序 号 41 学 号 20211120138 姓名 薛凡豪 成绩

试验次数 05 实验名称高级特性实践与综合应用 日期12.25

# 综合应用

## 1.1问题分析

1.由于本次实验是一次综合实验，并且考虑到实验要求，我们需要使用到数组、指针、文件等多种操作应用，并且要采用结构体和非结构体两种实现方式，所以要综合考虑问题。

2.本次实验非结构体方面我决定采用五个学生，三门成绩的方式来实现，并且利用文件的写入“w”和读取“r”来实现信息的输入和输出。并且给出菜单选项的方式进行功能实现。

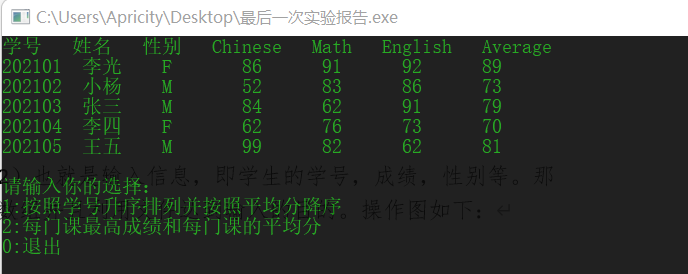
3.本次实验结构体方面我不再采用文件方式写入，而是用户自己输入的方式input数据，因为**本次结构体我想写基本不受限制的用户数目**，可以增加用户数目并且减少限制性，可以有更好的实用性和体验性。

4.由于本次实验较为复杂，所以便不再绘制流程图和伪代码描述，我会在每个功能后进行详细注明该功能的实现操作和思路以便更好地体现我的思路和实现过程，也能弥补流程图和伪代码缺失带来的影响。

## 2.2 功能实现

1.**实验任务（1）（2）的实现操作**

根据实验要求，本次实验的（1）（2）也就是输入信息，即学生的学号，成绩，性别等。那么为了完成此任务，我们可以定义数组并且利用文件达到输入的目的。操作图如下：



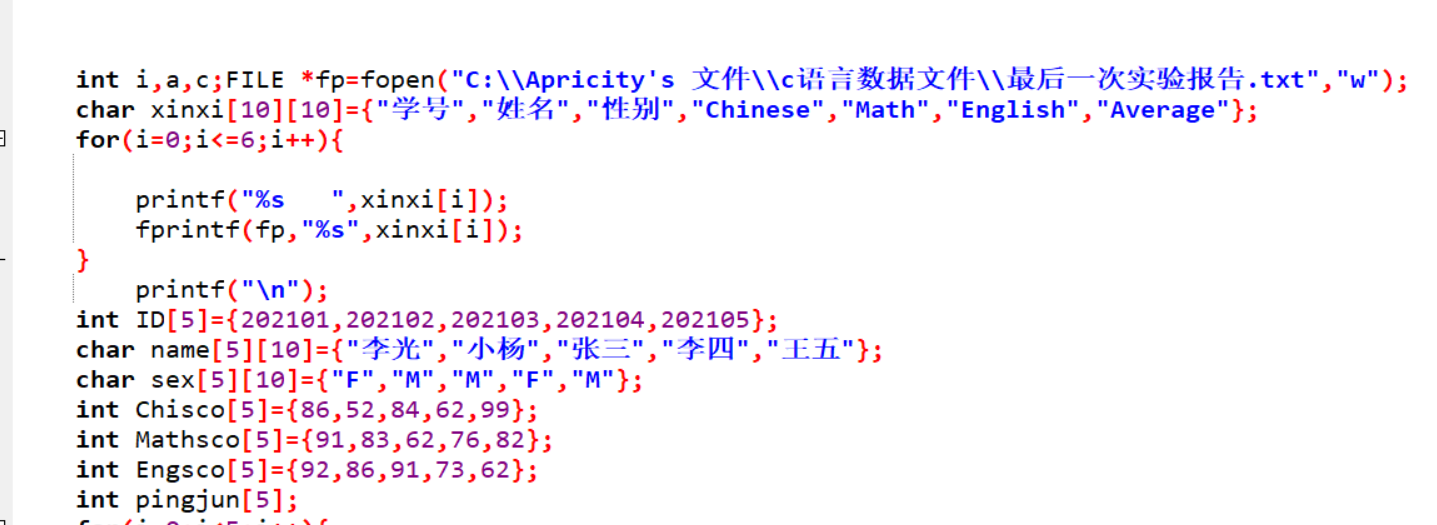
其中学生信息是通过数组打印而出，并且在下方附出菜单选项，也即本次实验任务我给出的功能选项：

1：按照学号升序 2：计算平均分并降序 3：每门课程的最高成绩 4：每门课的平均分

这四个功能将在下方给出解释。

那么为了实现本次的信息输入。我采用了数组和文件的方式进行实现。

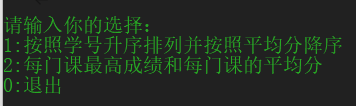
多说无益，上图



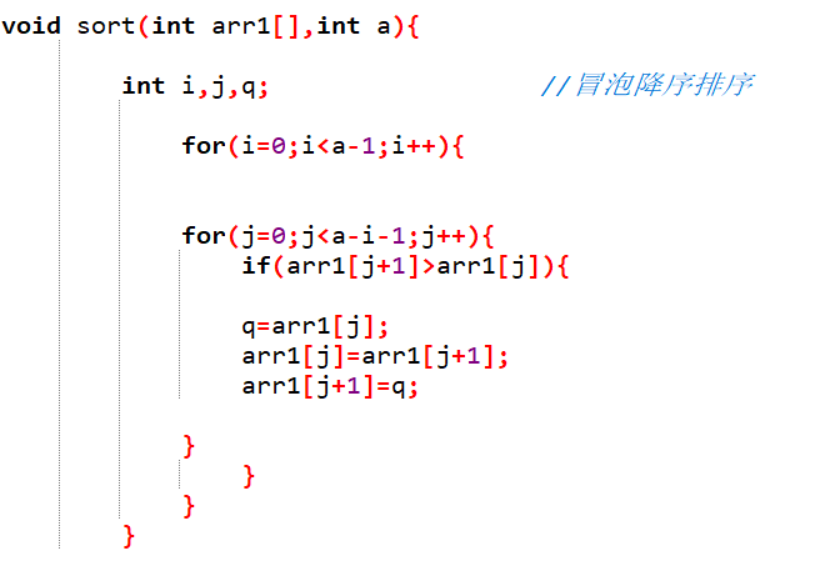
其实思路很简单，也就是利用一维数组进行信息的收纳。并且进行文件的写入。注意数据不要溢出所设定的数组大小即可。

2**.实验任务（3）（4）（5）（6）（8）的实现**

也即我所给出的菜单中的功能实现



这里的排序的原理我用到了冒泡排序，当然还要利用结构体达到整体排序的目的。

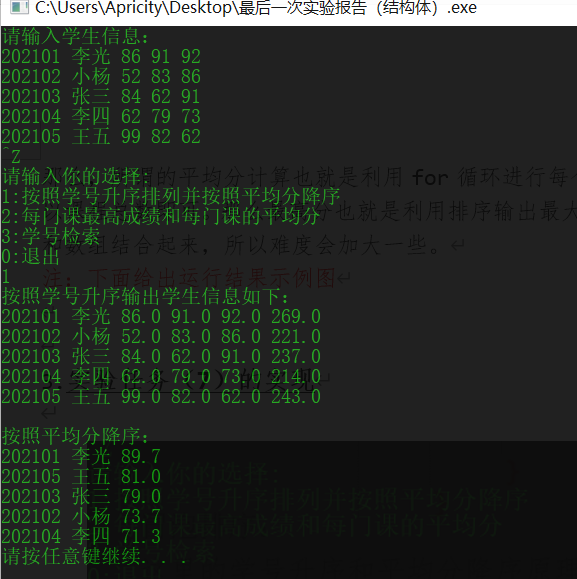
冒泡排序：

这里的学号升序和平均分降序原理相同，便不再赘述了。

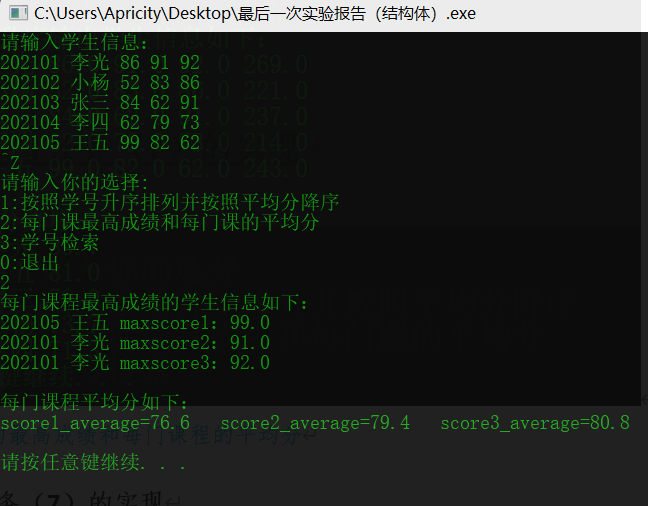
那么，所谓的平均分计算也就是利用for循环进行每个学生每门课程的累加total成绩除以课程总数即可。那么最高分也就是利用排序输出最大值，便能简单的实现了。要注意的是和数组结合起来，所以难度会加大一些。

注：下面给出运行结果示例图

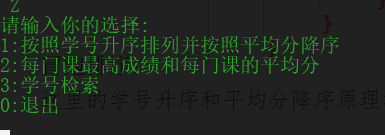
1.按照学号升序和平均分降序排列



2.每门课的最高成绩和每门课程的平均分



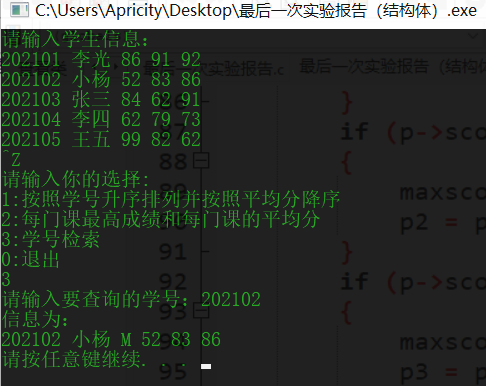
3.**实验任务（7）的实现**

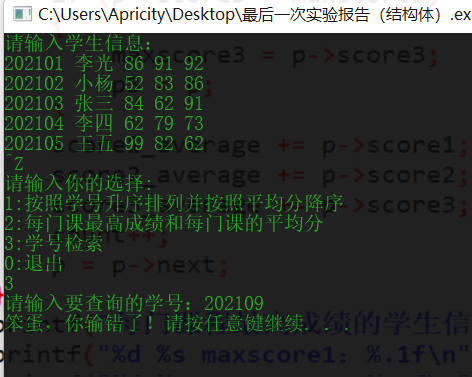
也即

功能3：学号检索的实现

学号检索的功能功能也即进行数字比对，如果相同，即可输出一系列学生信息，此处需要结构体的实现。（名字检索也一样，但是需要利用字符串数组来实现，并且利用strcmp函数实现比较）

下面给出运行实例



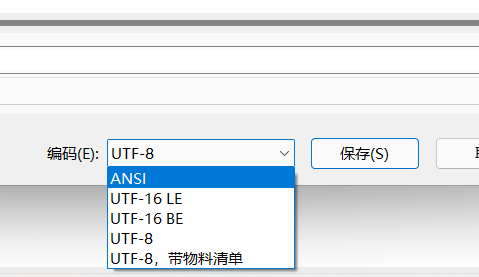


这里提供了错误性检查。其实每一步用户的输入都应有错误检查，这样整个程序才能更健壮，保证良好的实用性和避错性。

## 2.3总结

1.由于本次实验较为复杂和综合，所以思路一定要清晰，应多利用调用函数来解决问题，否则main函数会太冗杂。

2.注意数组的定义长度一定要充足，否则会出现很多问题。

3.由于我用的是devc++，这个版本不支持utf-8，那么我再利用文件的写入时，必须把记事本的编码格式改为ANSI 否则会出现乱码。（血的教训）

4.一定要注意字符串不可以直接比较，需要调用<string.h>文件库里的函数。常用的有strcmp,strlen等

5.一步一步实现！一步一步来！要稳住！不可着急！top-down!

6.源代码放在附录

# 附录

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct student {

int stuid;

char name[20];

float score1;

float score2;

float score3;

struct student \*next;

}stu,\*pstu;

void linklist\_print(pstu phead);

void linklist\_insert(pstu \*pphead, pstu \*pptail, int stuid, char \*name, float score1, float score2, float score3);

void selectinfor(pstu p);

void sort\_totalscore(pstu p);

#define N 5

float sum[5];

void linklist\_print(pstu p)

{

int count = 0;

while (p)

{

sum[count] = p->score1 + p->score2 + p->score3;

printf("%d %s %.1f %.1f %.1f %.1f\n",p->stuid,p->name, p->score1,p->score2,p->score3,sum[count]);

count++;

p = p->next;

}

printf("\n");

}

void linklist\_insert(pstu \*pphead, pstu \*pptail, int stuid, char \*name, float score1,float score2,float score3)

{

pstu pnew = (pstu)calloc(1,sizeof(stu));

pstu ppre, pcur;

pnew->stuid = stuid;

strcpy(pnew->name, name);

pnew->score1 = score1;

pnew->score2 = score2;

pnew->score3 = score3;

if (NULL == (\*pphead))

{

\*pphead = pnew;

\*pptail = pnew;

}

else if (pnew->stuid <= (\*pphead)->stuid)

{

pnew->next = (\*pphead);

(\*pphead) = pnew;

}

else if (pnew->stuid > (\*pphead)->stuid && pnew->stuid <= (\*pptail)->stuid)

{

ppre = \*pphead;

pcur = ppre->next;

while (pcur)

{

if (pnew->stuid > ppre->stuid && pnew->stuid <= pcur->stuid)

{

pnew->next = pcur;

ppre->next = pnew;

break;

}

ppre = pcur;

pcur = pcur->next;

}

}

else {

(\*pptail)->next = pnew;

\*pptail = pnew;

}

}

void selectinfor(pstu p)

{

pstu p1, p2, p3;

float maxscore1=0, maxscore2=0, maxscore3=0;

float score1\_average = 0, score2\_average = 0, score3\_average = 0;

int count = 0;

while (p)

{

if (p->score1 > maxscore1)

{

maxscore1 = p->score1;

p1 = p;

}

if (p->score2 > maxscore2)

{

maxscore2 = p->score2;

p2 = p;

}

if (p->score3 > maxscore3)

{

maxscore3 = p->score3;

p3 = p;

}

score1\_average += p->score1;

score2\_average += p->score2;

score3\_average += p->score3;

count++;

p = p->next;

}

printf("每门课程最高成绩的学生信息如下：\n");

printf("%d %s maxscore1：%.1f\n", p1->stuid, p1->name, p1->score1);

printf("%d %s maxscore2：%.1f\n", p2->stuid, p2->name, p2->score2);

printf("%d %s maxscore3：%.1f\n", p3->stuid, p3->name, p3->score3);

printf("\n");

printf("每门课程平均分如下：\n");

printf("score1\_average=%.1f score2\_average=%.1f score3\_average=%.1f\n", score1\_average / count, score2\_average / count, score3\_average / count);

printf("\n");

}

void sort\_totalscore(pstu p)

{ int i,j;

pstu pnode[N];

float temp;

pstu q;

for (i = 0; i < N; i++,p=p->next)

pnode[i] = p;

for (i = N; i > 1; i--)

{

for (j = 0; j < i - 1; j++)

{

if (sum[j] < sum[j + 1])

{

temp = sum[j + 1];

sum[j + 1] = sum[j];

sum[j] = temp;

q = pnode[j + 1];

pnode[j + 1] = pnode[j];

pnode[j] = q;

}

}

}

printf("按照平均分降序：\n");

for (i = 0; i < N; i++)

printf("%d %s %.1f\n", pnode[i]->stuid, pnode[i]->name, sum[i]/3);

}

int main()

{int a=1,b;

printf("请输入学生信息：\n");

pstu phead = NULL, ptail = NULL;

int stuid;

char name[20];

float score1, score2, score3;

while(a!=0){

while (scanf("%d%s %f%f%f", &stuid, name, &score1, &score2, &score3)!=EOF)

{

linklist\_insert(&phead,&ptail,stuid, name, score1, score2, score3);

}

printf("请输入你的选择:\n1:按照学号升序排列并按照平均分降序\n2:每门课最高成绩和每门课的平均分\n3:学号检索\n0:退出\n");

scanf("%d",&a);

if(a==1){

printf("按照学号升序输出学生信息如下：\n");

linklist\_print(phead);

sort\_totalscore(phead);

} //打印第1问的结果

if(a==2){

selectinfor(phead);

} //处理并打印第2、3问的结果

}

//处理打印第4问的结果

system("pause");

}

}