

**课程设计报告**

**课程设计题目：《学生信息管理系统** 》

**学生姓名**：刘凌杰

**专 业：软件工程**

**班 级**：1721804

**指导教师：张军**

**2018年 6 月 12 日**

### **Part1:**

1、 类的组合

定义point类，数据成员包括x,y，成员函数包括构造函数，拷贝构造函数和析构函数，以及setx，getx，sety，gety四个属性函数。定义line类，端点由两个point类的对象组成，包括构造函数，析构函数以及计算线段长度的函数getlength。在main函数中，定义line的对象，并输出其长度。

**// ConsoleApplication2.cpp : 定义控制台应用程序的入口点。**

**//**

**#include<iostream>**

**#include<cmath>**

**using namespace std;**

**class point**

**{**

**private:**

**float x,y;**

**public:**

**point()**

**{**

**x=0;**

**y=0;**

**}**

**point(float x,float y):x(x),y(y)**

**{**

**}**

**point(point &a)**

**{**

**x=a.x;**

**y=a.y;**

**}**

**~point()**

**{**

**}**

**void setx(int x)**

**{**

**this->x=x;**

**}**

**void sety(int y)**

**{**

**this->y=y;**

**}**

**float getx()**

**{**

**return x;**

**}**

**float gety()**

**{**

**return y;**

**}**

**};**

**class line**

**{**

**private:**

**point p,q;**

**public:**

**line()**

**{**

**p=point(0,0);**

**q=point(0,0);**

**}**

**line(float px,float py,float qx,float qy)**

**{**

**p.setx(px);**

**p.sety(py);**

**q.setx(qx);**

**q.sety(qy);**

**}**

**~line()**

**{**

**}**

**double getlength()**

**{**

**return sqrt(pow(p.getx()-q.getx(),2)+pow(p.gety()-q.gety(),2));**

**}**

**};**

**int main()**

**{**

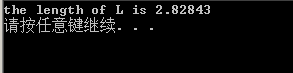
**line L(1,2,3,4);**

**cout<<"the length of L is "<<L.getlength()<<endl;**

**system("pause");**

**return 0;**

**}**

****

2、对象数组和函数

定义student类，数据成员包括姓名name和成绩score，成员函数包括构造函数，拷贝构造函数和析构函数。定义函数void highestscore(student s[]),输出分数最高的学生姓名和分数。在main函数中定义student s[N]，调用highestscore函数,输出分数最高的学生姓名和分数。

**// ConsoleApplication2.cpp : 定义控制台应用程序的入口点。**

**#include<iostream>**

**#include<string>**

**using namespace std;**

**int n=0;//记录学生人数**

**class student**

**{**

**private:**

**string name;**

**float score;**

**public:**

**student()**

**{**

**name="student";**

**score=0;**

**n++;**

**}**

**student(string name,float score):name(name),score(score)**

**{**

**}**

**student(student &a)**

**{**

**name=a.name;**

**score=a.score;**

**n++;**

**}**

**~student()**

**{**

**}**

**friend void highestscore(student s[]);**

**};**

**void highestscore(student s[])**

**{**

**int num;**

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

**{**

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

**{**

**if((s+j)->score>(s+j+1)->score)**

**{**

**num=(s+j)->score;**

**(s+j)->score=(s+j+1)->score;**

**(s+j+1)->score=num;**

**}**

**}**

**}**

**/\***

**for(int i=0;i<n;i++)**

**{**

**cout<<(s+i)->score<<" ";**

**cout<<endl;**

**}\*/**

**cout<<"the highest score is "<<s[n-1].score<<endl;**

**}**

**int main()**

**{**

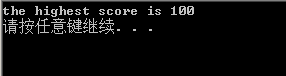
**student s[5]={student("",3),student("",100),student("",34),student("",68),student("",54)};**

**highestscore(s);**

**system("pause");**

**return 0;**

**}**

****

3、 静态数据成员

设计一个书类，能够保存书名、定价，所有书的本数和总价。（将书名和定价设计为普通数据成员；将书的本数和总价设计为静态数据成员）

**// ConsoleApplication2.cpp : 定义控制台应用程序的入口点。**

**#include<iostream>**

**#include<string>**

**using namespace std;**

**class book**

**{**

**private:**

**string name;**

**float price;**

**static float allprice,number;**

**public:**

**book()**

**{**

**name="book";**

**price=0;**

**number++;**

**}**

**book(string name,float price):name(name),price(price)**

**{**

**number++;**

**allprice+=price;**

**}**

**float getallprice()**

**{**

**return allprice;**

**}**

**float getnum()**

**{**

**return number;**

**}**

**};**

**float book::allprice=0;**

**float book::number=0;**

**int main()**

**{**

**book b1("book1",2),b2("book2",3.2),b3("book3",5.4);**

**cout<<b1.getallprice()<<endl;**

**cout<<b2.getnum()<<endl;**

**system("pause");**

**return 0;**

**}**

****

4、 动态内存分配

定义point类，数据成员包括x,y，成员函数包括构造函数，拷贝构造函数和析构函数，以及setx，getx，sety，gety四个属性函数。在main函数中，用new和delete分配和释放N个point的数组。（N是const常量，N=10）

**// ConsoleApplication2.cpp : 定义控制台应用程序的入口点。**

**//**

**#include<iostream>**

**#include<string>**

**using namespace std;**

**class point**

**{**

**private:**

**float x,y;**

**public:**

**point():x(0),y(0)**

**{**

**}**

**point(float x,float y):x(x),y(y)**

**{**

**}**

**point(point &a)**

**{**

**x=a.x;**

**y=a.y;**

**}**

**~point()**

**{**

**}**

**float getx()**

**{**

**return x;**

**}**

**float gety()**

**{**

**return y;**

**}**

**void setx(float x)**

**{**

**this->x=x;**

**}**

**void sety(float y)**

**{**

**this->y=y;**

**}**

**};**

**int main()**

**{**

**const int N=10;**

**point \*p= new point[N];**

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

**{**

**p[i].setx(i);**

**(p+i)->sety(i);**

**cout<<"("<<p[i].getx()<<","<<(p+i)->gety()<<")"<<endl;**

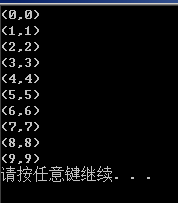
**}**

**delete []p;**

**system("pause");**

**return 0;**

**}**

****

5、 类的继承

定义一个point类，包含私有数据成员x，y，成员函数包括无参构造函数，带参构造函数，set和get属性函数。定义circle类，从point类公有派生，增加数据成员半径r，成员函数包括无参构造函数，带参构造函数，计算面积函数getarea。在main函数中定义一个circle的对象，并计算其面积。

**// ConsoleApplication2.cpp : 定义控制台应用程序的入口点。**

**#include<iostream>**

**using namespace std;**

**#define PI 3.1415926535**

**class point**

**{**

**private:**

**float x,y;**

**public:**

**point():x(0),y(0)**

**{**

**}**

**point(float x,float y):x(x),y(y)**

**{**

**}**

**point(point &a)**

**{**

**x=a.x;**

**y=a.y;**

**}**

**~point()**

**{**

**}**

**float getx()**

**{**

**return x;**

**}**

**float gety()**

**{**

**return y;**

**}**

**void setx(float x)**

**{**

**this->x=x;**

**}**

**void sety(float y)**

**{**

**this->y=y;**

**}**

**};**

**class circle:public point**

**{**

**private:**

**float radius;**

**public:**

**circle():point()**

**{**

**radius=0;**

**}**

**circle(float radius,float x,float y):point(x,y),radius(radius)**

**{**

**}**

**float area()**

**{**

**return radius\*radius\*PI;**

**}**

**};**

**int main()**

**{**

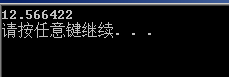
**circle c(2,2,2);**

**cout<<c.area()<<c.getx()<<c.gety()<<endl;**

**system("pause");**

**return 0;**

**}**

****

6、 虚基类

定义vehicle类，数据成员包括私有的weight，公有的构造函数，析构函数和输出函数dispaly；从vehicle类公有派生car类，增加数据成员载人数personnum，公有的构造函数,析构函数和输出display；从vehicle类公有派生truck类，增加数据成员载货量laod,公有的构造函数，析构函数和输出函数display；从car类和truck类共同公有派生出pickup类，包括公有的构造函数和输出函数。在main函数中，定义pickup类对象，并输出其基本信息。

**// ConsoleApplication2.cpp : 定义控制台应用程序的入口点。**

**#include<iostream>**

**using namespace std;**

**class vehicle**

**{**

**private:**

**int weight;**

**public:**

**vehicle()**

**{**

**weight=0;**

**}**

**vehicle(int w):weight(w)**

**{**

**}**

**~vehicle()**

**{**

**}**

**virtual int display()**

**{**

**return weight;**

**}**

**};**

**class car:virtual public vehicle**

**{**

**private:**

**int personnum;**

**public:**

**car():vehicle(),personnum(0)**

**{**

**}**

**car(int w,int per):vehicle(w),personnum(per)**

**{**

**}**

**~car()**

**{**

**}**

**int display()**

**{**

**return personnum;**

**}**

**};**

**class truck:virtual public vehicle**

**{**

**private:**

**int load;**

**public:**

**truck():vehicle()**

**{**

**load=0;**

**}**

**truck(int load,int w):vehicle(w),load(load)**

**{**

**}**

**~truck()**

**{**

**}**

**int display()**

**{**

**return load;**

**}**

**};**

**class pickup: virtual public car, virtual public truck**

**{**

**public:**

**pickup():car(),truck()**

**{**

**}**

**pickup(int load,int per,int w):truck(load,w),car(w,per)**

**{**

**}**

**int display()**

**{**

**cout<<"the personnum of car is "<<car::display()<<endl;**

**cout<<"the load of truck is "<<truck::display()<<endl;**

**return 0;**

**}**

**};**

**int main()**

**{**

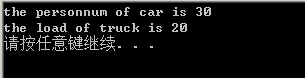
**pickup p(20,30,10);**

**p.display();**

**system("pause");**

**return 0;**

**}**

****

7、 运算符重载，友元函数和this指针

定义复数（complex ）类，其中至少包括实部和虚部，对复数类实现尽可能丰富的运算。

功能要求：

1）建立复数类

2）复数信息的初始化

3）复数信息的输出

4）将复数信息保存为文件

5）求复数的绝对值

6）实现复数的加、减、自加、自减等运算。

#include<iostream>

#include<cmath>

#include<fstream>

using namespace std;

class complex;

complex operator +(complex &a,complex &b);

complex operator -(complex &a,complex &b);

istream & operator>>(istream &is,complex &a);

ostream & operator<<(ostream &os,const complex &a);

class complex

{

private:

int real,image;

public:

complex()

{

real=0;

image=0;

}

complex(int r,int i):real(r),image(i)

{

}

friend istream & operator>>(istream &is,complex &a);

friend ostream & operator<<(ostream &os,const complex &a);

friend complex operator +(complex &a,complex &b);

friend complex operator -(complex &a,complex &b);

complex operator++(int);

complex operator++();

complex operator--(int);

complex operator--();

float fabs();//求复数的绝对值

void save();//将复数信息保存为文件

};

istream &operator>>(istream &is,complex &a)

{

cout<<"please input real and image:"<<endl;

is>>a.real>>a.image;

return is;

}

ostream &operator<<(ostream &os,const complex &a)

{

if(a.image>0)

os<<a.real<<"+"<<a.image<<"i"<<endl;

else

if(a.image<0)

os<<a.real<<a.image<<"i"<<endl;

else

os<<a.real<<endl;

return os;

}

complex operator +(complex &a,complex &b)

{

return complex(a.real+b.real,a.image+b.real);

}

complex operator -(complex &a,complex &b)

{

return complex(a.real-b.real,a.image-b.real);

}

complex complex::operator++(int)

{

complex a=\*this;

real++;

image++;

return a;

}

complex complex::operator++()

{

real++;

image++;

return \*this;

}

complex complex::operator--(int)

{

complex a=\*this;

real--;

image--;

return a;

}

complex complex::operator--()

{

real--;

image--;

return \*this;

}

float complex::fabs()

{

return sqrt(pow(real,2)+pow(image,2));

}

void complex::save()

{

ofstream ofs("D:\\complex.dat",ios::binary);

if(ofs)

{

ofs.write(reinterpret\_cast<char\*>(this),sizeof(\*this));

ofs.close();

cout<<"save successful!"<<endl;

}

else

cout<<"open fail!"<<endl;

}

int main()

{

complex c(1,2);

cout<<c;

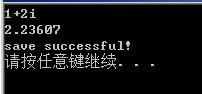
cout<<c.fabs()<<endl;

c.save();

system("pause");

return 0;

}



8、 虚函数和抽象类

定义一个抽象类shape，包括公有的计算面积area函数，计算体积volume函数，输出基本信息函数printinfo（三个函数均为纯虚函数）。从shape公有派生point类，增加私有数据成员x,y坐标，以及构造函数，析构函数。从point公有派生circle类，增加私有数据成员半径r，以及构造函数，析构函数。从circle公有派生cylinder类，增加私有数据成员高度h，以及构造函数，析构函数。（在定义三个派生类的过程中，自己考虑需要重定义哪个虚函数）。在main函数中，定义shape类的指针，指向派生类的对象，输出三类对象的基本信息，面积，体积；（将shape指针改为引用再尝试）。

#include<iostream>

using namespace std;

#define PI 3.1415926535

class shape

{

public:

virtual float area()=0;

virtual float volume()=0;

virtual void printinfo()=0;

};

class point:public shape

{

private:

float x,y;

public:

point()

{

x=0;

y=0;

}

point(float x,float y):x(x),y(y)

{

}

~point()

{

}

float area()

{

return 0;

}

float volume()

{

return 0;

}

void printinfo()

{

cout<<"("<<x<<","<<y<<")"<<endl;

}

};

class circle:public point

{

private:

float r;

public:

circle():r(0),point(0,0)

{

}

circle(float r):r(r),point(0,0)

{

}

circle(float r,float x,float y):r(r),point(r,y)

{}

~circle()

{

}

float get\_r()

{

return r;

}

float area()

{

return PI\*r\*r;

}

void printinfo()

{

cout<<"the radius is "<<r<<endl;

cout<<"circle's area is "<<area()<<endl;

}

};

class cylinder:public circle

{

private:

float h;

public:

cylinder():h(0),circle(0,0,0)

{

}

cylinder(float h, float r):h(h),circle(r)

{

}

cylinder(float h, float r,float x, float y):h(h),circle(r,x,y)

{

}

~cylinder()

{

}

float area()

{

return 2\*circle::area()+2\*PI\*get\_r()\*h;

}

float volume()

{

return circle::area()\*h;

}

void printinfo()

{

cout<<"the cylinder's high is "<<h<<endl;

cout<<"the cylinder's area is "<<area()<<endl;

cout<<"teh cylinder's volume is"<<volume()<<endl;

}

};

int main()

{

point po(2,2);

circle c(2);

cylinder cy(2,2);

shape \*p=&po;

p->printinfo();

p=&c;

p->printinfo();

p=&cy;

p->printinfo();

system("pause");

return 0;

}

/\*引用

int main()

{

point<float> po(2,2);

circle<float> c(2);

cylinder<float> cy(2,2);

shape &p=po;

p.printinfo();

shape &p1=c;

p1.printinfo();

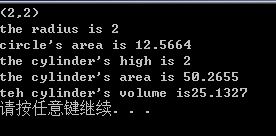
shape &p2=cy;

p2.printinfo();

system("pause");

return 0;

}\*/



9、模板

设计一个堆栈的类模板Stack，在模板中用类型参数T表示栈中存放的数据，用非类型参数MAXSIZE代表栈的大小。

#include<iostream>

using namespace std;

template<typename T>

class Stack

{

private:

int MAXSIZE;

T \*array;

int count;

public:

Stack()

{

MAXSIZE=0;

count=0;

array =new T[MAXSIZE];

}

Stack(int M)

{

MAXSIZE=M;

count=0;

array=new T[M];

}

~Stack()

{

delete []array;

cout<<"delete successfully!"<<endl;

}

void push(T i);

T top();

T pop();

bool empty();

bool fill();

};

//入栈

template<typename T>

void Stack<T>::push(T i)

{

if(fill())

cout<<"the Stack is full."<<endl;

else

{

array[count++]=i;

}

}

//抛栈

template<typename T>

T Stack<T>::top()

{

if(empty())

{

cout<<"the Stack is empty."<<endl;

return 0;

}

else

return array[count-1];

}

//栈顶

template<typename T>

T Stack<T>::pop()

{

if(empty())

{

cout<<"the Stack is empty."<<endl;

return 0;

}

else

{

return array[--count];

}

}

//判断栈是否满

template<typename T>

bool Stack<T>::fill()

{

return count==MAXSIZE;

}

//判断栈是否为空栈

template<typename T>

bool Stack<T>::empty()

{

return count==0;

}

int main()

{

Stack<int> s(10);

for(int i=1;i<=10;i++)

{

s.push(i);

}

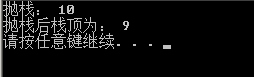
cout<<"抛栈： "<<s.pop()<<endl;

cout<<"抛栈后栈顶为： "<<s.top()<<endl;

system("pause");

return 0;

}



10 文件读写

定义学生类数组，有N个人（N=5），包括姓名和语数外三名课的成绩，通过重载<<和>>运算符实现学生数组的文件读写。

#include<iostream>

#include<fstream>

#include<string>

using namespace std;

#define N 5

class Student;

ifstream& operator>>(ifstream &ifs,Student a[]);

ofstream& operator<<(ofstream &ofs,Student a[]);

class Student

{

private:

string name;

float math,chinese,english;

public:

Student()

{

name="Student";

math=0;

chinese=0;

english=0;

}

Student(string n,float m,float c,float e)

{

name=n;

math=m;

chinese=c;

english=e;

}

void printinfo()

{

cout<<name<<endl;

cout<<math<<endl;

cout<<chinese<<endl;

cout<<english<<endl;

}

friend ifstream& operator>>(ifstream &ifs,Student a[]);

friend ofstream& operator<<(ofstream &ofs,Student a[]);

};

ifstream& operator>>(ifstream &ifs, Student a[])

{

ifs.open("D:\\student.dat",ios::binary);

if(ifs)

{

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

{

ifs.read(reinterpret\_cast<char \*>(a+i),sizeof(a[i]));

// a[i].printinfo();

}

cout<<"read successful!"<<endl;

ifs.close();

}

return ifs;

}

ofstream& operator<<(ofstream &ofs,Student a[])

{

ofs.open("D:\\student.dat",ios::binary);

if(ofs)

{

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

{

// a[i].printinfo();

ofs.write(reinterpret\_cast<char \*>(a+i),sizeof(a[i]));

}

cout<<"write successful!"<<endl;

ofs.close();

}

return ofs;

}

int main()

{

Student s[N]={Student("liu",66,66,66),Student("ling",77,77,77),Student("jie",88,88,88),Student("rao",99,99,99),Student("guang",100,100,100)};

ofstream fs;

fs<<s;

Student a[5];

ifstream ifs;

ifs>>a;

for(int i=0;i<5;i++)

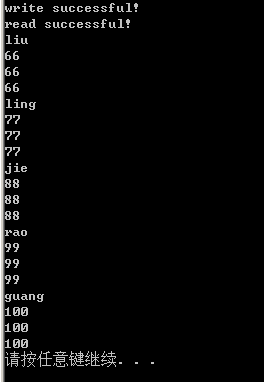
{

a[i].printinfo();

}

system("pause");

return 0;

}

### **Part2:**

### **一、题目：**5学生信息管理

基本要求：

定义学生（student）类，其中至少包括姓名、性别、学号、班级和联系电话。

功能要求：

1、设计菜单实现功能选择；

2、输入功能：输入学生信息，并保存到文件中；

3、能按照学生的学号排序

4、能根据学号修改学生信息

5、能根据学号删除学生信息

6、查询功能：

**二、内容和目的**

1、设计菜单实现功能选择；

2、输入功能：输入学生信息，并保存到文件中；

3、能按照学生的学号排序

4、能根据学号修改学生信息

5、能根据学号删除学生信息

6、查询功能：

**三、设计过程**

**1设计思路：**

* **首定义一个学生数组a[100]用来存放学生信息，之后的所有功能实现都是与这个数组互动；定义一个记录学生总数N，在程序运行开始用number()获取文件里学生人数N；**
* **写一个将数组a[100]信息保存在文件的函数outinformation()和将文件信息读入到数组a[100]的函数；在程序运行开始和结束之前分别调用number（）和outinformation（）达到文件信息更新的目的；**
* **用于while函数达到反复使用功能的目的；**

**用switch函数选择功能；**

**子函数之间相互调用，递归调用，在一个函数中实现功能；**

**点击跳入对应函数**

学号修改2

名字修改

性别修改

班级修改

电话修改

**0、退出**

输出

读入

**程序结构图**

* **Student类：**

class student

{

private:

char sex[10],Class[30],phone[20],name[20];

long long int number;

int num;

public:

student(char \*na="student",char \*se="男",char \*C="0000000",char \*ph="00000000000",long long int nu=0)

{

strcpy\_s(name,20,na);

strcpy\_s(sex,10,se);

strcpy\_s(Class,30,C);

strcpy\_s(phone,20,ph);

number=nu;

}

void alter(int n); //信息修改

friend void Classnum(); //获得班级信息

friend void ranking(); //学生学号排名

friend void classnum(); //查询各班人数

friend void seekname(); //姓名查询

friend void seeksex(); //性别查询

friend void seekclass(); //班级查询

friend void seekphone(); //电话查询

friend void seeknumber(); //学号查询

friend istream &operator >>(istream &is,student &a);

friend ostream &operator <<(ostream &os,student &a);

}; 姓名name

班级Class

**私有成员** 性别sex

电话phone

学号number

信息修改alter(int n);

获取班级信息Classnum()

学生学号排名ranking()

**Student 类** 查询各班人数classnum()

姓名查询seekname()

性别查询seeksex()

**公有成员** 班级查询seekclass()

电话查询seekphone()

学号查询seeknumber()

键盘输入>>重载

屏幕输出<<重载

* **文件：**

**储存到文件student.dat**

void outinformation()

{

ofstream ofs;

ofs.open("D:\\student.dat",ios::binary|ios::out);

if(ofs)

{

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

ofs.write(reinterpret\_cast<char \*>(a+i),sizeof(a[i]));

ofs.close();

}

}

**将文件信息读入到数组a,并获得文件里学生数量N：**

**用eof()函数读到文件尾部，每读到一个学生信息记录到a数组总，同时定义一个i来记录读取次数；读完后N=i-1则代表文件学生个数。**

void Number()

{

int i=0;

ifstream ifs;

ifs.open("D:\\student.dat",ios::binary|ios::in);

if(ifs)

{

while(!ifs.eof())

{

ifs.read(reinterpret\_cast<char \*>(a+i),sizeof(a[i]));

i++;

}

ifs.close();

}

N=i-1;

}

* **添加：**

**重载学生对象输入>>；每输入一个学生信息，总人数N增一；输入完一次用i判断是否继续添加，如i=’0’则回到调用主界面函数Interface();**

void input()

{

char i='1';

cin>>a[N];

N++;

cout<<"按任意键继续添加（退出请按0）"<<endl;

getchar();

i=getchar();

system("cls");

if(i=='0')

Interface();

else

input();

}

//重载>>

istream &operator >>(istream &is,student &a)

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 输 入 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"姓名:";

is>>a.name;

cout<<"性别（男/女，man/woman）:";

while(1)

{

is>>a.sex;

if(a.sex=="man"||a.sex=="woman"||a.sex=="男"||a.sex=="女")

break;

else

cout<<"请重新输入性别:"<<endl;

}

cout<<"班级:";

is>>a.Class;

cout<<"电话:";

is>>a.phone;

cout<<"学号:";

is>>a.number;

cout<<"输入完成！"<<endl;

return is;

}

* **查找：**

**该函数是查找功能的总汇，包含了各班人数查询和学生信息查询：**

**运用了switch()语句；按1学生信息查询；按2各班人数查询；按0返回主界面；如果没有学生则输出“无任何学生信息”**

void seek()

{

if(N!=0)

{

int i;

while(1)

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 1、 学生信息查询"<<endl;

cout<<" 2、 各班人数查询"<<endl;

cout<<" 0、 返回功能界面"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入序号选择查询："<<endl;

cin>>i;

system("cls");

switch(i)

{

case 1:studentseek();break;

case 2:classnum();break;

case 0:Interface();

default:

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 输入序号无效!"<<endl;pause();

}

}

}

else

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 无任何学生信息！"<<endl;

pause();

}

}

**查询各班人数：**

**首先获取所有班级名称：**

**用classname[30]存放已知班级名称，cnum记录班级总数；**

**定义逻辑型变量judge=true，用for（）将每个学生的班级名称与与已知班级进行对比，如果相同fudge=false;在if(judge)里添加加班级名称：如果班级名称不重复（judge==true）则记录班级名称到classname,同时cnum增一；反之不添加班级姓名。**

**获取各班人数：**

**写一个外循环循环班级名称，写一个内循环循环所有学生；**

**内循环中如学生的班级名称与班级相同，该班级学生数量（用num记录）加一，内循环一次将该班级（classname[i]）与其学生数量(num)输出；**

//查询各班学生数量

void classnum()

{

Classnum();

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

for(int j=0;j<cnum;j++)

{

int num=0;

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

{

if(a[i].Class==classname[j])

num++;

}

cout<<" （"<<classname[j]<<"）班的学生数量为 "<<num<<endl;

}

pause();

}

**查找学生信息：**

**该函数是查找学生信息，调用该函数进行选择信息查找学生信息，也可以筛选部分学生：**

**运用了switch()语句进行选择输入部分信息来查询该学生；按1调用学号查找函数seeknumber(),输入学号查找学生；按2调用姓名查找函数seekname(),输入姓名查找学生；按3调用电话查找函数phone(),输入电话查找学生；按4班级查找函数seekclass(),输入班级查找或筛选学生；按5调用性别查找函数，输入性别查找或筛选学生；否则显示“ 输入序号无效！”**

void studentseek()

{

int i;

while(1)

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"1、学号 2、姓名 3、 电话 4、班级 5、性别 "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请根据序号输入信息查找学生(返回请按0)："<<endl;

cin>>i;

system("cls");

switch(i)

{

case 1:seeknumber();break;

case 2:seekname();break;

case 3:seekphone();break;

case 4:seekclass();break;

case 5:seeksex();break;

case 0:seek();

default:

cout<<"输入序号无效！"<<endl;

}

}

}

void Classnum()

{

cnum=0;

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

{

if(i!=0)

{

bool judge=true; //记录班级是否重复

for(int j=0;j<cnum;j++)

{

if(a[i].Class==classname[j])

judge=false;

}

if(judge)

{

classname[cnum++]=a[i].Class;

}

}

else

{

classname[cnum++]=a[i].Class;

}

}

}

//学号查找

void seeknumber()

{

long long int numfind;

bool n=true;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入你要查找学生的学号："<<endl;

cin>>numfind;

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

{

if(numfind==a[i].number)

{

cout<<"你要查找的学生信息如下："<<endl;

cout<<a[i];

n=false;

}

}

if(n)

cout<<"不存在该学生！"<<endl;

pause();

}

//姓名查找

void seekname()

{

string namefind;

bool n=true;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入你要查找学生的姓名："<<endl;

cin>>namefind;

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

{

if(namefind==a[i].name)

{

cout<<"你要查找的学生信息如下："<<endl;

cout<<a[i];

n=false;

}

}

if(n)

cout<<"不存在该学生！"<<endl;

pause();

}

//电话查找

void seekphone()

{

string phonefind;

bool n=true;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入你要查找学生的电话："<<endl;

cin>>phonefind;

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

{

if(phonefind==a[i].phone)

{

cout<<"你要查找的学生信息如下："<<endl;

cout<<a[i];

n=false;

}

}

if(n)

cout<<"不存在该学生！"<<endl;

pause();

}

//班级筛选

void seekclass()

{

Classnum();

string classfind;

bool n=true,m=false;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入你要查找学生的班级："<<endl;

cin>>classfind;

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

{

if(classfind==classname[i])

m=true;

}

if(m)

{

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

{

if(classfind==a[i].Class)

{

cout<<"你要查找的学生信息如下："<<endl;

cout<<a[i];

n=false;

}

}

if(n)

cout<<"不存在该学生！"<<endl;

}

else

cout<<"不存在该班级！"<<endl;

pause();

}

//性别筛选

void seeksex()

{

string sexfind;

bool n=true;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查 询 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入你要查找学生的性别："<<endl;

cin>>sexfind;

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

{

if(sexfind==a[i].sex)

{

cout<<"你要查找的学生信息如下："<<endl;

cout<<a[i];

n=false;

}

}

if(n)

cout<<"不存在该学生！"<<endl;

pause();

}

void seek();

* **修改：**

**该函数是选择学生进行修改：**

**首先调用students()将所有学生信息显示，再输入序号i进行选择学生（i等于0时调用回到主界面函数**Information()）,i在学生总数内则调用该学生对象的修改信息函数alter();

**alter()函数里用了switch()函数对学生对象进行选择信息修改；用while循环修改；如要退出按0，switch()调用alter1()回到主修改界面**

void alter1()

{

int i;

if(N!=0)

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 修 改 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

students();

cout<<"\n请输入要修改学生的序号(退出请按0）："<<endl;

cin>>i;

if(i==0)

{

system("cls");

Interface();

}

system("cls");

if(i<=N&&i>0)

{

a[i-1].alter(i);

}

else

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 修 改 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 你输入的序号不在学生总数范围之内！"<<endl;

pause();

alter1();

}

}

else

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 修 改 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 无任何学生信息！"<<endl;

pause();

}

}

该函数

//学生对象信息修改

void student::alter(int n)

{

char i;

while(1)

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 修 改 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 该学生信息"<<endl;

cout<<a[n-1];

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 1、姓名 2、 性别 3、 班级 4、电话 5、 学号 "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入序号选择你要修改的学生信息（返回请按0)："<<endl;

cin>>i;

switch(i)

{

case '1':cout<<"请输入你要改成的名字： ";cin>>name;cout<<"\n名字修改完成"<<endl;break;

case '2':cout<<"请输入你要改成的性别： ";cin>>sex;cout<<"\n性别修改完成"<<endl;break;

case '3':cout<<"请输入你要改成的班级： ";cin>>Class;cout<<"\n性别修改完成"<<endl;break;

case '4':cout<<"请输入你要改成的电话： ";cin>>phone;cout<<"\n电话修改完成"<<endl;break;

case '5':cout<<"请输入你要改成的学号： ";cin>>number;cout<<"\n学号修改完成"<<endl;break;

case '0':system("cls");alter1();

default:

cout<<"输入序号无效！"<<endl;

}

pause();

}

}

* **删除：**

**删除学生对象：**

**该函数将所有学生信息显示出来，输入i进行选择学生对象修改；如需删除第i个学生，用for循环将i之后的所有学生对象给前一个对象赋值，在总人数N减一达到删除目的；用while()函数进行循环选择删除，在while ()里用if判断i是否为‘1’，为“1”退出循环进入主界面；**

void expurgate()

{

if(N!=0)

{

int i;

while(1)

{

if(N==0)

{

Interface();

}

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 删 除 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

students();

cout<<"请输入你要删除学生的序号(退出请按0）："<<endl;

cin>>i;

if(i<=N&&i>0)

{

for(int j=i;j<=N-1;j++)

{

a[j-1]=a[j];

}

N--;

system("cls");

students();

cout<<"删除成功！请按任意键继续"<<endl;

getchar();

getchar();

system("cls");

}

else

{

if(i==0)

{

system("cls");

Interface();

}

else

{

cout<<"你输入的序号不在学生总数范围之内！"<<endl;

pause();

expurgate();

}

}

}

}

else

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 删 除 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 无任何学生信息！"<<endl;

pause();

}

}

* **排位**

对学生进行学号排位：

用冒泡排序法对学生对象进行排位；如N等于0则输出“无任何学生信息！”

//根据学号将学生排列

void ranking()

{

if(N!=0)

{

student t;

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

{

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

{

if(a[j].number>a[j+1].number)

{

t=a[j];

a[j]=a[j+1];

a[j+1]=t;

}

}

}

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 排 序 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

students();

cout<<"排序成功！请按任意键继续"<<endl;

getchar();getchar();

}

else

{

cout<<"无任何学生信息！"<<endl;

pause();

}

}

* **显示所有信息：**

**重载类输出符号>>,再用for循环将所有学生对象信息输出；**

//所有学生信息输出

void students()

{

if(N==0)

cout<<" 无学生信息！"<<endl;

else

{

cout<<" 所有学生信息"<<endl;

for(int j=0;j<N;j++)

{

cout<<a[j];

}

}

}

//重载<<

ostream &operator <<(ostream &os,student &a)

{

os<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

os<<" 学号: "<<a.number<<endl;

os<<" 姓名: "<<a.name<<endl;

os<<" 性别: "<<a.sex<<endl;

os<<" 班级: "<<a.Class<<endl;

os<<" 电话: "<<a.phone<<endl;

os<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

return os;

* **功能界面：**

将功能选择写在主界面，用swich()函数调用对应功能函数：1.添加函数input()2、查询函数seek()3、修改函数alter()4、删除函数expurgate()5、排名ranking()6、输出所有函数()0、退出；用while()循环调用功能；

void Interface()

{

int i;

while(1)

{

cout<<" 欢迎进入学生信息管理系统"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 1、 添加学生信息"<<endl;

cout<<" 2、 查询学生信息"<<endl;

cout<<" 3、 修改学生信息"<<endl;

cout<<" 4、 删除学生信息"<<endl;

cout<<" 5、 按学号将学生信息排序"<<endl;

cout<<" 6、 输出所有学生信息"<<endl;

cout<<" 0、 退出"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入你想要操作的序号："<<endl;

cin>>i;

system("cls");

switch(i)

{

case 1:input();break;

case 2:seek();break;

case 3:alter1();break;

case 4:expurgate();break;

case 5:ranking();break;

case 6:students();pause();break;

case 0:outinformation();exit(0);break;

default:

cout<<"输入序号无效！"<<endl;pause();

}

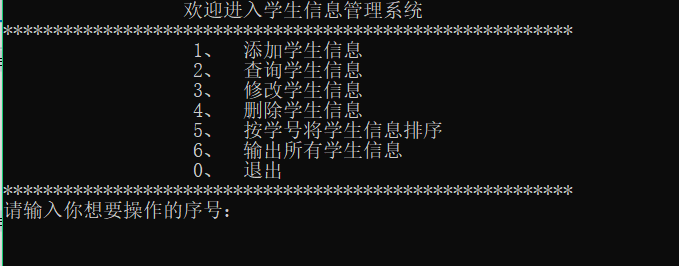
system("cls");

}

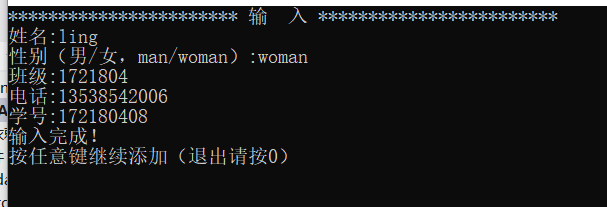
}

**四、运行结果分析**

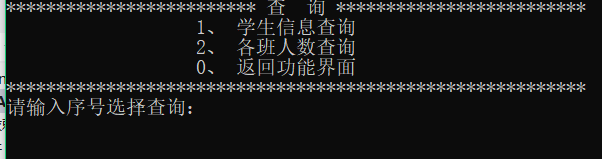
**1．运行程序**

****

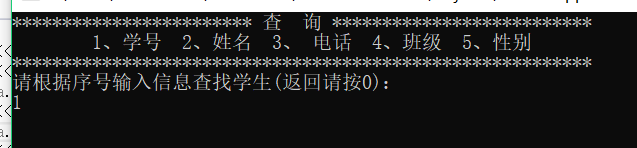
**2．输入1添加学生，输入学生信息**

****

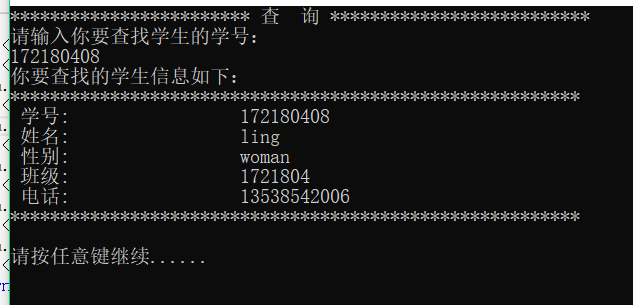
**3．按0退回到主界面选择2，进入查询功能**

****

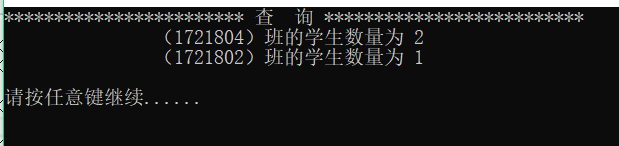
**4.选择1学生信息查询，按1学号查询**

****

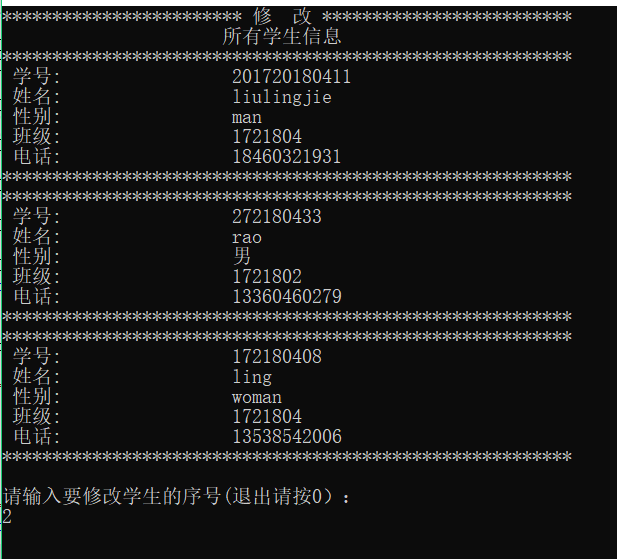
**5.输入学号，查找出学生信息**

****

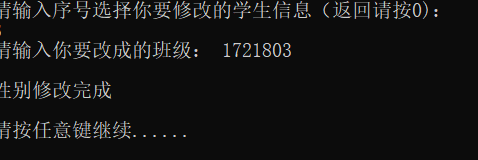
**6.按0返回查询界面，选择2班级人数查询**

****

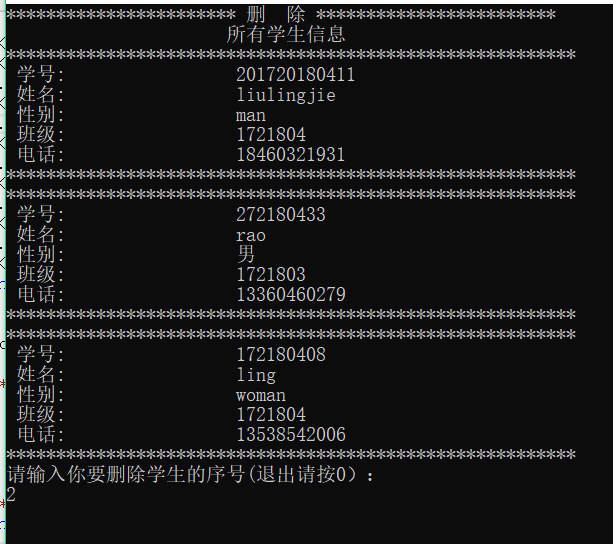
**7.返回主界面选择3修改功能，选择你要修改学生的序号，再选择要修改的信息进行修改，修改完成选择返回主界面**

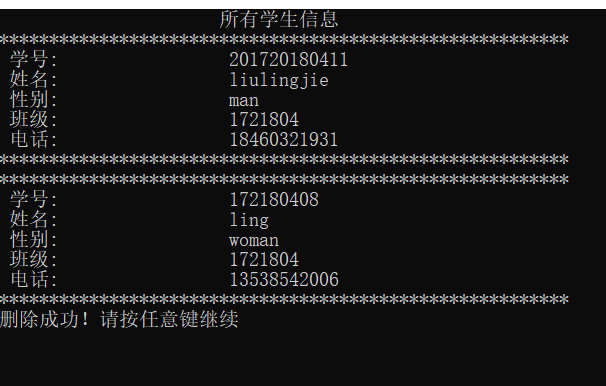
****

****

****

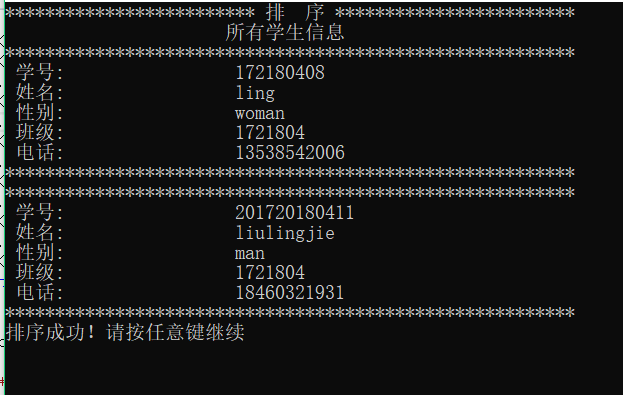
**8.选择输入4进行删除，选择序号2，确认删除成功，按0退出**

****

****

**9.回到主界面选择5进行学号排序和输出，按任意键返回；**

**选择6输出所有学生信息，在按0退出程序。**

****

**五、总结及体会**

* **在写程序前构思一个整体框架；所有功能实现都写在子函数中；最后用main()函调用；**
* **涉及知识点多，要灵活运用；**
* **界面先不用在意细节，整体写完再美观界面；**