项目说明文档

数据结构课程设计

——考试报名系统

作 者 姓 名： 陈君

学 号： 2250420

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

# 1 分析

## 1.1 背景分析

考试报名系统是一个学校不可缺少的部分，它对于学校的管理者和学生来说都至关重要，所以一个良好的考试报名系统应该能够为用户提供充足的信息和功能。考试报名系统对于学校加强考试管理有极其重要的作用。随着学生数量和考试数量的日益庞大，如何管理如此庞大的数据显得极为复杂，传统的手工管理工作量大且容易出错。

随着计算机科学技术的不断成熟，使用计算机对考试报名系统进行管理，具有手工管理所无法比拟的优势。这些优点能够极大地提高学校和学生的效率，也是学校走向信息化、科学化、国际化的重要条件。因此，开发一套考试报名系统具有十分重要的意义。

## 1.2 功能分析

作为一个最简易的考试报名系统，首先应该有的功能就是输入同学们的考试报名情况并且可以予以显示。其次，考试报名系统还应该具有插入、删除、修改功能，以保证同学可以随时更改自己的考试报名情况。最后，考试报名系统软件还应该确保软件可以正常关闭。

综上所述，一个考试报名系统至少应该具有输入、输出、插入、删除、修改、退出的功能。设计（改进方法，使用树这个数据结构，将学号按照升序顺序排列，找的时候按照二分查找的方法，可以提高效率。）

## 2.1 数据结构设计

如上功能分析所述，该系统要求大量的增加、删除、修改操作，而链表进行增加、删除等操作十分简便，因此考虑使用链表数据结构。同时，为了实现简易，在第一个结点之前附加一个头结点，这样就使得增加或者删除头结点与处理其他结点方法相同，使得程序简洁。

## 2.2 类结构设计

经典的链表一般包括两个抽象数据类型（ADT）——链表结点类（LNode）与链表类（LinkList），而两个类之间的耦合关系可以采用嵌套、继承等多种关系。为方便处理，本系统采用设计listnode和List两个类，并且把List作为listnode类的友元，这样使得链表类（List）可以访问链表结点。

## 2.3 成员与操作设计

class List;

class listnode {

public:

char number[100] = {0};//考号

char name[100] = { 0 };//姓名

char sex[100] = { 0 };//性别

int age;//年龄

char creear[100] = { 0 };//职业

listnode\* next; //结点数据

friend List;

/\*\*\*重载运算符,直接输入输出listnode\*\*\*/

friend ostream& operator<<(ostream& os, listnode& list1)//重载左移运算符，一般要用友元函数，然后定义在全局中

{

return os << setw(8) << setiosflags(ios::left) << list1.number << setw(8) << list1.name

<< setw(8) << list1.sex << setw(8) << list1.age << setw(8) << list1.creear << endl;

}

friend istream& operator>>(istream& input, listnode& list1)

{

return input >> list1.number >> list1.name >> list1.sex >> list1.age >> list1.creear;

}

listnode()

{

age = 0;

next = NULL;

}

};

/\*

\*链表类，存储每个结点

\*/

class List

{ //链表类

public:

List(int n);//构造函数

List();//默认构造函数

void putout();//提取器

int insert(char str[100]);//插入功能

int delete\_node(char str[100]);//删除功能

void reset(char str[100]);//修改功能

int getsize() {

return size;

}//获取链表大小

void search(char str[100]);//查找功能

listnode\* getfirst() { return first; }

~List() {

delete first;

}

//LinkList的析构函数，通过调用MakeEmpty()实现对内存的回收

private:

listnode\* first, \* last; //表头和表尾指针

int size;

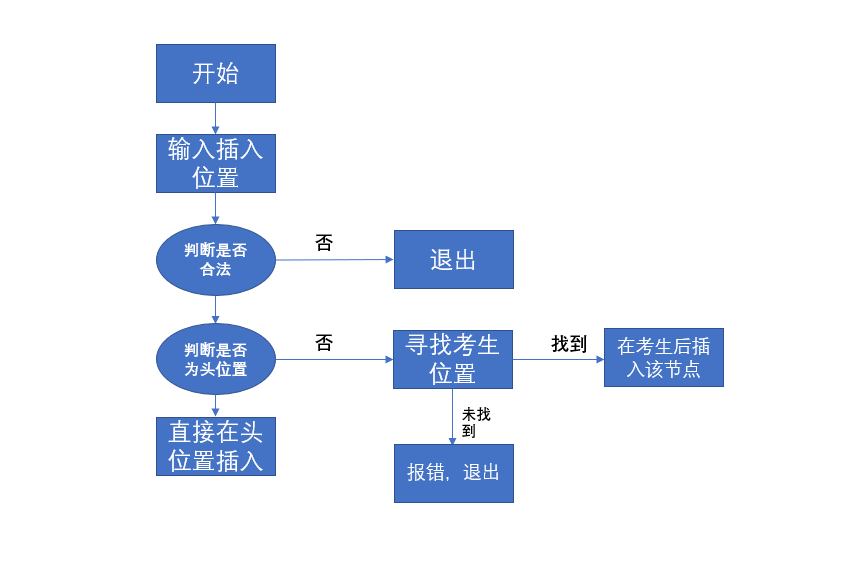
## };2.4 系统设计

系统给出提示“首先请建立考生信息系统！”，然后提示用户输入初始化的考生人数“请输入考生人数”，接着调用List的构造函数，提示用户输入考生数据，然后询问用户接下来的操作，可以进行的操作有1为插入，2为删除，3为查找，4为修改，5为统计,0为取消操作，当用户输入0后，释放链表的内存，然后自动退出系统。

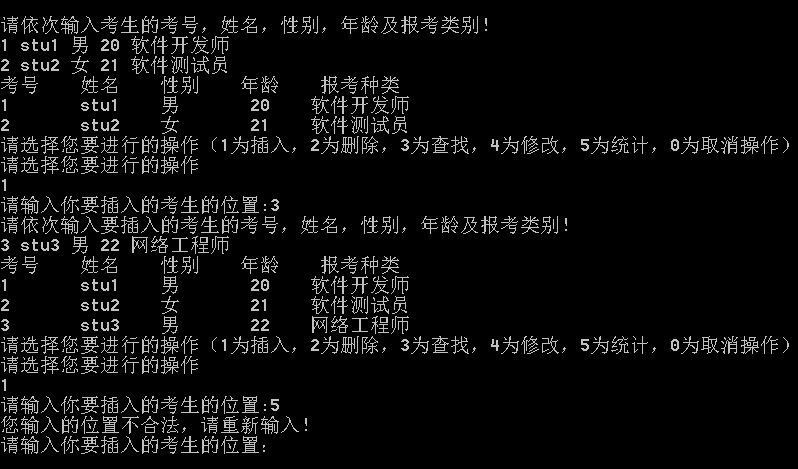
# 3 实现

## 3.1 插入功能的实现

### 3.1.1 插入功能流程图



3.1.3 插入功能截屏示例



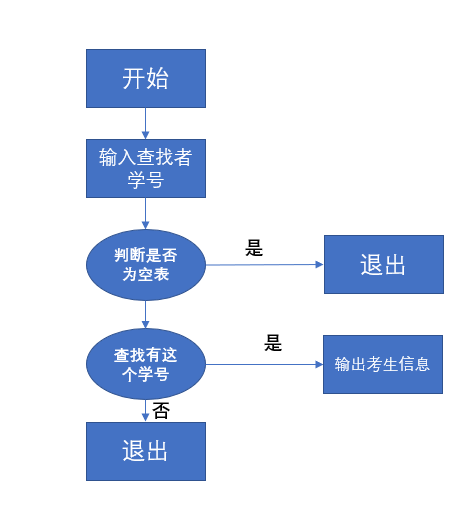
## 3.2 删除功能的实现

### 3.2.1 删除功能流程图

### }3.2.3 删除功能截屏示例

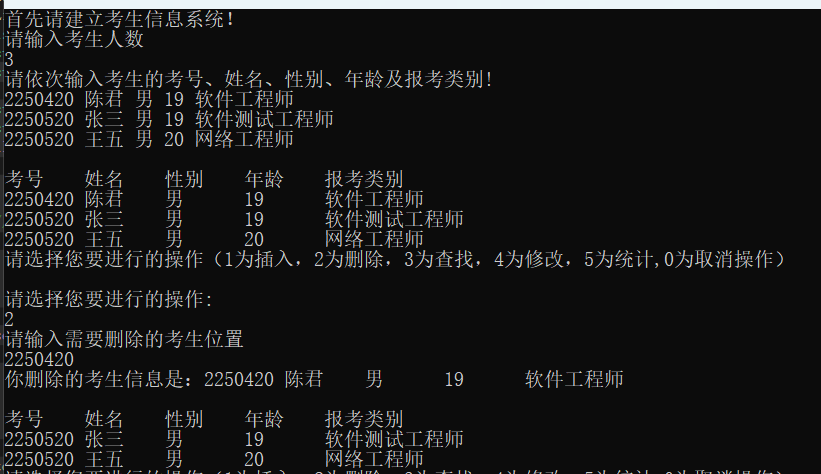
## 3.3 查找功能的实现

### 3.3.1 查找功能流程图



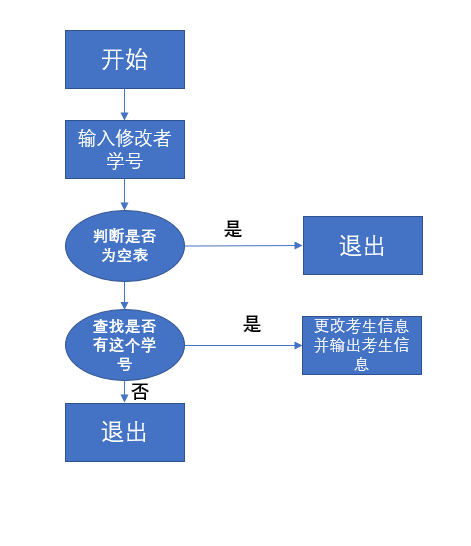
### 

### }3.3.3 查找功能截图示例



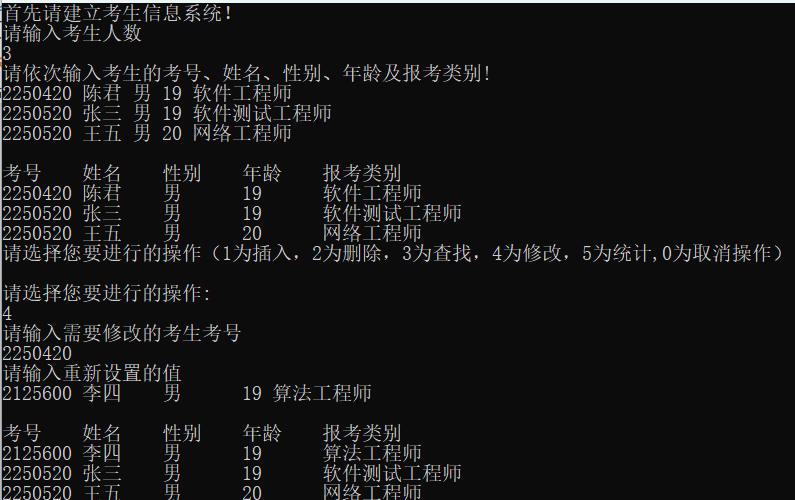
## 3.4 修改功能的实现

### 3.4.1 修改功能流程图



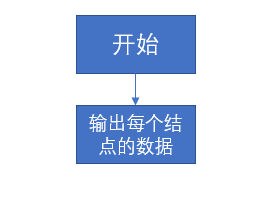
### 

### 3.4.3 修改功能截屏示例

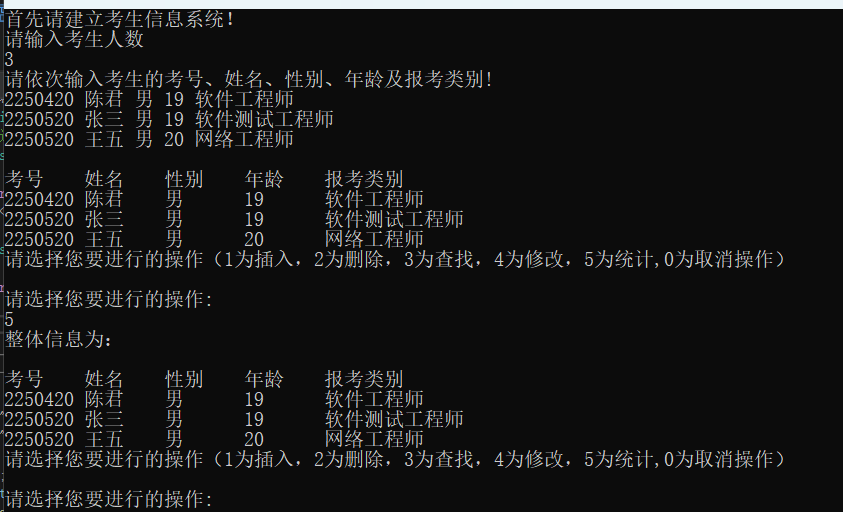


## 3.5 统计功能的实现

### 3.5.1 统计功能流程图

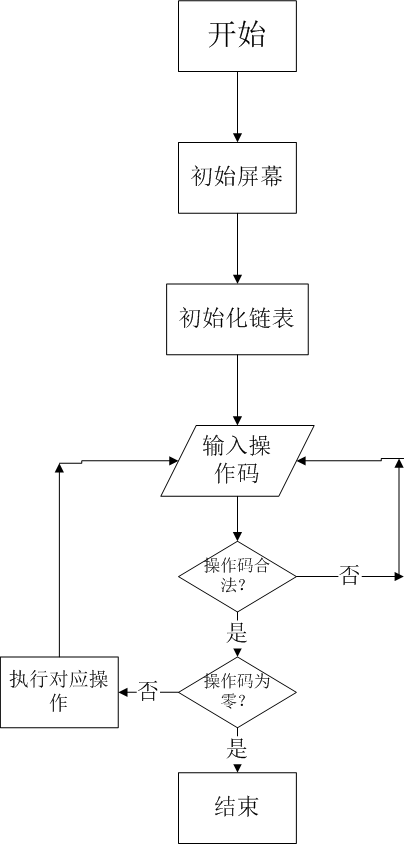


3.5.3 统计功能截屏示例

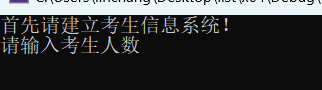


## 3.6 总体系统的实现

### 3.6.1 总体系统流程图



### }3.6.3 总体系统截屏示例



4 **测试**

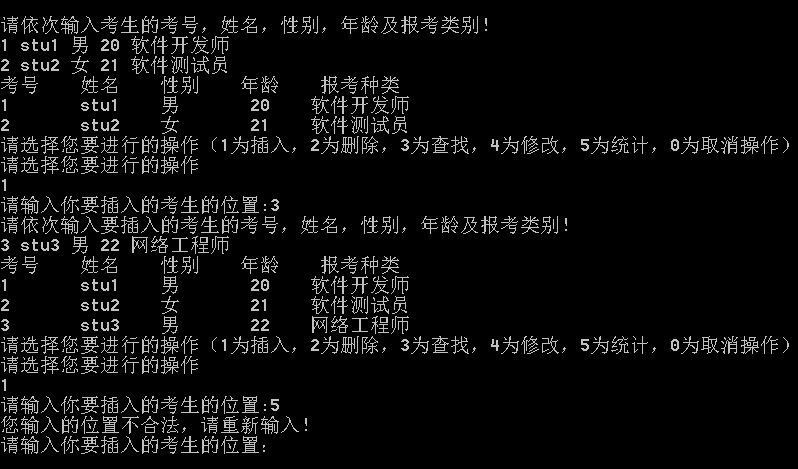
## 4.1 功能测试

### 4.1.1 插入功能测试

**测试用例**：

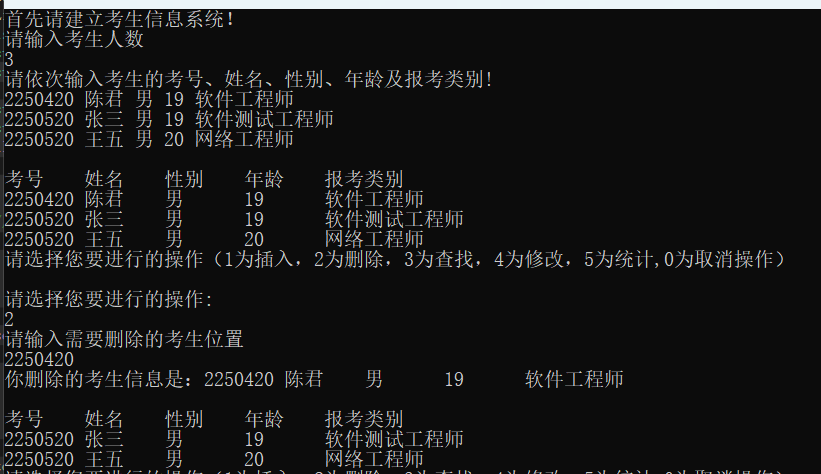
2250420 陈君 男 19 软件工程师

2250520 张三 男 19 软件测试工程师

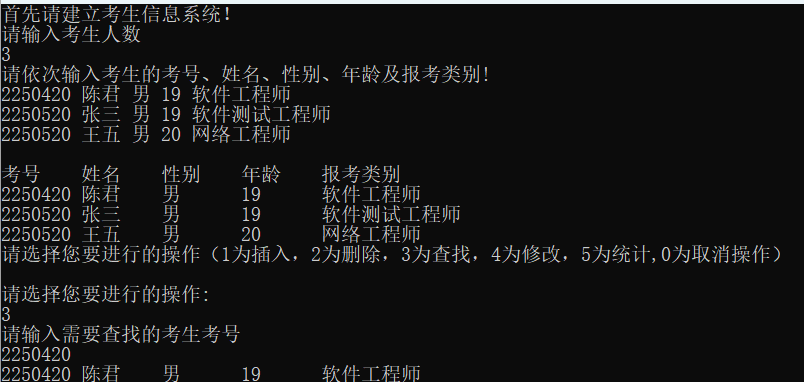
**实验结果**

### 4.1.2 删除功能测试

**实验结果：**



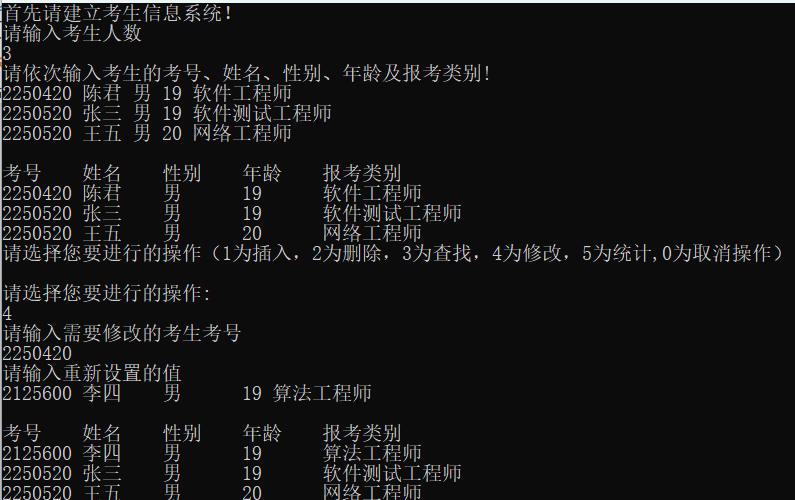
### 4.1.3 查找功能测试

**实验结果：**

### 4.1.4 修改功能测试

**测试用例：**将考号1修改为性别女，年龄20，报考种类移动开发员。

**实验结果：**



### 4.1.5 统计功能测试

**测试用例：**统计当前数据

**预期结果：**

1 stu1 女 20 移动开发员

2 stu2 女 21 软件测试员

3 stu3 男 22 网络工程师

**实验结果：**



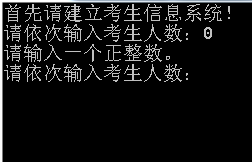
## 4.2 边界测试

### 4.2.1 初始化无输入数据

**测试用例：**初始无输入数据

**预期结果：**给出错误提示，程序运行正常不崩溃。

**实验结果：**

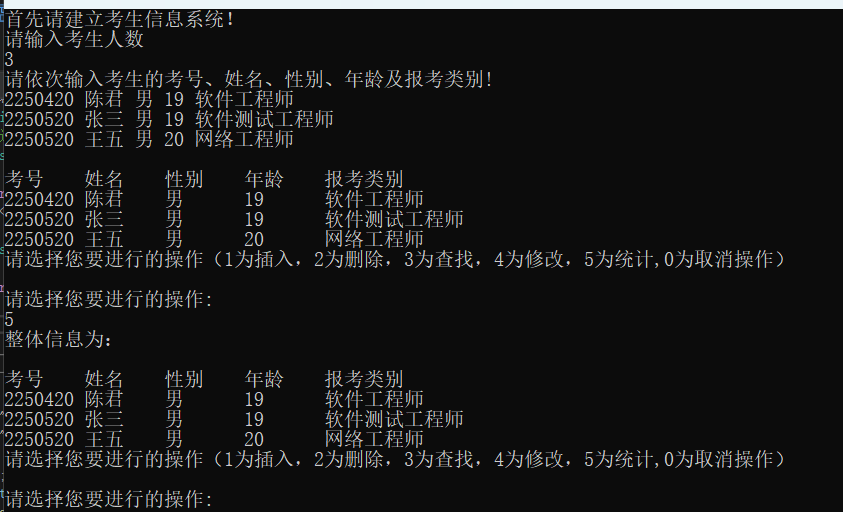


### 4.2.2 删除后链表为空

**测试用例：**删除前链表只有一个结点，删除后链表为空

**预期结果：**程序正常运行，不崩溃。

**实验结果：**



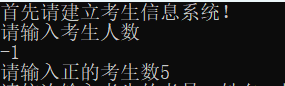
## 4.3 出错测试

### 4.3.1 考生人数错误

**测试用例：**输入考生人数为负数

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**

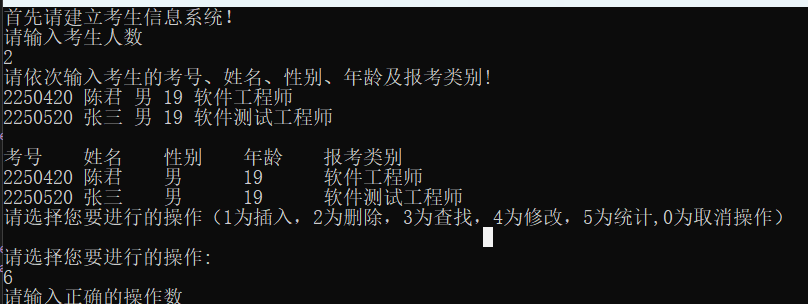
****

### 4.3.2 操作码错误

**测试用例：**输入操作码错误

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**

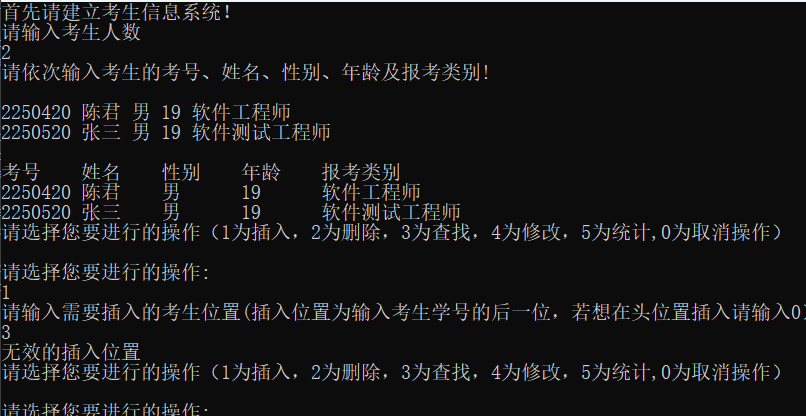
****

### 4.3.3 插入位置不存在

**测试用例：**链表里有两条记录，向链表的第四个位置插入结点

**预期结果：**程序给出错误信息，程序正常运行不崩溃。

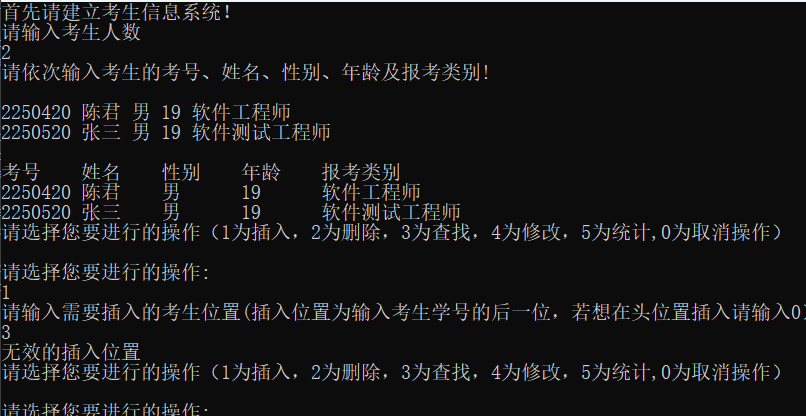
**实验结果：**

****

### 4.3.4 删除考号不存在

**测试用例：**要删除的考号不存在

**预期结果：**程序给出错误信息，程序正常运行不崩溃。

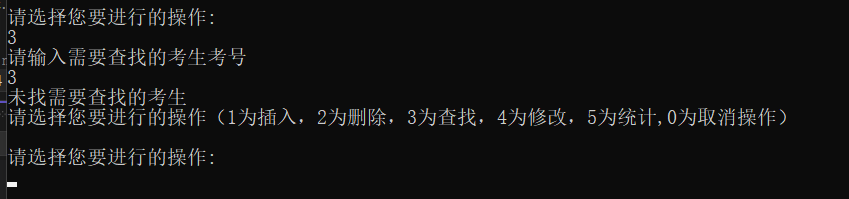
**实验结果：**

### 4.3.5 查找考号不存在

**测试用例：**要查找的考号不存在

**预期结果：**程序给出错误信息，程序正常运行不崩溃。

**实验结果：**

****

### 4.3.6 修改考号不存在

**测试用例：**要修改的考号不存在

**预期结果：**程序给出错误信息，程序正常运行不崩溃。

**实验结果：**

