桂林电子科技大学2018-2019学年 第2学期

**程序设计与问题求解 实验报告**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | **实验十 综合程序设计** | | | | | | | |  | 辅导教师意见：  成绩 教师签名： |
| 院 系 | **计算机与信息安全学院** | | | 专业 | | **计算机** | | |
| 学 号 | **1800300636** | | | 姓名 | | **奚锐** | | |
| 同 作 者 |  | | | | | | | |
| 实验日期 | **2019** | 年 | **6** | | 月 | | **13** | 日 |
|  |  | | | | | | | |

### 一、实验目的

1．熟悉和回顾编程的一般方法

2．循序渐进编写实际编程问题

3．进一步学习和提高综合编程能力和解决编程中出现的问题的能力

### 二、实验内容

**１.　设计和完成：简单学生成绩管理信息系统**

学生成绩信息包括：学号，姓名，课程名，平时成绩，实验成绩，考试成绩，总评成绩。

**２.　要求实现如下功能：**

1. 能够实现学生成绩信息的插入、删除和修改；
2. 能够实现各种查询（分别根据学生学号、姓名、课程名称等）；
3. 能够实现按照考试成绩、总评成绩进行排序；
4. 能够查询某门课程的最高分、最低分并输出相应学生信息；
5. 能够查询某门课程的优秀率（90 分及以上）、不及格率；

**源代码：**源代码：

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<windows.h>

#include<math.h>

FILE\* fpin, \* fpout;

bool isedit=false;

char filename[] = "std.txt";

typedef struct classmem{

long number;

char name[20];

int classcount=0;

int total=0;

float summery=0;

};

typedef struct stu\_record {

long number;

char name[20];

char CourseName[20];

long classnum;

long CourseNum;

int CourseScore;

int TestScore;

int FinalScore;

};

typedef struct stud\_node

{

int id;

stu\_record record;

stud\_node\* next;

};

stud\_node\* newstu;

int size = sizeof(struct stud\_node);

stud\_node\* create(FILE\* fp) {//从文件中读取数据，存在一个新的链表中

stud\_node\* head, \* tail, \* p;

int num;

num = 1;

p = (struct stud\_node\*)malloc(size);

tail = p;

while (!feof(fp))

{

if (num == 1) {

head = p;

}

//fscanf(fp, "%s", &p->record.name);

//fscanf(fp, "%s", &p->record.CourseName);

fscanf(fp, "%s %s %ld %ld %ld %d %d", &p->record.name, &p->record.CourseName, &p->record.number, &p->record.CourseNum, &p->record.classnum, &p->record.CourseScore, &p->record.TestScore);

p->record.FinalScore=(p->record.TestScore+p->record.CourseScore)\*0.5;

p->id=num;

p->next = NULL;

tail->next = p;

tail = p;

p = (struct stud\_node\*)malloc(size);

num++;

}

// p->next = NULL;

tail->next = NULL;

//tail = p;

return head;

printf("successful");

}

stud\_node\* read(char fname[20],char mode[10])//读取文件，获得新链

{

if ((fpin=fopen(fname, mode))==NULL) {

printf("open file failed\n");

return NULL;

}

else {

return create(fpin);

}

}

stud\_node\* stu\_search(stud\_node\* originchain , int op)//提供各类搜索方法，反回查找所得的新链的首地址

{

long number;

char name[20];

long classnum,coursenum;

switch(op)

{

case 0:

printf("输入想要查找的名字");

gets(name) ;

puts(name);

break;

case 1:

printf("输入想要查找的学号");

scanf("%ld",&number);

getchar();

break;

case 2:

//getchar();

printf("输入课号");

scanf("%ld",&coursenum);

break;

case 3:

printf("输入想要查找的班级");

scanf("%ld",&classnum);

break;

default :

return NULL;

}

stud\_node\* head=NULL;

stud\_node\* p;

int count=0;

p=(stud\_node\*)malloc(size);

while(originchain!=NULL)

{

//printf("%d",strcmp(originchain->record.name,name));

if(strcmp(originchain->record.name,name)==0||originchain->record.CourseNum==coursenum||classnum==originchain->record.classnum||originchain->record.number==number)

{

printf("matched\n");

if(count==0)

{

p=originchain;

head=p;

}

else

{

p->next=originchain;

p=(stud\_node\*)malloc(size);

p=originchain;

}

count++;

}

originchain=originchain->next;

}

p->next=NULL;

if(count==0)

{

printf("无法查询到任何结果");

}

return head;

}

stu\_record AddNew(int mode) {//可以直接写新记录到文件中，也可以返回一个结构体

system("cls");

printf("[编辑视图->添加新记录]");

long number;

char name[20];

char CourseName[20];

long classnum;

long CourseNum;

int CourseScore;

int TestScore;

stu\_record newstu;

getchar();

printf("输入学生姓名\n");

gets(name);

printf("输入课程名称\n");

scanf("%s",&CourseName);

printf("输入学号\n");

scanf("%ld",&number);

printf("课程序号\n");

scanf("%ld",&CourseNum);

printf("班级\n");

scanf("%ld",&classnum);

printf("现在请输入学生平时成绩\n");

scanf("%d",&CourseScore);

printf("现在请输入学生期末成绩\n");

scanf("%d",&TestScore);

printf("%s %s %ld %ld %ld %d %d %d\n",name,CourseName,number,CourseNum,classnum,CourseScore,TestScore,(CourseScore+TestScore)\*0.5);

if(mode==1)

{

fprintf(fpin,"%s %s %ld %ld %ld %d %d\n",name,CourseName,number,CourseNum,classnum,CourseScore,TestScore);

}

else if(mode==2)

{

strcpy(newstu.name,name);

newstu.number=number;

newstu.classnum=classnum;

strcpy(newstu.CourseName,CourseName);

newstu.CourseNum=CourseNum;

newstu.CourseScore=CourseScore;

newstu.TestScore=TestScore;

newstu.FinalScore =(CourseScore+TestScore)\*0.5;

}

return newstu;

}

int edit(stud\_node\* a,int id)//修改某条记录

{

system("cls");

printf("[编辑视图->编辑单条记录]\n");

int op,temp;

long templd;

char tempstr[20];

stud\_node\* b;

while(a->id!=id&&a->next!=NULL)

a=a->next;

b=(stud\_node\*)malloc(sizeof(stud\_node));

b->record=a->record;

printf( "[1.姓名]%s [2.课程名]%s [3.学号]%ld [4.课号]%ld [5.班级]%ld [6.平时分]%d [7.考试分]%d\n", a->record.name, a->record.CourseName, a->record.number, a->record.CourseNum, a->record.classnum, a->record.CourseScore, a->record.TestScore);

scanf("%d",&op);

getchar();

if(op==1)

{

gets(tempstr);

strcpy(b->record.name,tempstr);

}

else if(op==2)

{

gets(tempstr);

strcpy(b->record.CourseName,tempstr);

}

else if(op==3){

scanf("%ld",&templd);

b->record.number=templd;

}

else if(op==4){

scanf("%ld",&templd);

b->record.CourseNum=templd;

}

else if(op==5){

scanf("%ld",&templd);

b->record.classnum=templd;

}

else if(op==6){

scanf("%d",&temp);

b->record.CourseScore=temp;

}

else if(op==7){

scanf("%d",&temp);

b->record.TestScore=temp;

}

else {

//nothing changed

printf("edit failed\n");

}

b->record.FinalScore=(b->record.CourseScore+b->record.TestScore)\*0.5;

//printf( "[1.姓名]%s [2.课程名]%s [3.学号]%ld [4.课号]%ld [5.班级]%ld [6.平时分]%d [7.考试分]%d\n", a->record.name, a->record.CourseName, a->record.number, a->record.CourseNum, a->record.classnum, a->record.CourseScore, a->record.TestScore);

printf("[1]保存 [2]取消更改\n");

scanf("%d",&op);

if(op==1)

{

a->record=b->record;

printf( "[1.姓名]%s [2.课程名]%s [3.学号]%ld [4.课号]%ld [5.班级]%ld [6.平时分]%d [7.考试分]%d\n", a->record.name, a->record.CourseName, a->record.number, a->record.CourseNum, a->record.classnum, a->record.CourseScore, a->record.TestScore);

}

printf("修改成功\n");

return 0;

}

stud\_node\* upadatechain(stud\_node\* a,stud\_node\*b)//将a链内容合并进b中

{

stud\_node\*p,\*last,\*head,\*next,\*bhead,\*ahead;

ahead=a;

while(a->id==0)

{

p=(stud\_node\*)malloc(sizeof(stud\_node));

p=a;

p->next=b;

b=p;

a=a->next;

printf("a record added\n");

}

bhead=b;

for(b;b->next!=NULL;b=b->next)

{

stud\_node \*n1,\*n2;

n1=ahead;

for(n2=n1;n2->next!=NULL;n2=n2->next)

{

if(abs(n2->id)==abs(b->id))

{

b->record=n2->record;

b->id=n2->id;

}

}

}

return bhead;

}

//打印表格

int recordsword(stud\_node\* origin,int total)//排序

{

stu\_record mylist[total-1],temp;

int listcount=0;

int excellent=0,failed=0;

for(origin,listcount;origin->next!=NULL;origin=origin->next,listcount++){

mylist[listcount]=origin->record;

if(mylist[listcount].FinalScore>=90)

excellent++;

if(mylist[listcount].FinalScore<60)

failed++;

}

for(int i=0;i<listcount;i++)

{

int max=i;

for(int j=i+1;j<listcount;j++)

{

if(mylist[j].FinalScore>mylist[max].FinalScore)

max=j;

}

if(max!=i)

{

temp=mylist[i];

mylist[i]=mylist[max];

mylist[max]=temp;

}

}

printf("课程名：%s\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n优秀率：%.2f不及格率：%.2f\n",mylist[0].CourseName,1.0\*excellent/total,1.0\*failed/total);

for(int i=0;i<listcount;i++)

{

printf("%s %d\n",mylist[i].name,mylist[i].FinalScore);

}

getchar();

}

int classsum(stud\_node\* origin)

{

classmem temp;

int max=0,mcount;

stud\_node\* p=origin;

for(p;p->next!=NULL;p=p->next)

{

mcount=(p->record.number%100)-1;

if(mcount>max)

max=mcount;

}

classmem newclass[max];

system("cls");

if(max==0){

printf("无法查询到该班级的信息");

return 0;

}

printf("班级名称：%d\n",origin->record.number) ;

for(origin;origin->next!=NULL;origin=origin->next)

{

mcount=(origin->record.number%100)-1;

//printf("%d\n",mcount);

newclass[mcount].classcount++;

//printf("%d\n",newclass[mcount].classcount++);

newclass[mcount].number=origin->record.number;

newclass[mcount].total+=origin->record.FinalScore;

strcpy(newclass[mcount].name,origin->record.name);

}

for(int i=0;i<=max-1;i++)

{

newclass[i].summery=1.0\*newclass[i].total/newclass[i].classcount;

}

int a;

for(int i=0;i<max-1;i++){

a=i;

for(int j=i;j<max;j++)

{

if(newclass[j].summery>newclass[a].summery)

a=j;

}

if(a!=i){

temp=newclass[i];

newclass[i]=newclass[a];

newclass[a]=temp;

}

}

printf("姓名 总评 学号\n");

for(int i=0;i<=max-1;i++){

printf("%s %.2f %ld \n",newclass[i].name,newclass[i].summery,newclass[i].number);

}

return 0;

}

stud\_node\* summeryview(stud\_node\* resultchain ,int op)//为查找得到的结果进行显示

{

stud\_node\* resultchainhead=resultchain;

int sec;

stud\_node\* p;

int id;

int total=1;

system("cls");

switch(op){

case 1://学生概要

printf("[主页->摘要]\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("姓名：%s\n学号：%ld\n所在班级：%ld\n",resultchain->record.name,resultchain->record.number,resultchain->record.classnum);

printf("课程 课号 平时成绩 期末成绩 总分\n");

for(resultchain;resultchain->next!=NULL;resultchain=resultchain->next)

printf("%s %d %d %d %d\n", resultchain->record.CourseName,resultchain->record.CourseNum, resultchain->record.CourseScore,resultchain->record.TestScore, resultchain->record.FinalScore);

printf("[1]返回 [2]编辑该摘要");

scanf("%d",&sec);

break;

case 2://某个课程班级概要

printf("[主页->列表]\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("课程名:%s\n课号:%ld\n姓名 总分 考试分 平时分 学号 班级\n",resultchain->record.CourseName,resultchain->record.CourseName);

for(resultchain;resultchain->next!=NULL;resultchain=resultchain->next)

{

printf("%s %d %d %d %d %d\n",resultchain->record.name,resultchain->record.FinalScore, resultchain->record.CourseScore,resultchain->record.TestScore, resultchain->record.number,resultchain->record.classnum);

total++;

}

printf("[1]返回 [3]编辑该列表 [4]统计");

scanf("%d",&sec);

break;

defalult:

return 0;

}

resultchain=resultchainhead;

system("cls");

while(sec!=1)

{

isedit=true;

switch(sec)

{

case 2:

printf("[主页->摘要(编辑视图)]\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("姓名：%s\n学号：%ld\n所在班级：%ld\n",resultchain->record.name,resultchain->record.number,resultchain->record.classnum);

printf("记录ID　　课程 课号 平时成绩 期末成绩 总分\n");

for(resultchain;resultchain->next!=NULL;resultchain=resultchain->next)

printf("%d %s %d %d %d %d\n",resultchain->id, resultchain->record.CourseName,resultchain->record.CourseNum, resultchain->record.CourseScore,resultchain->record.TestScore, resultchain->record.FinalScore);

printf("键入记录ID");

scanf("%d",&id);

resultchain=resultchainhead;

edit(resultchain,id);

printf("[1]结束 [2]继续");

scanf("%d",&sec);

break;

case 3:

printf("[主页->列表(编辑视图)\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("课程名:%s\n课号:%ld\n姓名 总分 考试分 平时分 学号 班级\n",resultchain->record.CourseName,resultchain->record.CourseName);

for(resultchain;resultchain->next!=NULL;resultchain=resultchain->next)

printf("%d %s %d %d %d %d %d\n",resultchain->id, resultchain->record.name,resultchain->record.FinalScore, resultchain->record.CourseScore,resultchain->record.TestScore, resultchain->record.number,resultchain->record.classnum);

resultchain=resultchainhead;

printf("[1]结束 [4]添加 [5]删除");

scanf("%d",&sec);

if(sec==4)

{

p=(stud\_node\*)malloc(sizeof(stud\_node));

p->id=0;

p->record=AddNew(2);

p->next=resultchain;

resultchain=p;

resultchainhead=resultchain;

printf("%d %s %d %d %d %d %d\n",resultchain->id, resultchain->record.name,resultchain->record.FinalScore, resultchain->record.CourseScore,resultchain->record.TestScore, resultchain->record.number,resultchain->record.classnum);

total++;

}

if(sec==5)

{

scanf("%d",&id);

for(resultchain;resultchain->next!=NULL;resultchain=resultchain->next)

if(resultchain->id==id)

resultchain->id=-(resultchainhead->id);

resultchain=resultchainhead;

printf("修改成功\n");

}

printf("摁下1退出");

scanf("%d",&sec);

break;

case 4:

recordsword(resultchain,total);

printf("摁下1退出");

scanf("%d",&sec);

break;

default:

return resultchainhead;

}

}

return resultchainhead;

}

int main() {

int op=1;

while(op!=0){

isedit=false;

printf("学生信息管理系统\n[home]\n");

printf("1,输入学生信息\n");

printf("2，按姓名查询\n");

printf("3,按学号查询\n");

printf("4,按班级查询\n");

printf("5，按课号查询\n");

printf("9,退出\n");

scanf("%d", &op);

//stud\_node\* modifychain=create(fpin);

stud\_node\* searchchain=NULL;

stud\_node \* resultchain=NULL;

switch(op)

{

case 1:

fpin=fopen(filename, "a");

AddNew(1);

break;

case 2:

getchar();

if((resultchain=stu\_search(read(filename ,"r"),0))!=NULL)

resultchain=summeryview(resultchain,1);

fclose(fpin);

break;

case 3:

if((resultchain=stu\_search(read(filename ,"r"),1))!=NULL)

resultchain=summeryview(resultchain,1);

break;

case 4:

if((resultchain=stu\_search(read(filename ,"r"),3))!=NULL)

classsum(resultchain);

break;

case 5:

if((resultchain=stu\_search(read(filename ,"r"),2))!=NULL)

resultchain=summeryview(resultchain,2);

break;

defalult:

return 0;

}

fclose(fpin);

fpin=fopen(filename, "r");

searchchain=read(filename ,"r");

fclose(fpin);

fopen(filename, "w");

if(isedit){

printf("保存所做修改吗？[1,保存/2,退出]");

scanf("%d",&op);

if(op==1)

{

searchchain=upadatechain(resultchain,searchchain);

printf("执行修改........");

}

}

while(searchchain->next!=NULL)

{

if(searchchain->id>=0)

fprintf(fpin, "%s %s %ld %ld %ld %d %d\n", searchchain->record.name, searchchain->record.CourseName, searchchain->record.number, searchchain->record.CourseNum, searchchain->record.classnum, searchchain->record.CourseScore, searchchain->record.TestScore);

searchchain=searchchain->next;

}

fclose(fpin);

printf("继续操作吗？[1,继续] [0.离开]");

scanf("%d",&op);

system("cls");

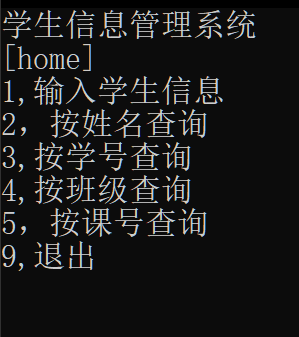
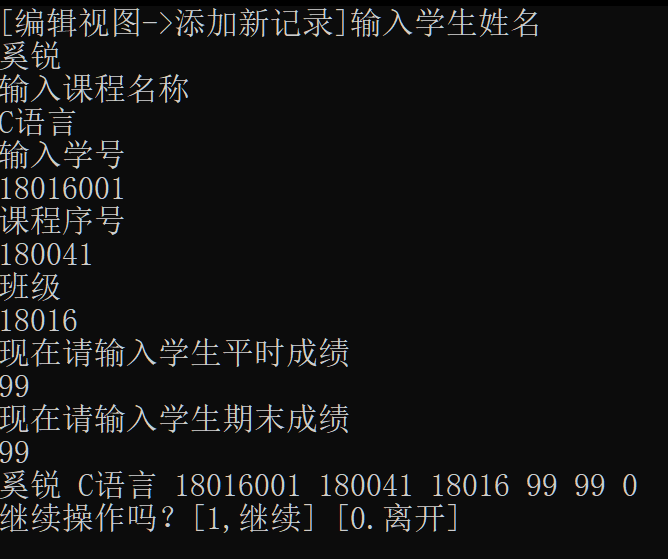
}

return 0;

}

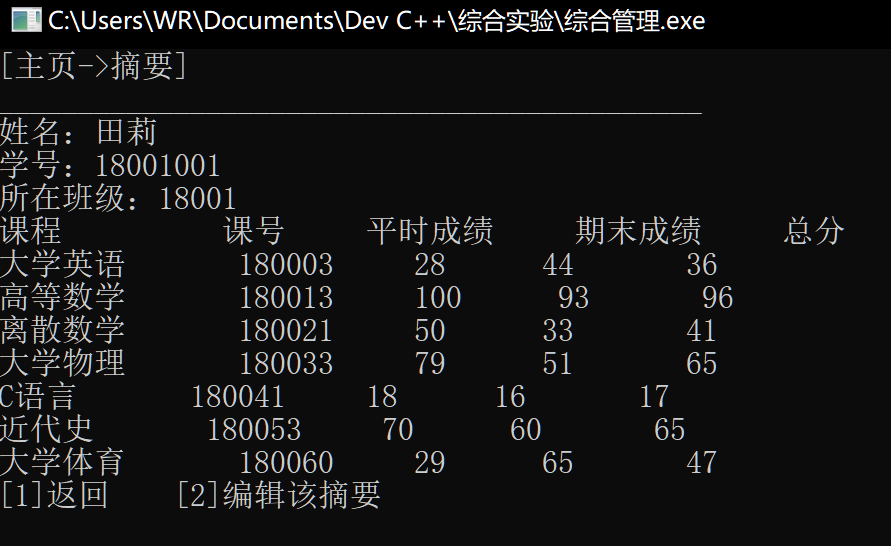
运行截图：

存储数据的文件名：std.txt

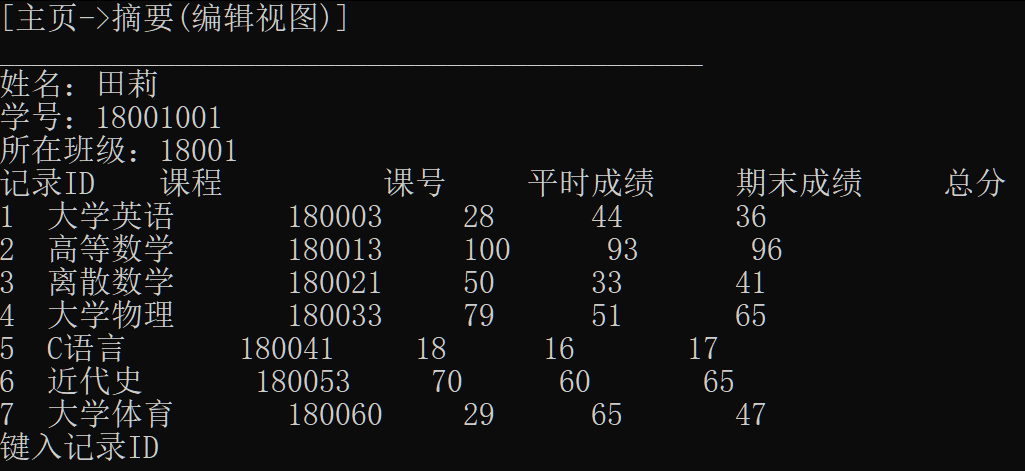
存储数据的文件内容：

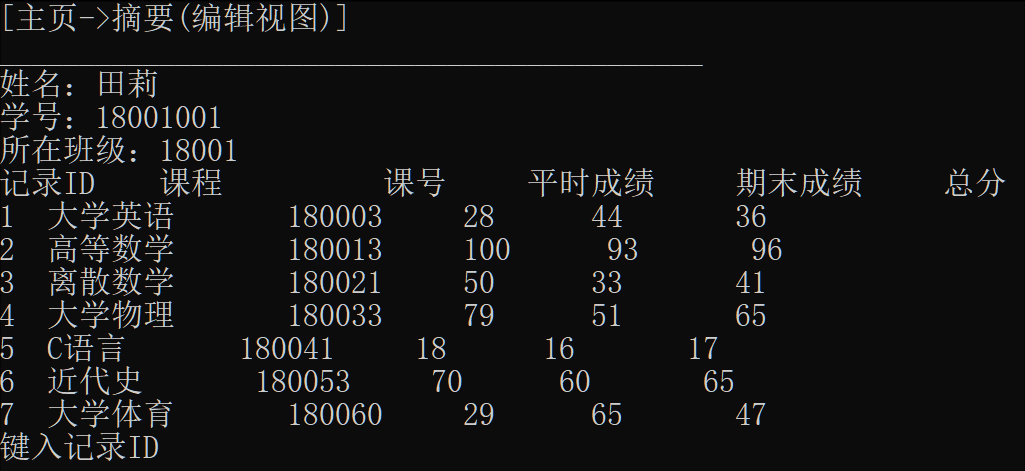
添加记录

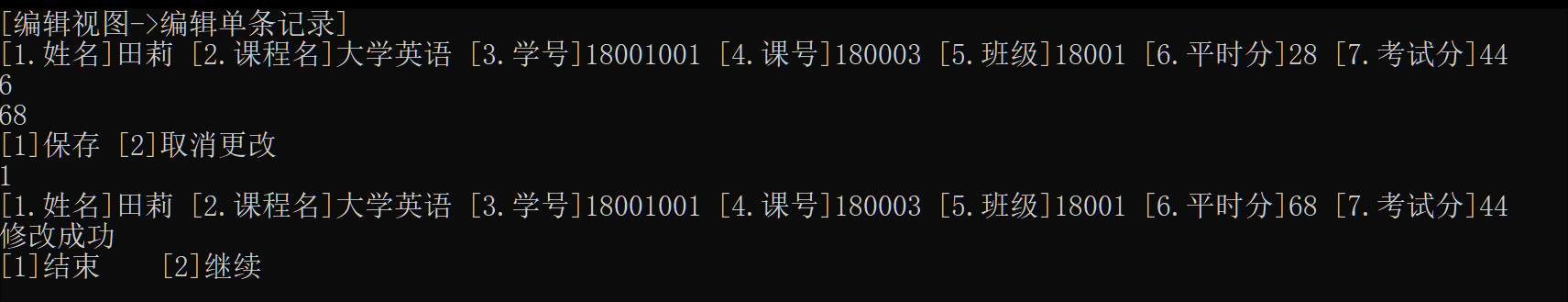
主页



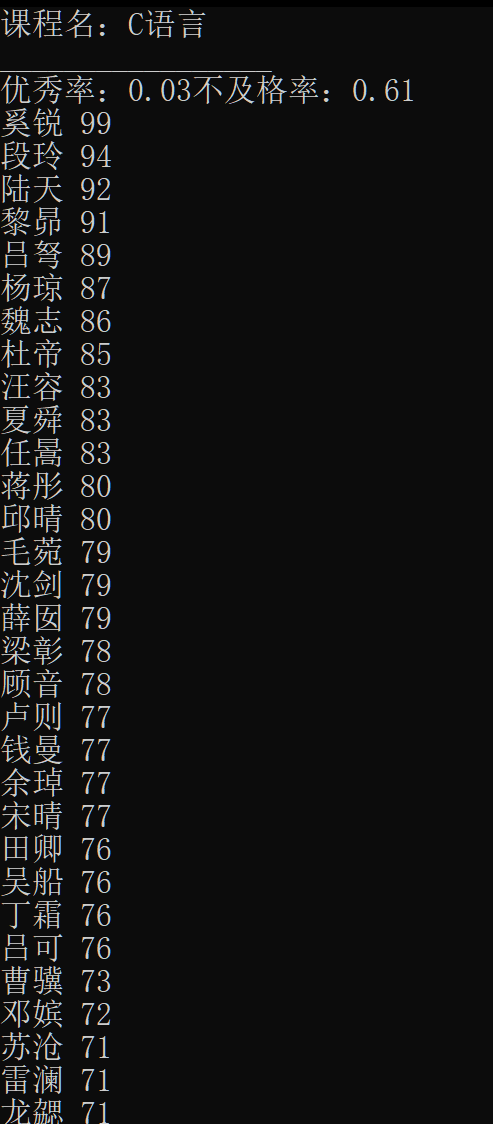
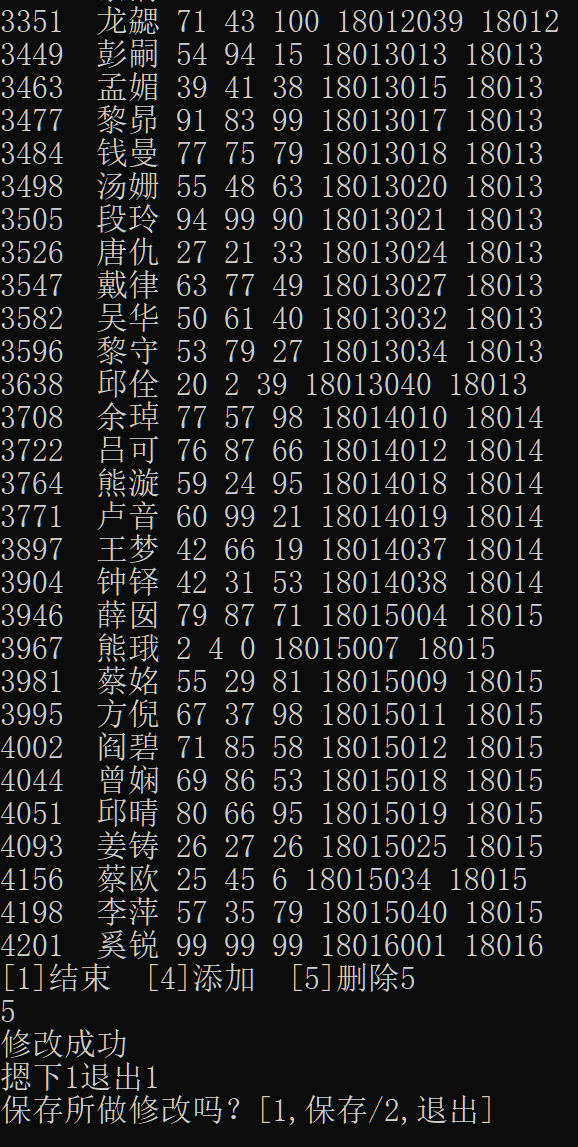
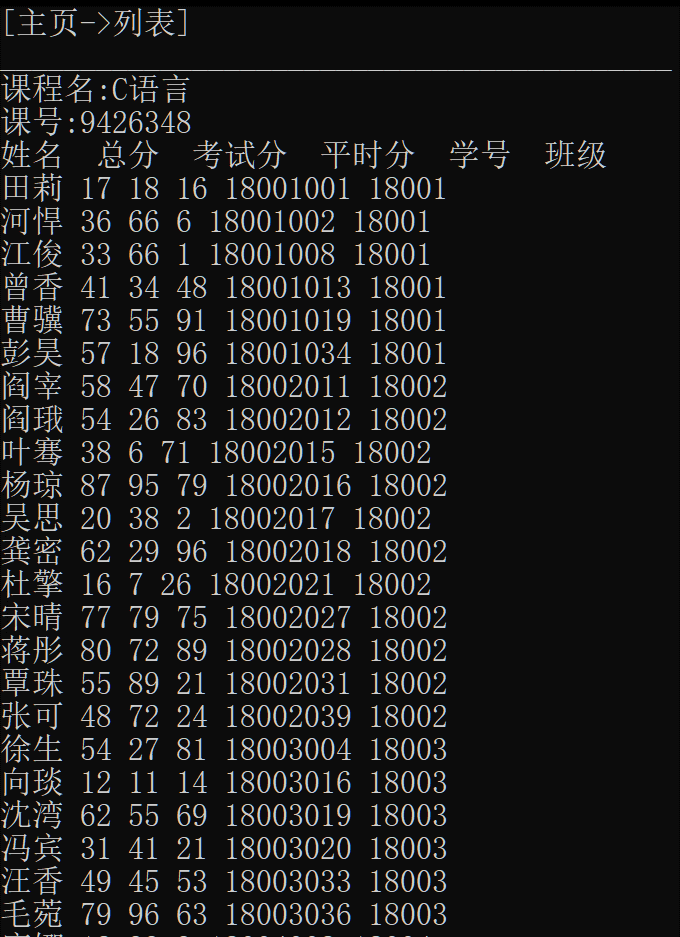
查询学生信息并修改



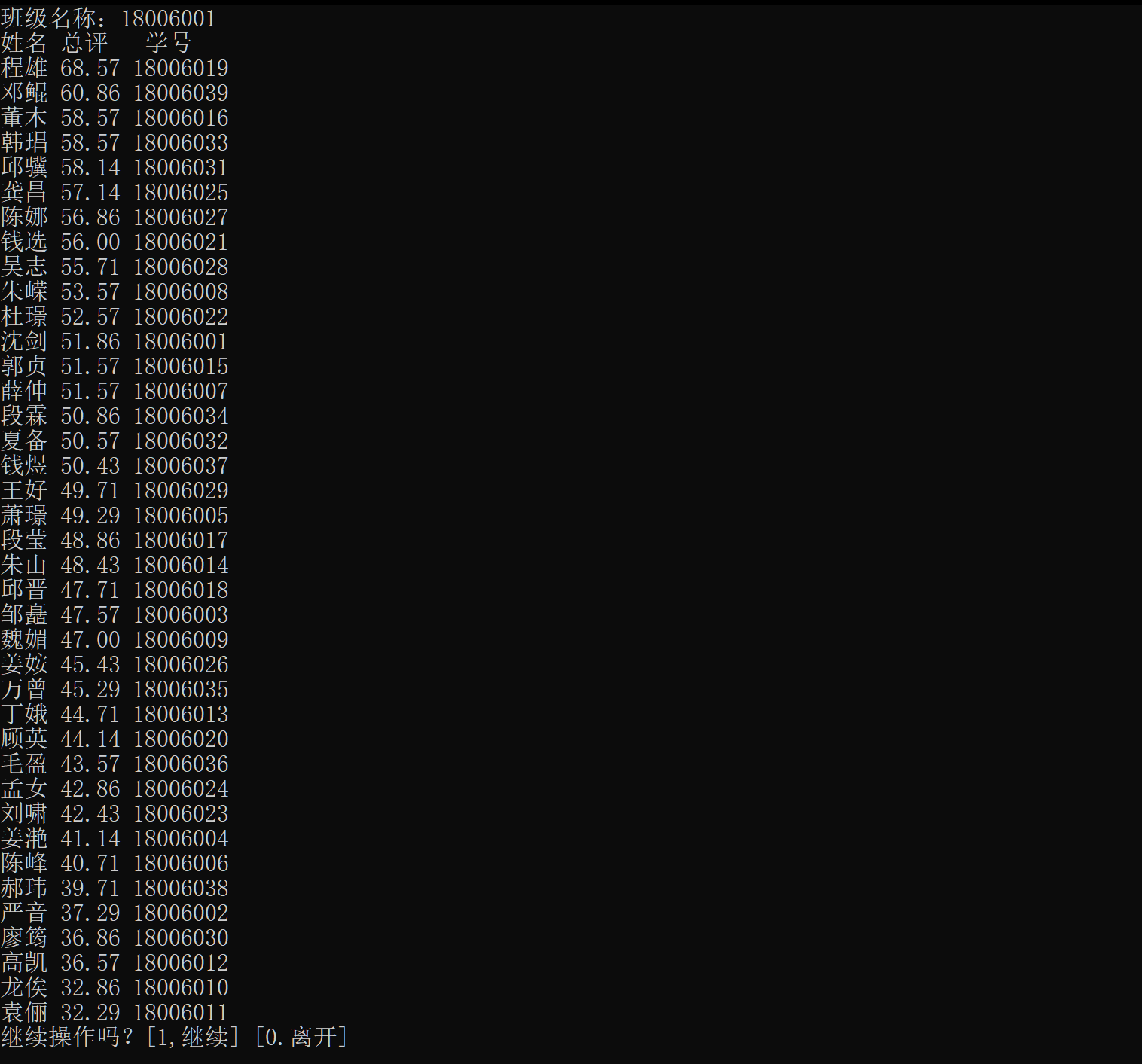




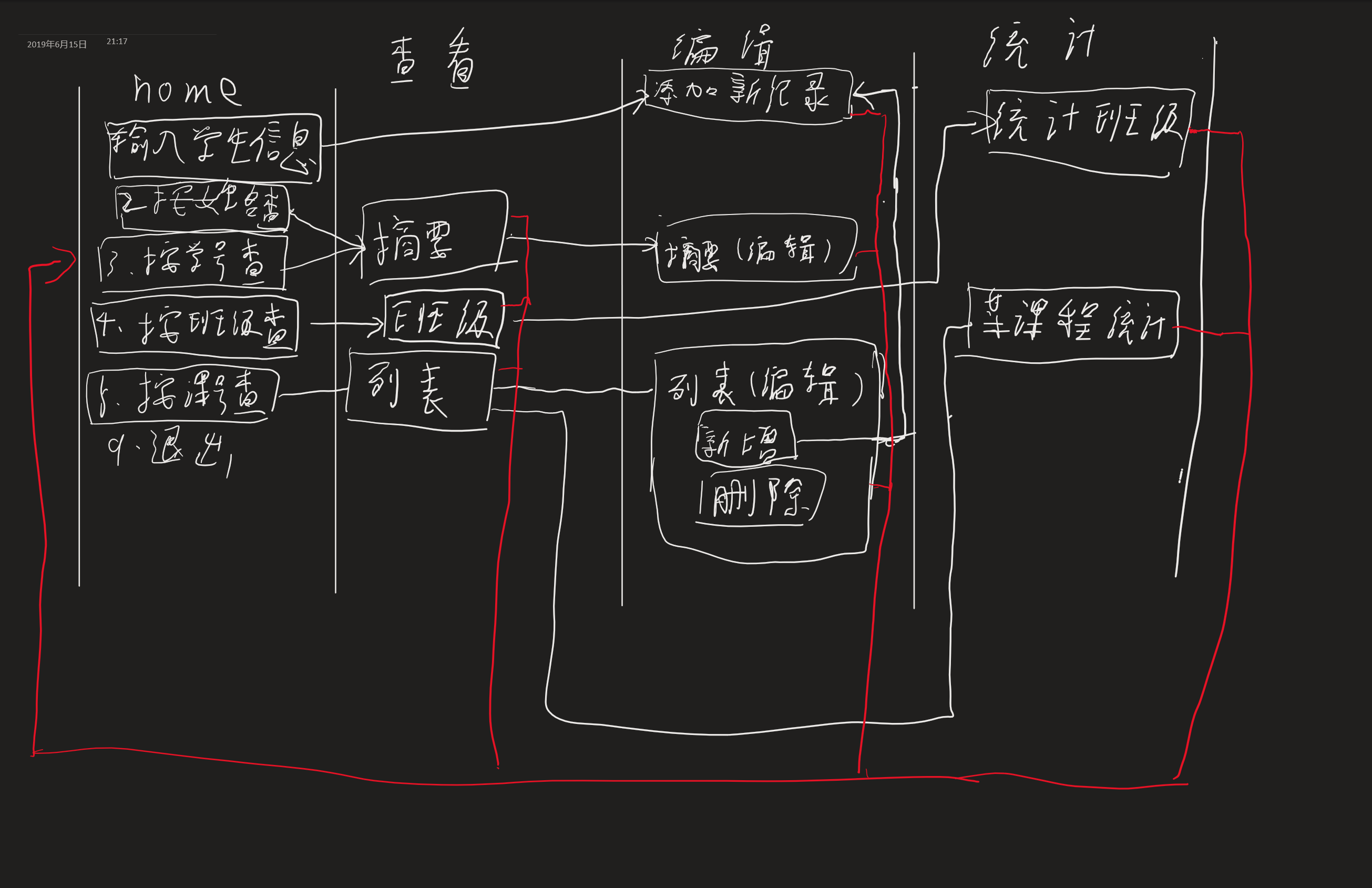
按照课号查询，排序并修改。



查看班级成员信息



程序流程图



|  |
| --- |
| **三．实验小结**  **先写好流程图，分析出哪些部分是可重复利用的，将其封装成函数，可以提高代码复用性，当出现问题时可以更加高效地分析问题所在。** |