo);

}

2）功能：退出程序

**3.9修改密码函数模块**

1）函数原形：//修改密码函数

void change\_key\_func(char\*key)

{

char key1[20];

printf("密码修改为:");

scanf("%s",key1);

strcpy(key, key1);

put\_key\_func(key);

printf("修改成功......");

Sleep(1000);

system("cls");

}

2）功能：修改密码

3）变量：

char\*key：传入密码key

char key1[20]：接收输入的修改密码

**3.10挂科函数模块**

1）函数原形：//挂科函数

//挂科函数

void do\_not\_pass\_func(list& peo, int cla, int flog)

{

int i,n1=0,n2=0,n3=0;

if (flog == 1)

{//年级挂科

printf(" 1.年级语文挂科名单\n");

for (i = 0; i < peo.length; i++)

{

if (peo.all\_message[i].chinese\_score < 60)

{

n1++;

a\_output\_func(peo, i);

}

}

if (n1 == 0)

printf(" 语文无挂科人员\n");

c\_func();

printf("\n 2.年级数学挂科名单:\n");

for (i = 0; i < peo.length; i++)

{

if (peo.all\_message[i].math\_score < 60)

{

n2++;

a\_output\_func(peo, i);

}

}

if (n2 == 0)

printf(" 数学无挂科人员:\n");

c\_func();

printf("\n 3.年级英语挂科名单\n");

for (i = 0; i < peo.length; i++)

{

if (peo.all\_message[i].english\_score < 60)

{

n3++;

a\_output\_func(peo, i);

}

}

if (n3 == 0)

printf(" 英语无挂科人员:\n");

}

//班级挂科

else if (flog == 2)

{

printf(" %d班语文挂科名单:\n", cla);

for (i = 0; i < peo.length; i++)

{

if (peo.all\_message[i].chinese\_score < 60)

a\_output\_func(peo, i);

}

a\_func();

printf("\n %d班数学挂科名单:\n", cla);

for (i = 0; i < peo.length; i++)

{

if (peo.all\_message[i].math\_score < 60)

a\_output\_func(peo, i);

}

a\_func();

printf("\n %d班英语挂科名单:\n", cla);

for (i = 0; i < peo.length; i++)

{

if (peo.all\_message[i].english\_score < 60)

a\_output\_func(peo, i);

}

a\_func();

putchar(10);

}

}

2）功能：输出挂科名单

3）变量：

list& peo：传入学生信息顺序表

int cla：班级

int flog：作为输出年级挂科函数与班级挂科函数的判定参数

**3.11匹配函数模块**

1）函数原形：//匹配函数

/匹配函数

int match1\_func(list& peo, int num)

{//学号匹配

int i;

for (i = 0; i < peo.length; i++)

{

if (peo.all\_message[i].num == num)

return i;

}

return -1;

}

int match2\_func(list peo, char\* name)

{//姓名匹配

int i;

for (i = 0; i < peo.length; i++)

{

if (strcmp(peo.all\_message[i].name, name) == 0)//应该有问题 已经改

return i;

}

return -1;

}

2）功能：以学号或者姓名进行匹配，成功则返回进行下标，失败返回-1；

3）变量：

list& peo：传入总体学生信息顺序表

int num ：接收输入的需要匹配的学号

char\* name：接收输入的需要匹配的姓名

**3.12排序函数模块**

1）函数原形：//排序函数

void sort\_func(list& peo, int flog)

{

FILE\* fp;

int i, j;

stu tem;

// 1、按学号排序

if (flog == 1)

{

for (i = 1; i < peo.length ; i++)

{

for (j = 0; j < peo.length - i; j++)

if (peo.all\_message[j].num > peo.all\_message[j + 1].num)

{

tem = peo.all\_message[j];

peo.all\_message[j] = peo.all\_message[j + 1];

peo.all\_message[j + 1] = tem;

}

}

}

// 2、按语文成绩排序

if (flog == 2)

{

for (i = 1; i < peo.length ; i++)

{

for (j = 0; j < peo.length - i; j++)

if (peo.all\_message[j].chinese\_score > peo.all\_message[j + 1].chinese\_score)

{

tem = peo.all\_message[j];

peo.all\_message[j] = peo.all\_message[j + 1];

peo.all\_message[j + 1] = tem;

}

}

}

// 3、按数学成绩排序

if (flog == 3)

{

for (i = 1; i < peo.length ; i++)

{

for (j = 0; j < peo.length - i; j++)

if (peo.all\_message[j].math\_score < peo.all\_message[j + 1].math\_score)

{

tem = peo.all\_message[j];

peo.all\_message[j] = peo.all\_message[j + 1];

peo.all\_message[j + 1] = tem;

}

}

}

// 4、按英语成绩排序

if (flog == 4)

{

for (i = 1; i < peo.length ; i++)

{

for (j = 0; j < peo.length - i; j++)

if (peo.all\_message[j].english\_score < peo.all\_message[j + 1].english\_score)

{

tem = peo.all\_message[j];

peo.all\_message[j] = peo.all\_message[j + 1];

peo.all\_message[j + 1] = tem;

}

}

}

//5、按总成绩排序

if (flog == 5)

{

for (i = 1; i < peo.length; i++)

{

for (j = 0; j < peo.length - i; j++)

if (peo.all\_message[j].all\_score < peo.all\_message[j + 1].all\_score)

{

tem = peo.all\_message[j];

peo.all\_message[j] = peo.all\_message[j + 1];

peo.all\_message[j + 1] = tem;

}

}

}

}

2）功能：按照学号，语文成绩，数学成绩，英语成绩，总成绩进行排序，使顺序表数组有序。

3）变量：

list& peo：传输学生信息数组。

int flog：以某种方式排序。

FILE\* fp：关联打开的文件。

int i, j：排序时的循环变量。

stu tem：排序交换时的临时变量。

**3.13密码确定函数模块**

1）函数原形：//密码函数

void key\_show(char\*key)

{

while (1)

{

char key1[20];

system("color 2");

printf("请输入密码:");

scanf("%s", key1);

if (strcmp(key, key1) == 0)

{

printf("密码正确，正在进入......");

Sleep(1000);

system("cls");

break;

}

else

printf("输入错误，请重新输入密码......");

Sleep(1000);

system("cls");

}

}

2）功能：确认使用者输入的密码，正确才能进入程序。

3）变量：

char\*key：传入读取出来的密码字符串。

char key1[20]：存储输入的密码字符串。

**3.14修改密码函数模块**

1）函数原形：//修改密码函数

void change\_key\_func(char\*key)

{

char key1[20];

printf("密码修改为:");

scanf("%s",key1);

strcpy(key, key1);

put\_key\_func(key);

printf("修改成功......");

Sleep(1000);

system("cls");

}

2）功能：修改程序密码

3）变量：

char\*key：传入读取的密码字符串。

char key1[20]：存储输入的修改密码。

**3.15密码写入文件函数模块**

1）函数原形：//密码写入记忆

void put\_key\_func(char\* key)

{

FILE\* fp;

if ((fp = fopen("key.txt", "w")) == NULL)

{

printf("学生信息文件打开失败\n");

exit(0);

}

fputs(key,fp);

fclose(fp);

}

2）功能：将密码写入到文件里储存。

3）变量：

char\* key：传入密码字符串。

FILE\* fp：单开文件，与文件关联。

**3.16备份函数模块**

1）函数原形：//读取函数

int read\_func(list& peo)

{

FILE\* fp;

char str[1000];

int N = 1000;

int i, flog = 0;

if ((fp = fopen("get.txt", "r")) == NULL)

{

printf("学生信息文件打开失败\n");

exit(0);

}

fgets(str,N,fp);

while (fscanf(fp, "%d%d%s%s%s%f%f%f%f", &peo.all\_message[flog].num, &peo.all\_message[flog].cla, peo.all\_message[flog].name, peo.all\_message[flog].sex, peo.all\_message[flog].profession ,& peo.all\_message[flog].chinese\_score, &peo.all\_message[flog].math\_score, &peo.all\_message[flog].english\_score, &peo.all\_message[flog].all\_score) != EOF)

flog++;

peo.length = flog;

fclose(fp);

return 0;

}

2）功能：读取文件内的学生信息。

3）变量：

list& peo：存储读取学生信息。

FILE\* fp：打开文件，关联文件

char str[1000]：存储读取的表头信息。

Int flog ：顺序表下表。

**3.17写入文件函数模块**

1）函数原形：//覆盖写回

int cover\_func(list& peo)

{

FILE\* fp;

int i;

char ko = '\n';

void sort\_func(list&peo,int flog);

if ((fp = fopen("get.txt", "w+")) == NULL)

{

printf("学生信息文件打开失败\n");

exit(0);

}

sort\_func(peo,1);

fprintf(fp, "学号 班级 姓名 性别 专业 语文成绩 数学成绩 英语成绩 总成绩\n");

for (i = 0; i < peo.length; i++)

{

fprintf(fp, "%-11d%-10d%-13s%-11s%-12s%-13.2f%-14.2f%-15.2f%-8.2f%c", peo.all\_message[i].num, peo.all\_message[i].cla, peo.all\_message[i].name, peo.all\_message[i].sex, peo.all\_message[i].profession, peo.all\_message[i].chinese\_score, peo.all\_message[i].math\_score, peo.all\_message[i].english\_score, peo.all\_message[i].all\_score, ko);

}

fclose(fp);

return 0;

}

2）功能：将学生信息写入文件。

3）变量：

list& peo：传入学生信息。

FILE\* fp：打开文件，关联文件。

int i：顺序表下标。

char ko = '\n'：文件内的换行符。

**3.18总成绩计算函数模块**

1）函数原形：//总成绩计算函数

void all\_score\_func(list& peo)

{

int i;

for (i = 0; i < peo.length; i++)

peo.all\_message[i].all\_score = peo.all\_message[i].chinese\_score + peo.all\_message[i].math\_score + peo.all\_message[i].english\_score;

}

2）功能：计算学生总成绩。

3）变量：

list& peo：传入学生信息。

int i：循环下标。

**3.19输出函数模块**

1）函数原形：//输出函数

void a\_output\_func(list& peo, int i)

{

if (peo.all\_message[i].chinese\_score < 60 || peo.all\_message[i].english\_score < 60 || peo.all\_message[i].math\_score < 60)

{

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE); //获取输出设备的句柄

CONSOLE\_SCREEN\_BUFFER\_INFO csbiInfo;

GetConsoleScreenBufferInfo(handle, &csbiInfo); //获取窗口缓冲区信息

WORD wOldColorAttrs = csbiInfo.wAttributes; //获取初始定义的文本属性，即默认属性

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_RED);

printf("学号:%-5d班级:%-5d姓名:%-15s性别:%-8s专业:%-8s语文:%-8.2f数学:%-8.2f英语:%-8.2f总成绩:%-8.2f\n", peo.all\_message[i].num, peo.all\_message[i].cla, peo.all\_message[i].name, peo.all\_message[i].sex, peo.all\_message[i].profession, peo.all\_message[i].chinese\_score, peo.all\_message[i].math\_score, peo.all\_message[i].english\_score, peo.all\_message[i].all\_score);

SetConsoleTextAttribute(handle, wOldColorAttrs);

}

else

printf("学号:%-5d班级:%-5d姓名:%-15s性别:%-8s专业:%-8s语文:%-8.2f数学:%-8.2f英语:%-8.2f总成绩:%-8.2f\n", peo.all\_message[i].num, peo.all\_message[i].cla, peo.all\_message[i].name, peo.all\_message[i].sex, peo.all\_message[i].profession, peo.all\_message[i].chinese\_score, peo.all\_message[i].math\_score, peo.all\_message[i].english\_score, peo.all\_message[i].all\_score);

}

void b\_outpute\_func(list& peo, int i)

{

printf("%-11d%-10d%-13s%-11s%-12s%-13.2f%-14.2f%-15.2f%-8.2f%c", peo.all\_message[i].num, peo.all\_message[i].cla, peo.all\_message[i].name, peo.all\_message[i].sex, peo.all\_message[i].profession, peo.all\_message[i].chinese\_score, peo.all\_message[i].math\_score, peo.all\_message[i].english\_score, peo.all\_message[i].all\_score);

}

2）功能：输出学生信息。

3）变量：

list& peo：传入学生信息。

**3.20分配到班级函数模块**

1）函数原形：//分配到班级

void distribute\_func(list& peo, list& peo1, list& peo2, list& peo3)

{

int i,n1=0,n2=0,n3=0;

peo1.length = 0;

peo2.length = 0;

peo3.length = 0;

for (i = 0; i < peo.length; i++)

{

if (peo.all\_message[i].cla == 1)

{

peo1.all\_message[n1] = peo.all\_message[i];

n1++;

peo1.length++;

}

if (peo.all\_message[i].cla == 2)

{

peo2.all\_message[n2] = peo.all\_message[i];

n2++;

peo2.length++;

}

if (peo.all\_message[i].cla == 3)

{

peo3.all\_message[n3] = peo.all\_message[i];

n3++;

peo3.length++;

}

}

sort\_func(peo1, 5);

sort\_func(peo2, 5);

sort\_func(peo3, 5);

}

2）功能：将从文件读取的全部学生信息分配到各个班级中去

3）变量：

list& peo ：总学生信息结构体数组，将学生信息传输进来。

list& peo1：一班学生信息结构体数组，将一班学生信息传输进来。

list& peo2：二班学生信息结构体数组，将二班学生信息传输进来。

list& peo3：二班学生信息结构体数组，将二班学生信息传输进来。

**3.21各种重复利用的输出函数模块**

1）函数原形：//-------

void a\_func()

{

printf("------------------------------------------------------------------------------------------\n");

}

void b\_func()

{

printf("你的选择是：");

}

//

void c\_func()

{

printf("---------------------------------------------\n");

}

//返回主菜单

void return\_func()

{

printf("按任意键返回主菜单......");

getch();

system("cls");

}

//输入错误

void error\_func()

{

system("color 4");

printf("输入错误，重新输入\n");

Sleep(3000);

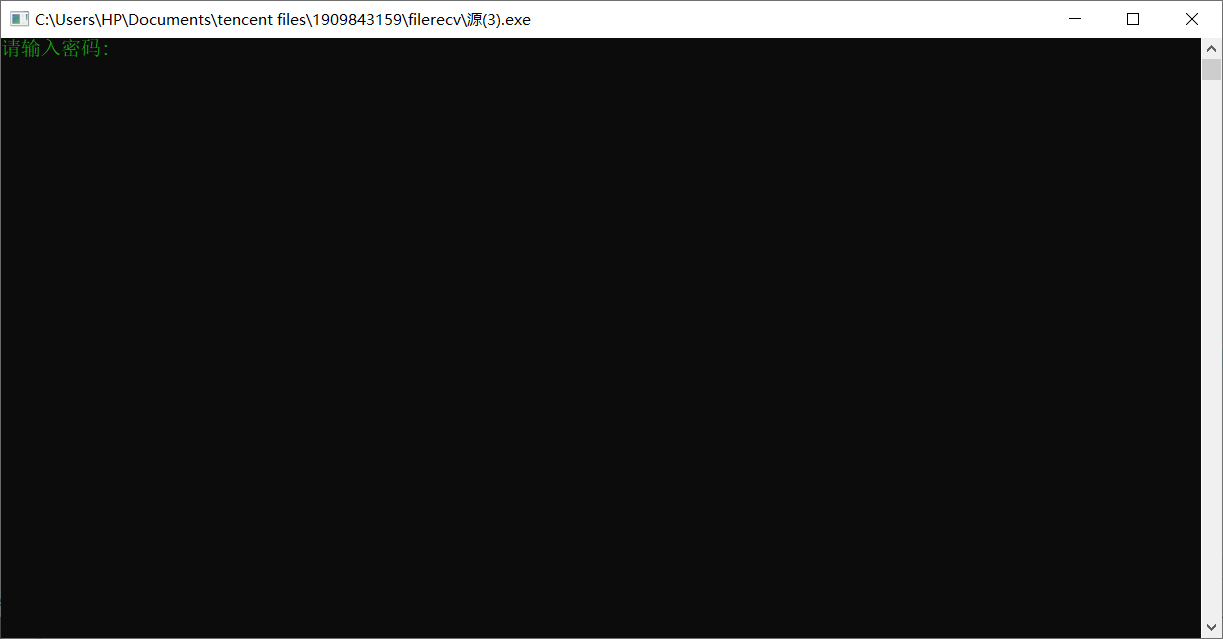
system("color B4");

}

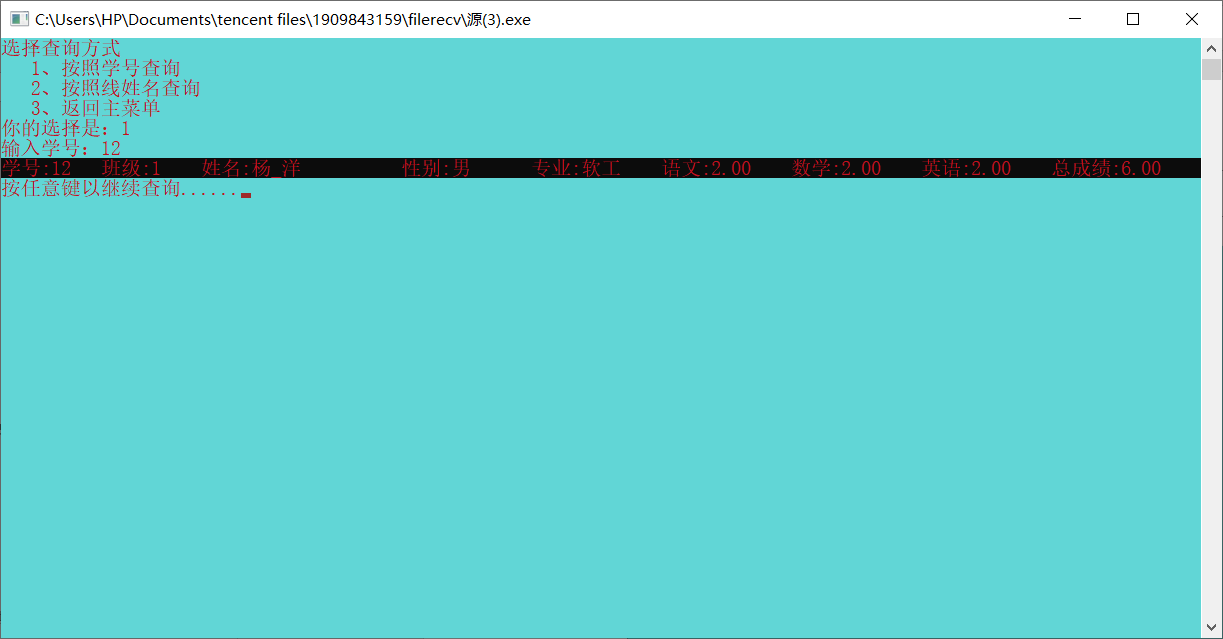
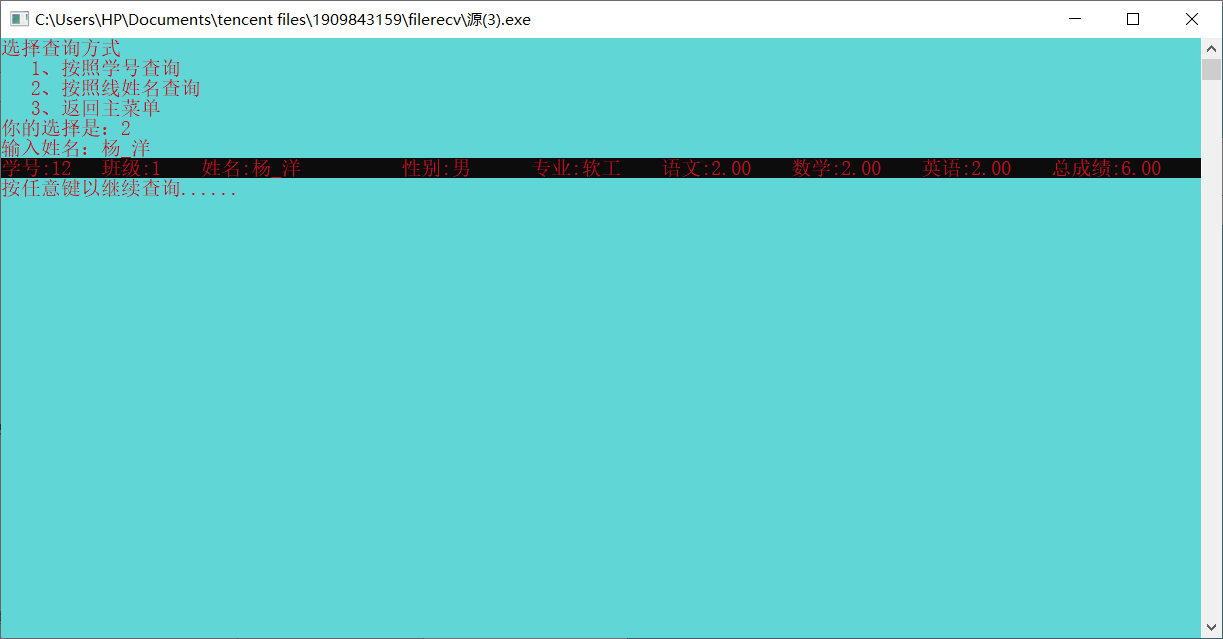
2）功能：封装各种重复利用的输出函数。

**4 总结**

**4.1程序调试情况**

（1）通过正确的密码进入程序（2）密码正确后进入欢迎界面 

（3）进入主菜单（4）查询学生信息分为

可以按照学号查找与姓名查找 1)按学号查询： 2)按照姓名查询： 3)错误输入反馈：（**5）添加学生信息**

可以添加单人信息也可以添加多人信息（6）修改学生信息

（7）删除学生信息

可以按照学号删除与姓名删除

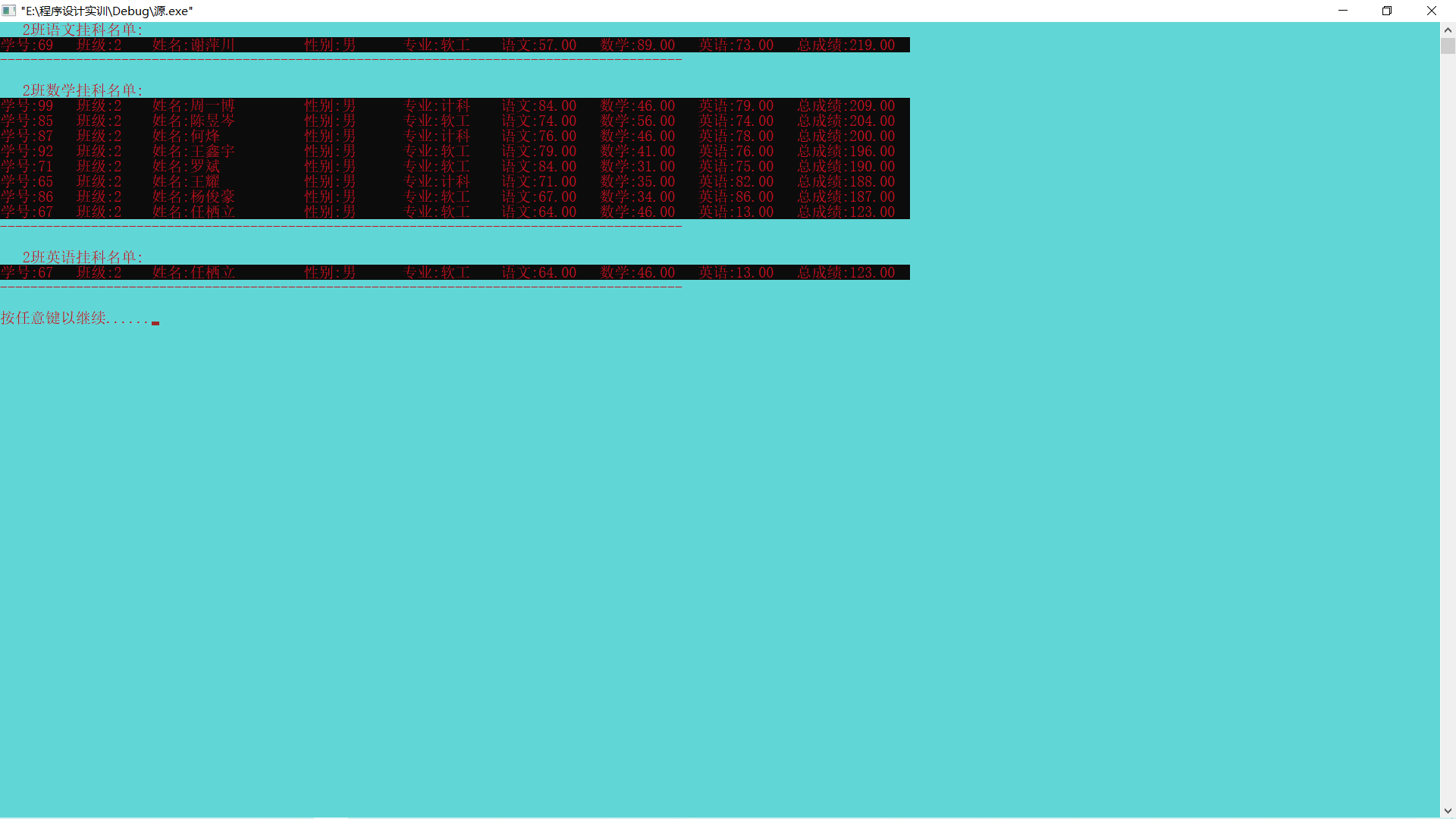
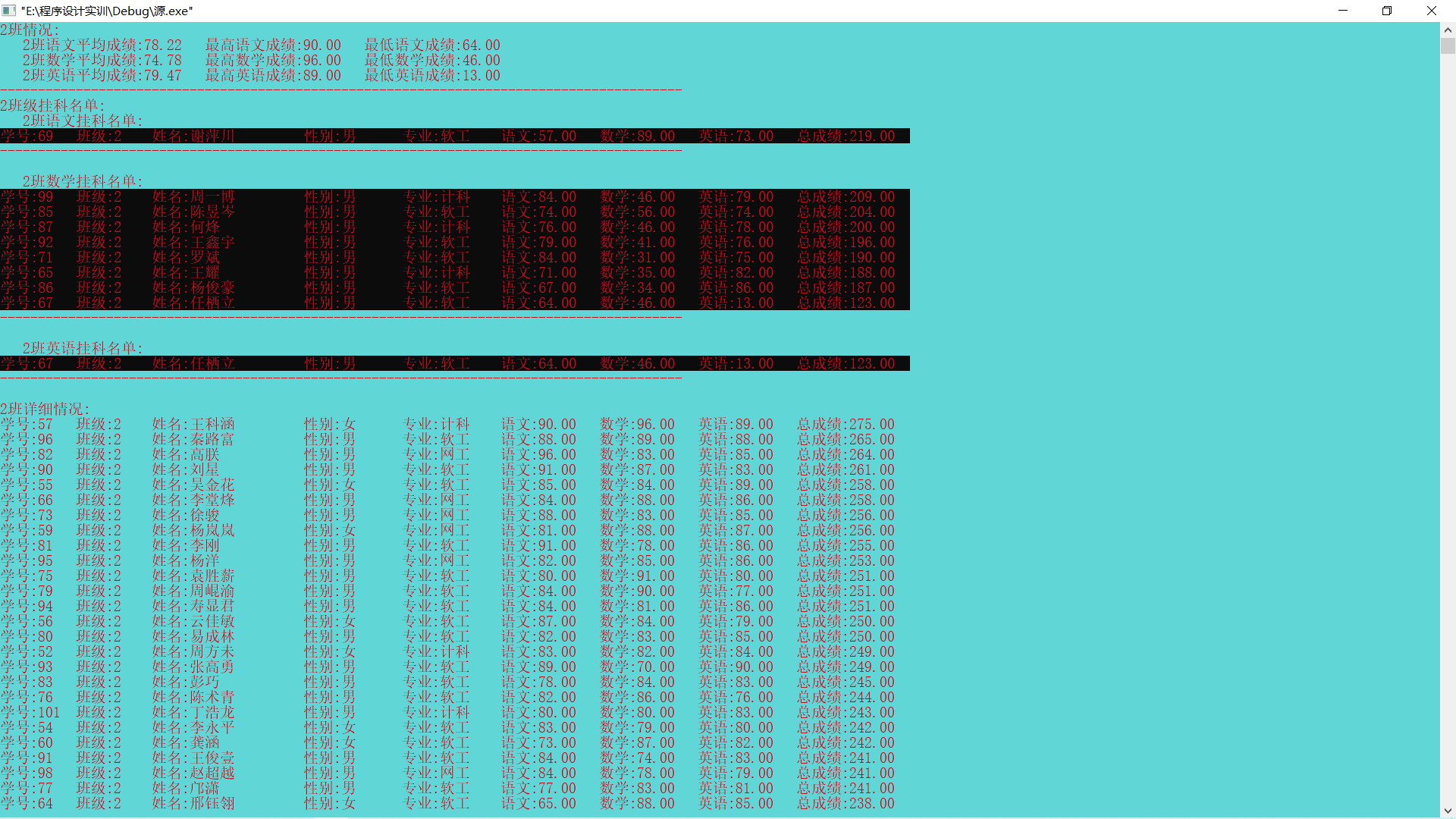


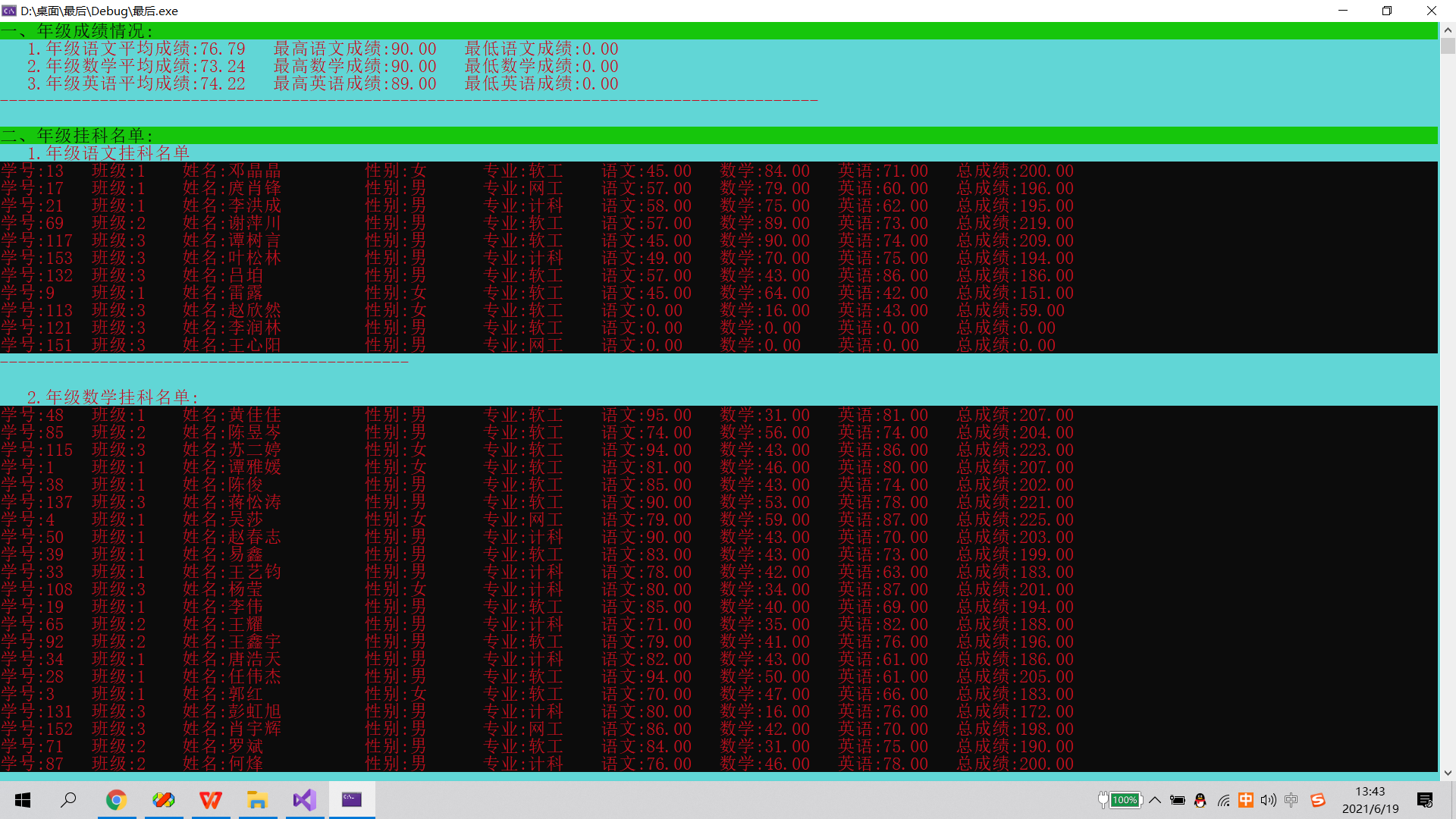
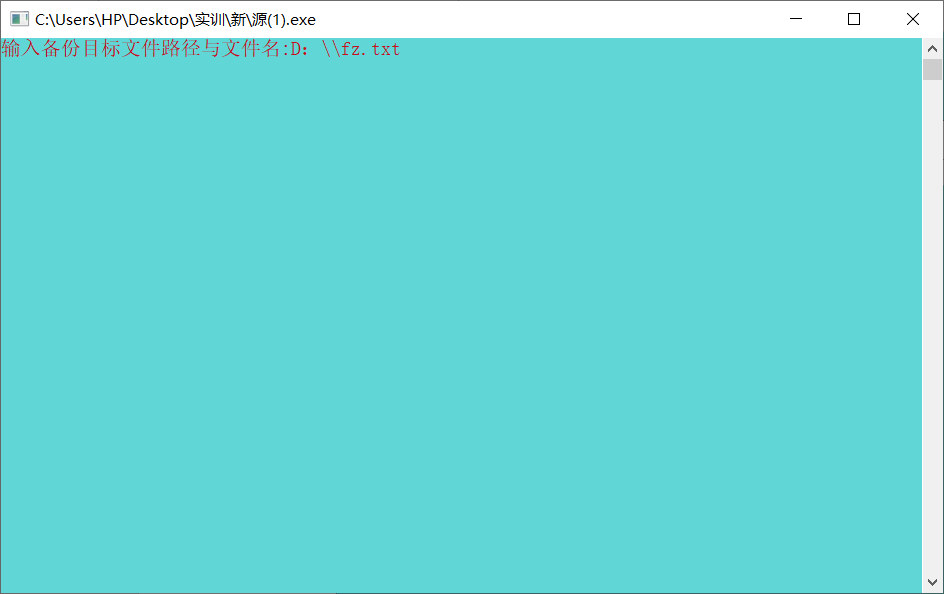
1)按照学号删除：

 2)按照姓名删除：

（8）班级学生情况

可以分别看各个班级的语文成绩，数学成绩，英语成绩挂科名单和挂科情况

1)班级学科平均成绩： 2)班级挂科名单： 3)班级情况总览：（9）年级情况总览

可以看年级的各科目平均，最高，最低和挂科名单，和总成绩的排名表（10）备份文件（11）读取文件（12）修改密码（13）退出程序

**4.2本人在程序设计中感想**

1、困难：

（1）将文件内信息读取到程序顺序表很多错误，例如无法读取，读取错位，读取乱码等，通过不断地尝试，网上查资料，发现问题，解决问题。读取错误一般都是以下问题：1）没有没有注意文件打开方式，例如想读取文件内容，但是却以“w”方式打开文件。2）还有一种经常犯的错误是没有注意fscanf（）函数使用时，没有注意字符串不需要加取地址符，而字符，整形等数据类型需要加取地址符号。

（2）将顺序表内容写入到文件也出了很多错误，例如写入错位啥的，问题和读取时基本差不多，解决方法和注意点也差不多。

（3）在设计菜单的时候，使用一个死循环while（）或者for（；；）加一个出口条件完成，当进入选项时，要注意使用getch（）函数，防止执行结果瞬间结束，就回到了主菜单函数，从而看不不见运行结果，所以使用getch（）函数再加一个提示语句“按任意键以继续”，就可以防止程序运行瞬间闪过结果。

2、收获：

（1）大量的亲自敲代码，熟练的掌握c语言的各种语法细节，提高了熟悉程度，提升了输入速度。

（2）程序中以顺序表存储数据，是程序的数据核心，通过这个程序的大量使用顺序表，熟练的掌握数据结构中顺序表的使用。

（3）掌握c++引用传参，引用传参能比指针语法更简单，更直观。

（4）程序设计文件的写入与读取与备份，是我熟练的掌握了文件操作，只需要读取文件内容时，就一“r”方式打开，需要覆盖写入一“w”方式打开，追加就以“a”方式打开。

（5）文件操作使我能熟练的使用fscanf（）函数，fprintf（）函数，fopen（）函数，rewind（）函数，fgets（）函数，，fputs（）函数等等一些列文件相关的函数。

（6）用system（color xx）函数可以美化控制台窗口，以一个参数控制背景颜色，第二个参数控制字体颜色。

（7）使用颜色控制函数

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE); //获取输出设备的句柄

CONSOLE\_SCREEN\_BUFFER\_INFO csbiInfo;

GetConsoleScreenBufferInfo(handle, &csbiInfo); //获取窗口缓冲区信息

WORD wOldColorAttrs = csbiInfo.wAttributes; //获取初始定义的文本属性，即默认属性

能够自由控制输出前景颜色与背景颜色。

（8）学会重复利用函数，将重复利用的函数封装起来，例如匹配函数，输出函数，避免重复写代码。