项目说明文档

数据结构课程设计

——考试报名系统

作 者 姓 名： 邓泉

学 号： 1953871

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1 分析](#_Toc495668153)

[1.1 背景分析](#_Toc495668154)

[1.2 功能分析](#_Toc495668155)

[2 设计](#_Toc495668156)

[2.1 数据结构设计](#_Toc495668157)

[2.2 类结构设计](#_Toc495668158)

[2.3 成员与操作设计](#_Toc495668159)

[2.4 主程序设计](#_Toc495668160)

[3 实现](#_Toc495668161)

3.1 创建功能的实现....................................................................................................

3.1.1 创建功能流程图.........................................................................................

[3.2.2 创建功能核心代码](#_Toc495668164)

[3.2.3 创建功能截屏示例.](#_Toc495668165)....................................................................................

[3.2 插入功能的实现](#_Toc495668162)

[3.2.1 插入功能流程图](#_Toc495668163)

[3.2.2 插入功能核心代码](#_Toc495668164)

[3.2.3 插入功能截屏示例](#_Toc495668165)

[3.3 删除功能的实现](#_Toc495668166)

[3.3.1 删除功能流程图](#_Toc495668167)

[3.3.2 删除功能核心代码](#_Toc495668168)

[3.3.3 删除功能截屏示例](#_Toc495668169)

[3.4 查找功能的实现](#_Toc495668170)

[3.4.1 查找功能流程图](#_Toc495668171)

[3.4.2 查找功能核心代码](#_Toc495668172)

[3.4.3 查找功能截图示例](#_Toc495668173)

[3.5 修改功能的实现](#_Toc495668174)

[3.5.1 修改功能流程图](#_Toc495668175)

[3.5.2 修改功能核心代码](#_Toc495668176)

[3.5.3 修改功能截屏示例](#_Toc495668177)

[3.6 统计功能的实现](#_Toc495668178)

[3.6.1 统计功能流程图](#_Toc495668179)

[3.6.2 统计功能核心代码](#_Toc495668180)

[3.6.3 统计功能截屏示例](#_Toc495668181)

[3.7 总体系统的实现](#_Toc495668182)....................................................................................................

[3.7.1 总体系统流程图](#_Toc495668183)

[3.7.2 总体系统核心代码](#_Toc495668184)

[4 测试](#_Toc495668186)

[4.1 功能测试](#_Toc495668187)

[4.1.1 插入功能测试](#_Toc495668188)

[4.1.2 删除功能测试](#_Toc495668189)

[4.1.3 查找功能测试](#_Toc495668190)

[4.1.4 修改功能测试](#_Toc495668191)

[4.1.5 统计功能测试](#_Toc495668192)

[4.2 边界测试](#_Toc495668193)

[4.2.1 初始化无输入数据](#_Toc495668194)

[4.2.3 删除后链表为空](#_Toc495668196)

[4.3 出错测试](#_Toc495668197)

[4.3.1 考生人数错误](#_Toc495668198)

[4.3.2 操作码错误](#_Toc495668199)

[4.3.4 删除考号不存在](#_Toc495668201)

[4.3.5 查找考号不存在](#_Toc495668202)

[4.3.6 修改考号不存在](#_Toc495668203)

4.3.7 创建系统时输入错误.................................................................................

# 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，然而随着学生数量和考试数量的日益庞大，考试报名工作给各高校报名工作带来了新的挑战，给教务管理部门增加了很大的工作量。传统的手工登记操作具有费时费力，准确性差，难以处理大量数据的特点，在各类考试频繁举办，考生人数日益增加的当今社会，已经不能满足各高校机构的需求。因此，利用计算机科学技术，建立一套稳定、高效且易于操作的考试报名系统，具有重要的现实意义。

## 1.2 功能分析

考试报名系统的主要任务是记录考生的信息，因此首先应当具备的功能是考生信息的输入、存储和在需要时对考生信息的提取，查询。此外，为了应对实际使用中会遇到的各种常见情况，例如考生信息的变更、考生弃考等，考试报名系统还应该具有修改、删除功能，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。

# 2 设计

## 2.1 数据结构设计

在实际使用中，被存储的考生信息应具有合适的数据结构。作为常用的顺序存储结构，数组和链表是我们的主要考察对象。由于该考试报名系统会频繁收到插入考生信息，删除考生信息的请求，且对于删除操作并不保证只删除容器中末尾的元素，如使用数组对数据进行存储，则程序效率将十分低下。此外，鉴于在考试报名截止前难以准确估计报考人数，使用数组常会面临预留存储空间太大或太小的问题。而如果使用链表存储考生数据，以上提到的问题都能得到很好的解决。首先，链表并不要求数据存储在物理上连续，这意味着不需要预先估计存储数据所需的空间大小。此外，无论是对链表进行插入还是删除操作，也无论是对头尾节点还是中间节点，操作的时间复杂度都为O（1）。

因此，选择采用带有附加头结点的单链表。

## 2.2 类结构设计

本系统使用的链表由链表结点类（LinkNode）和链表类（List）组成。为方便处理，采用struct描述链表结点类（LinkNode），这样使得链表类（List）可以直接访问链表结点。

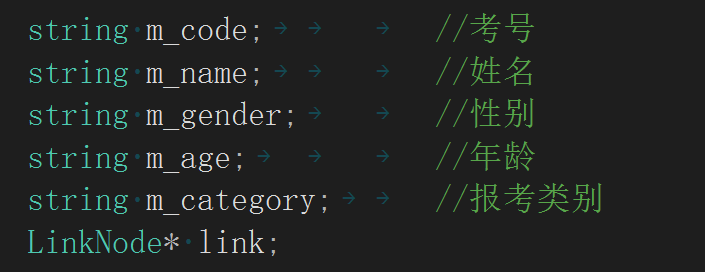
链表节点含有六个成员，分别是该结点所包含的元素值（即学生信息，包含考号、姓名、性别、年龄和报考类别）和指向下一结点的指针。

链表类仅包含三个私有成员，分别是指向头结点的头指针、指向尾结点的尾指针、链表长度。

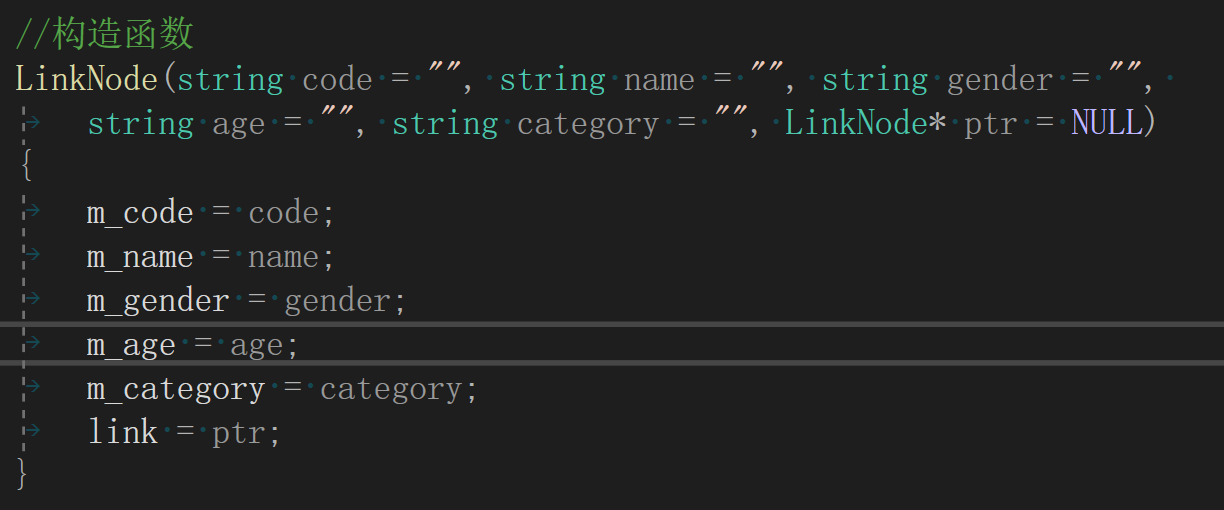
## 2.3 成员与操作设计

**链表结点类（LinkNode）**

**公有成员：**

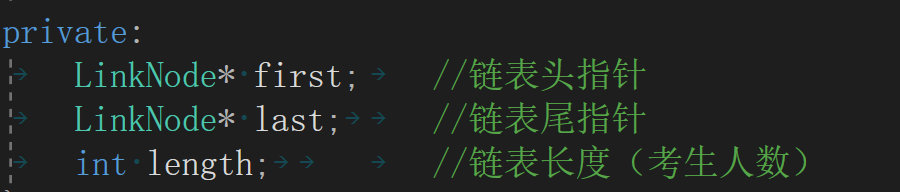
****

**公有操作：**

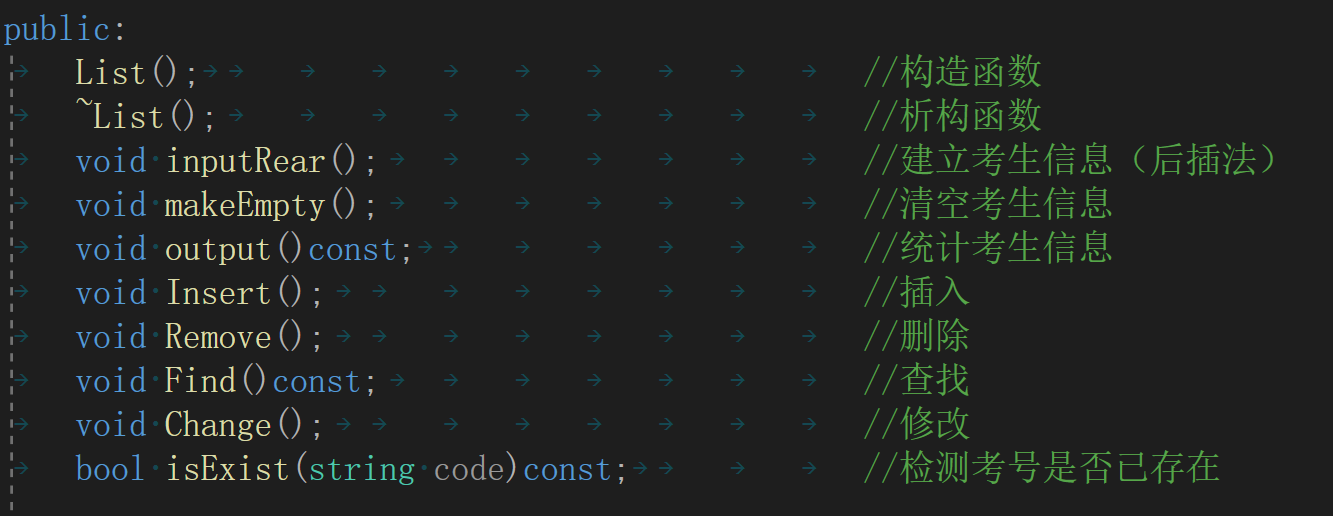
****

**链表类（List）**

**私有成员：**

****

**公有操作：**

****

## 

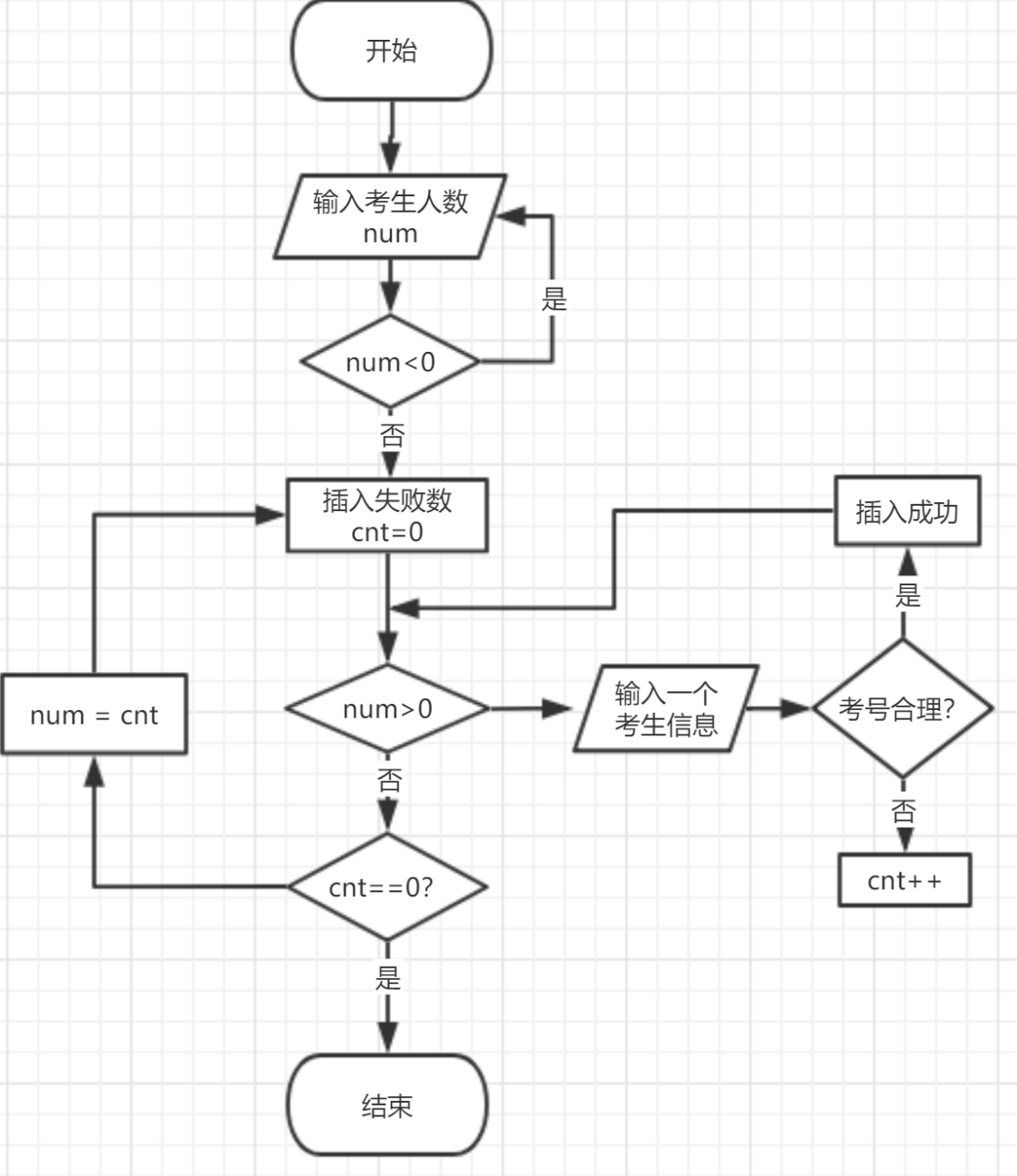
## 2.4 主程序设计

系统首先建立List的对象L,输出窗口将引导用户逐步建立学生信息表,然后展示所建立的信息系统。随后进入循环，根据用户输入，调用L对象相应的函数，根据用户所输入的操作码（choice）执行链表L对应的成员函数, 完成插入，删除，查找，修改，统计功能，直至用户选择操作0退出系统。

# 3 实现

3.1 创建功能的实现

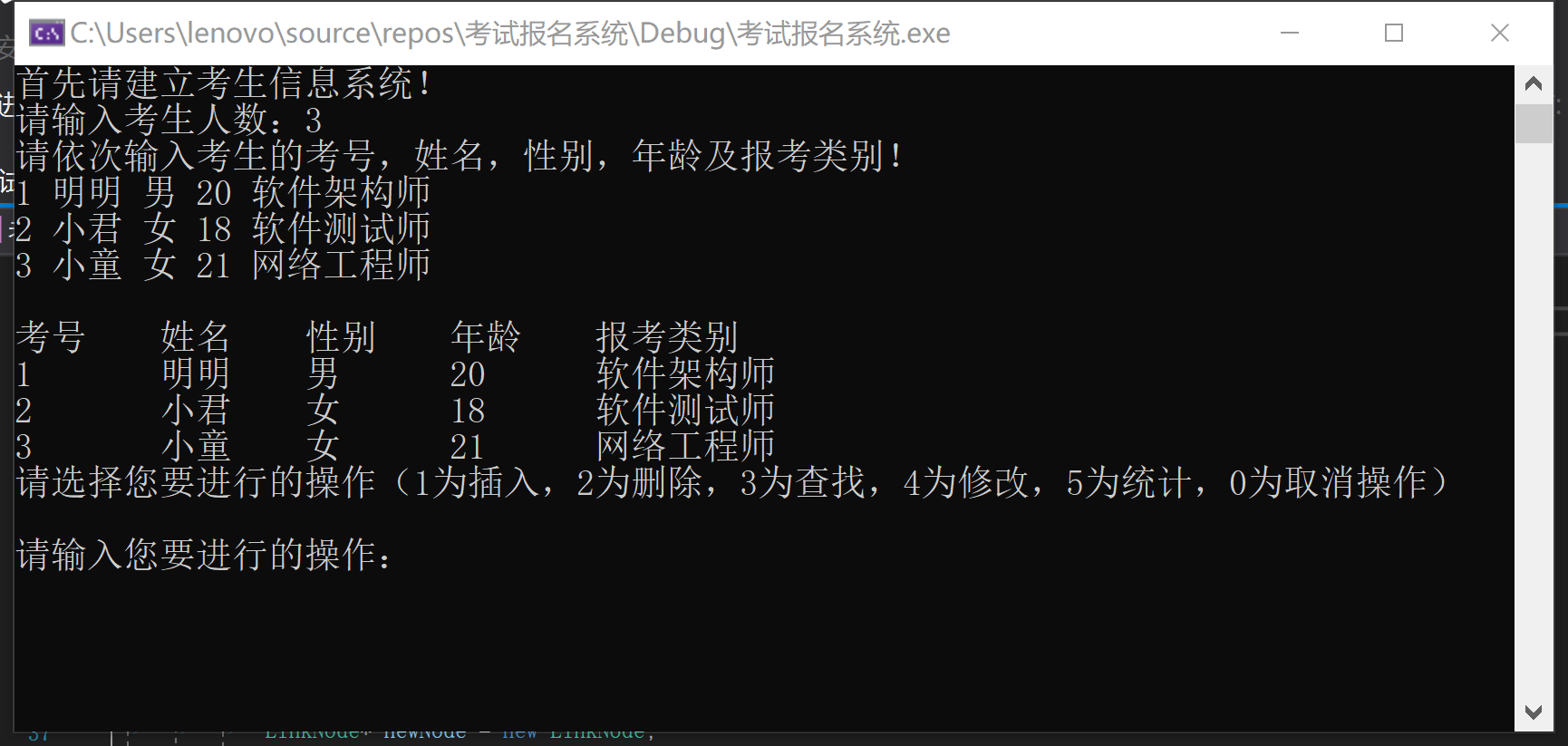
3.1.1 创建功能流程图



3.1.2 创建功能核心代码

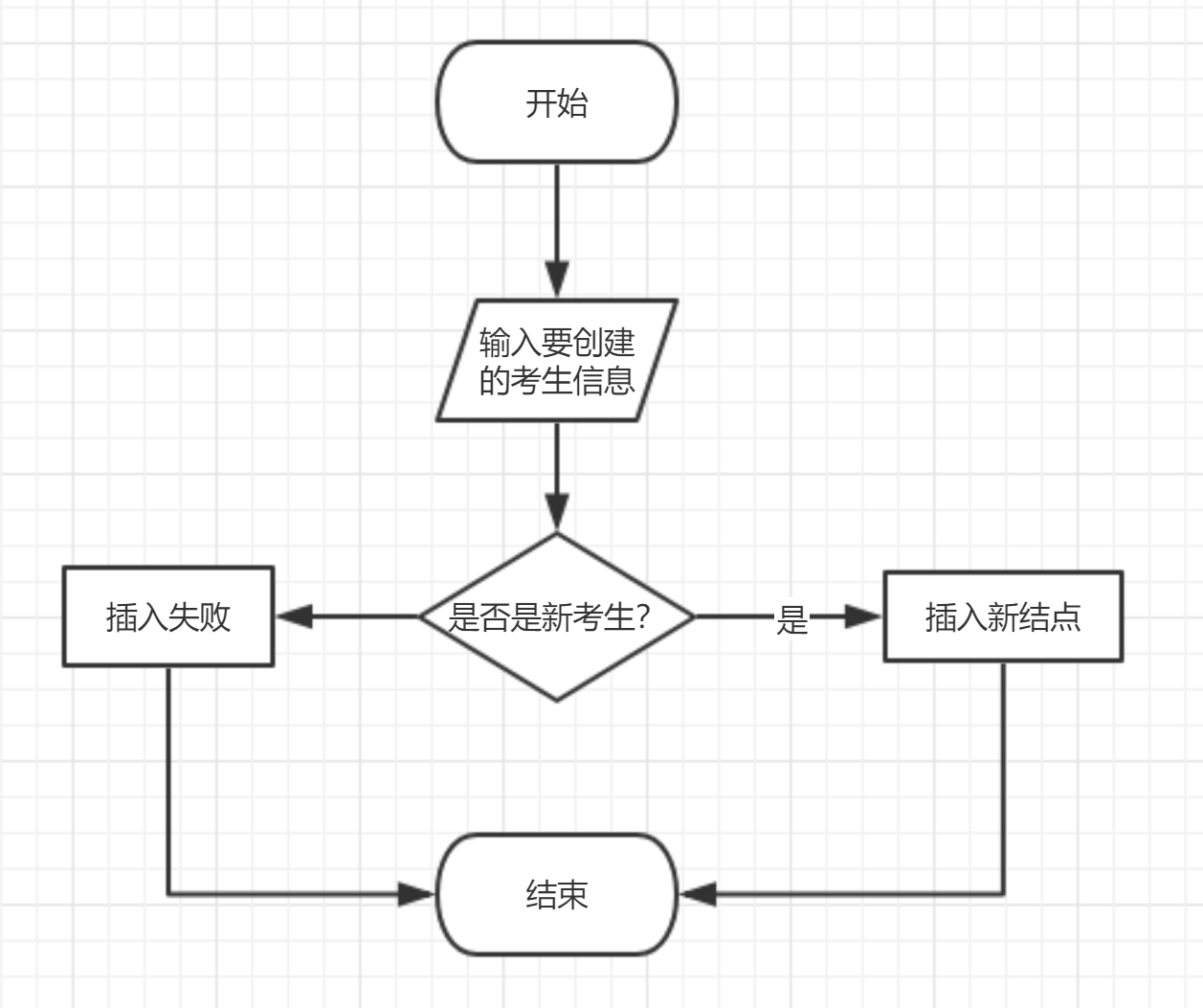


3.1.3 创建功能截屏示例

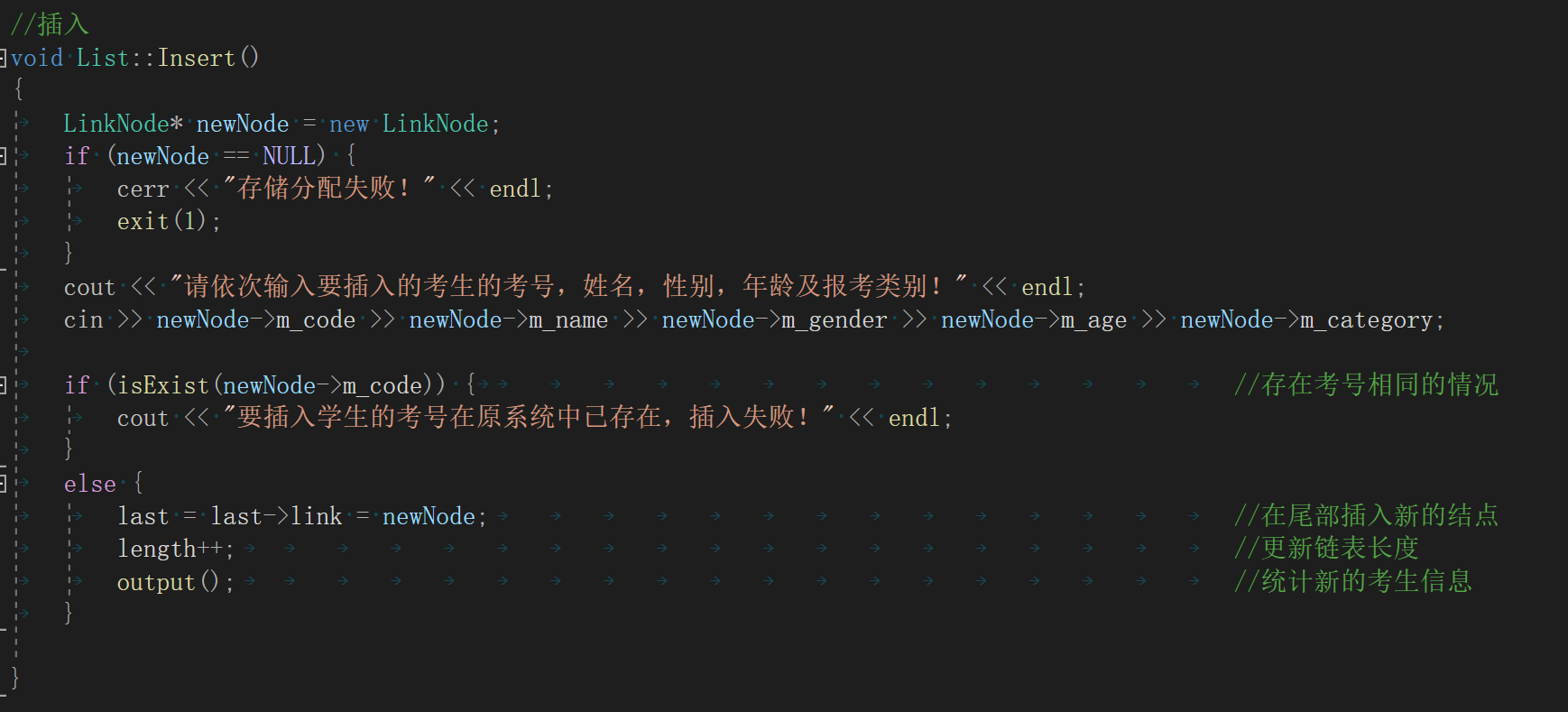


## 3.2 插入功能的实现

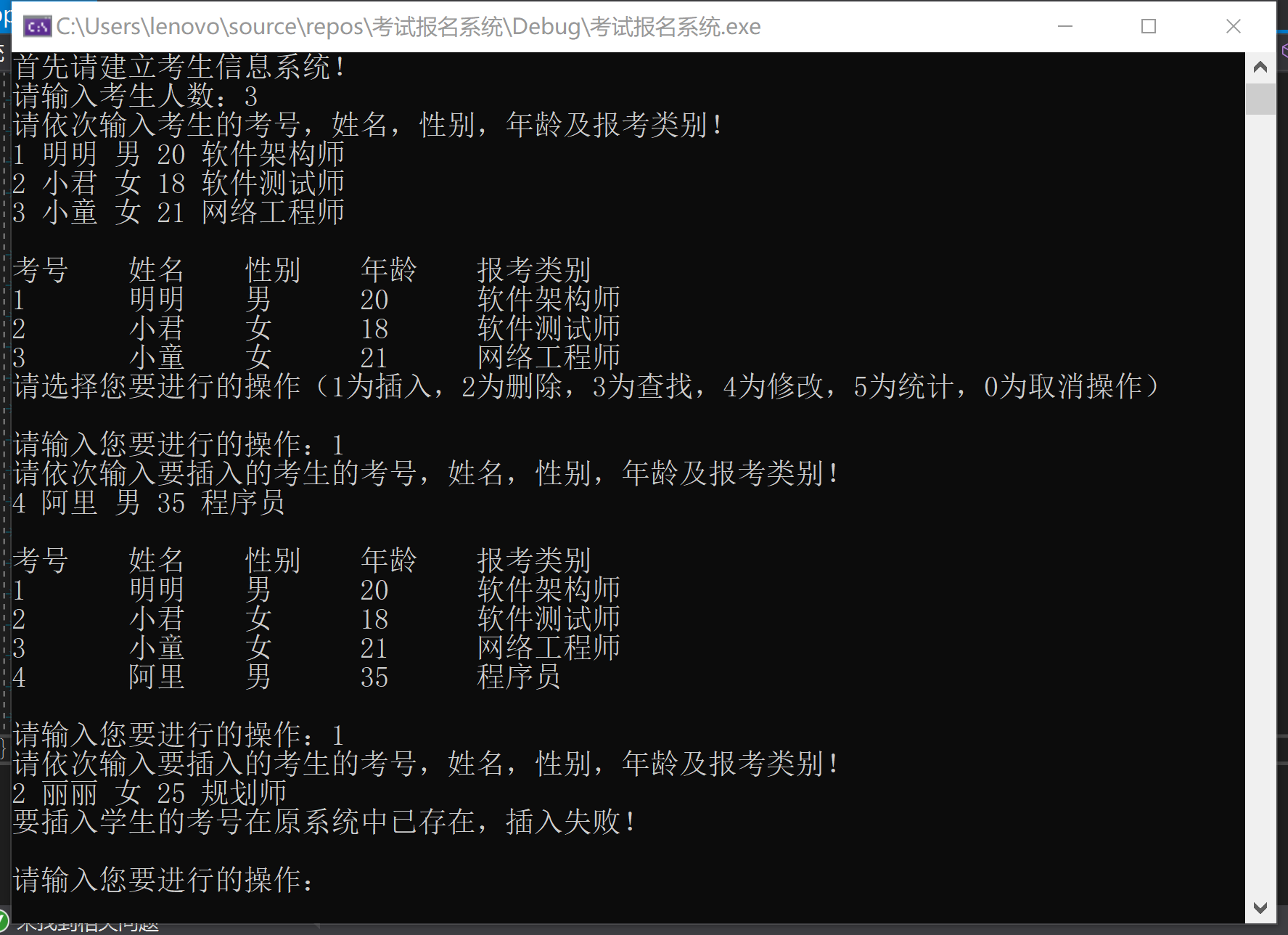
### 3.2.1 插入功能流程图



### 3.2.2 插入功能核心代码

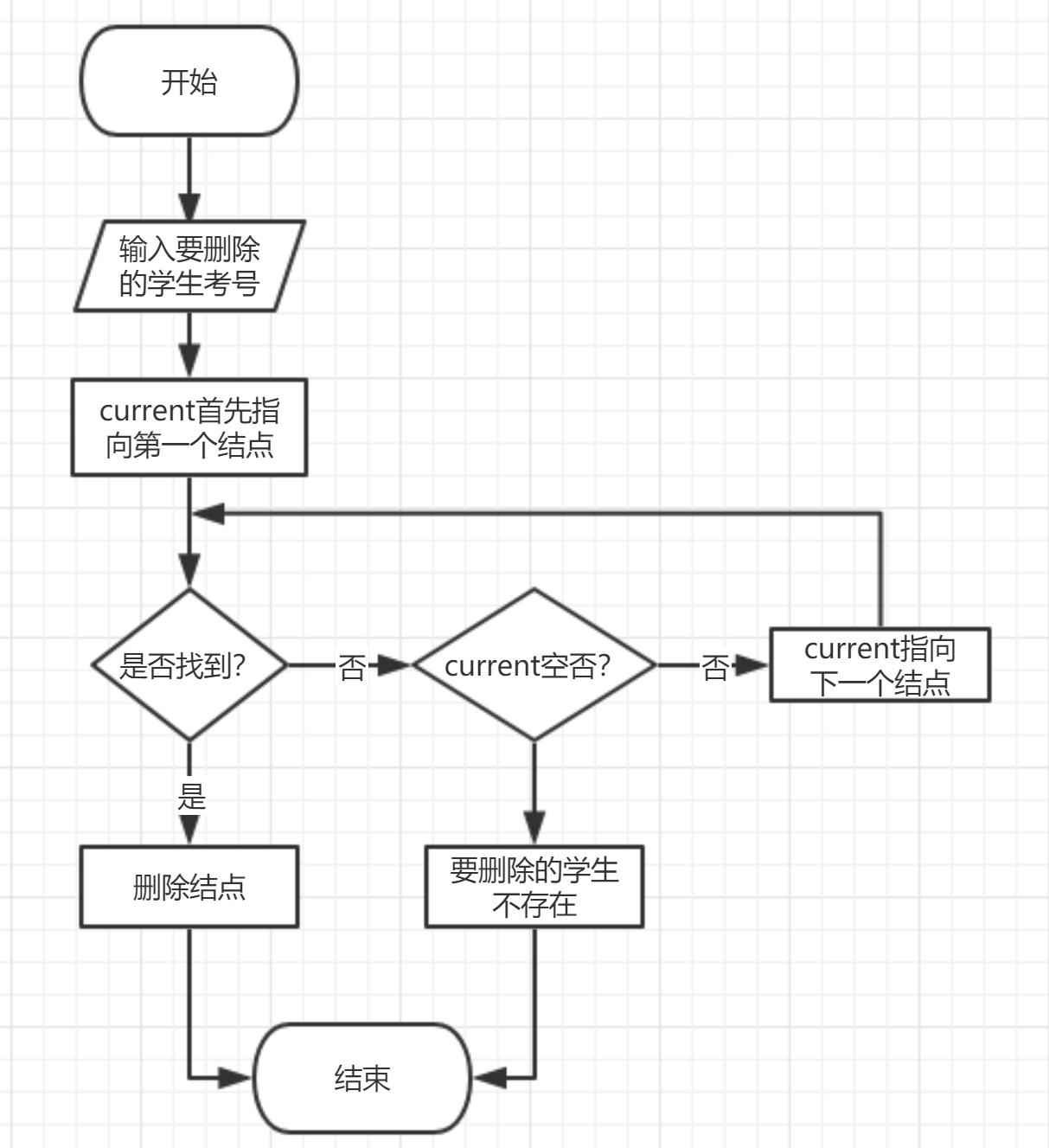


### 3.2.3 插入功能截屏示例

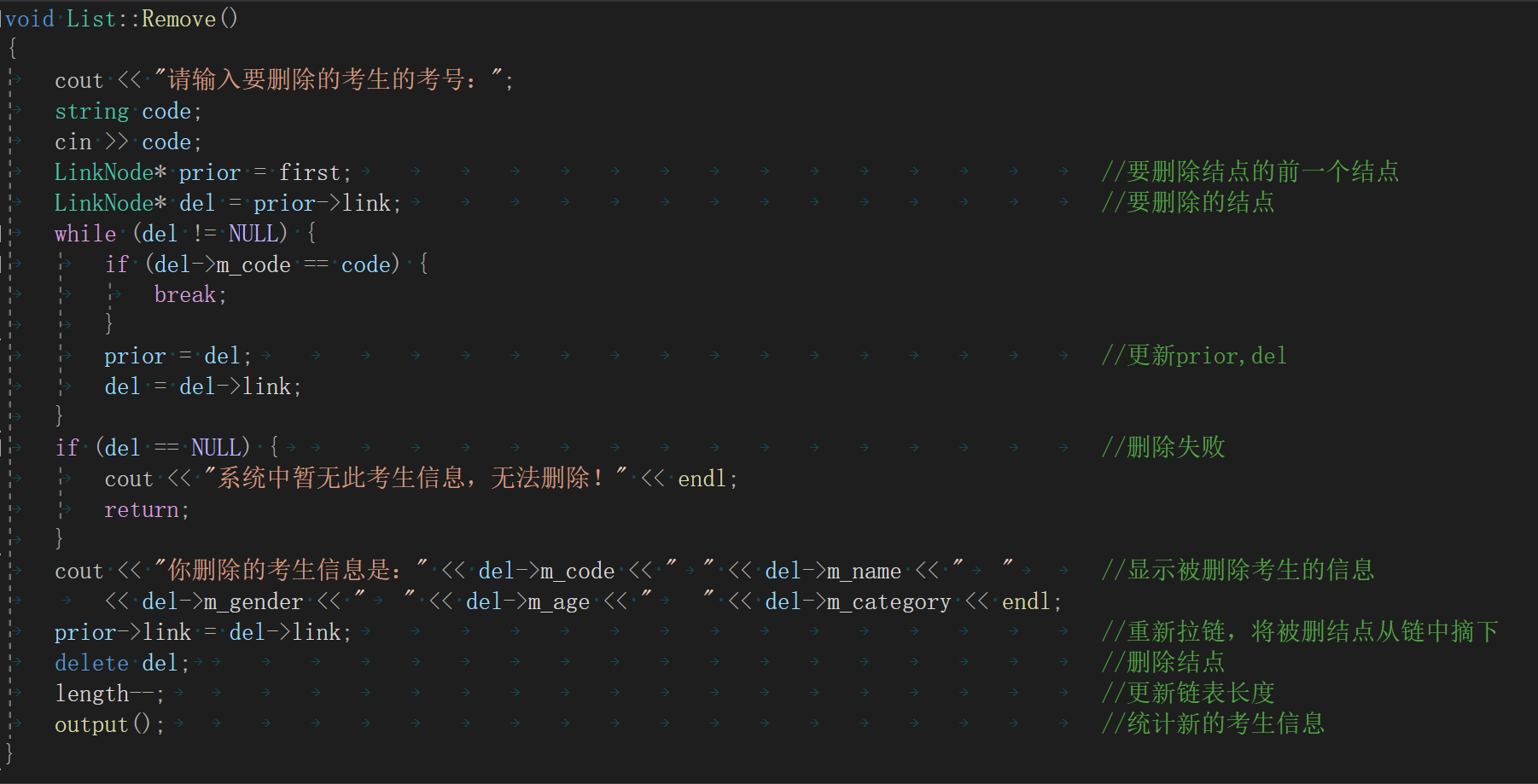


## 3.3 删除功能的实现

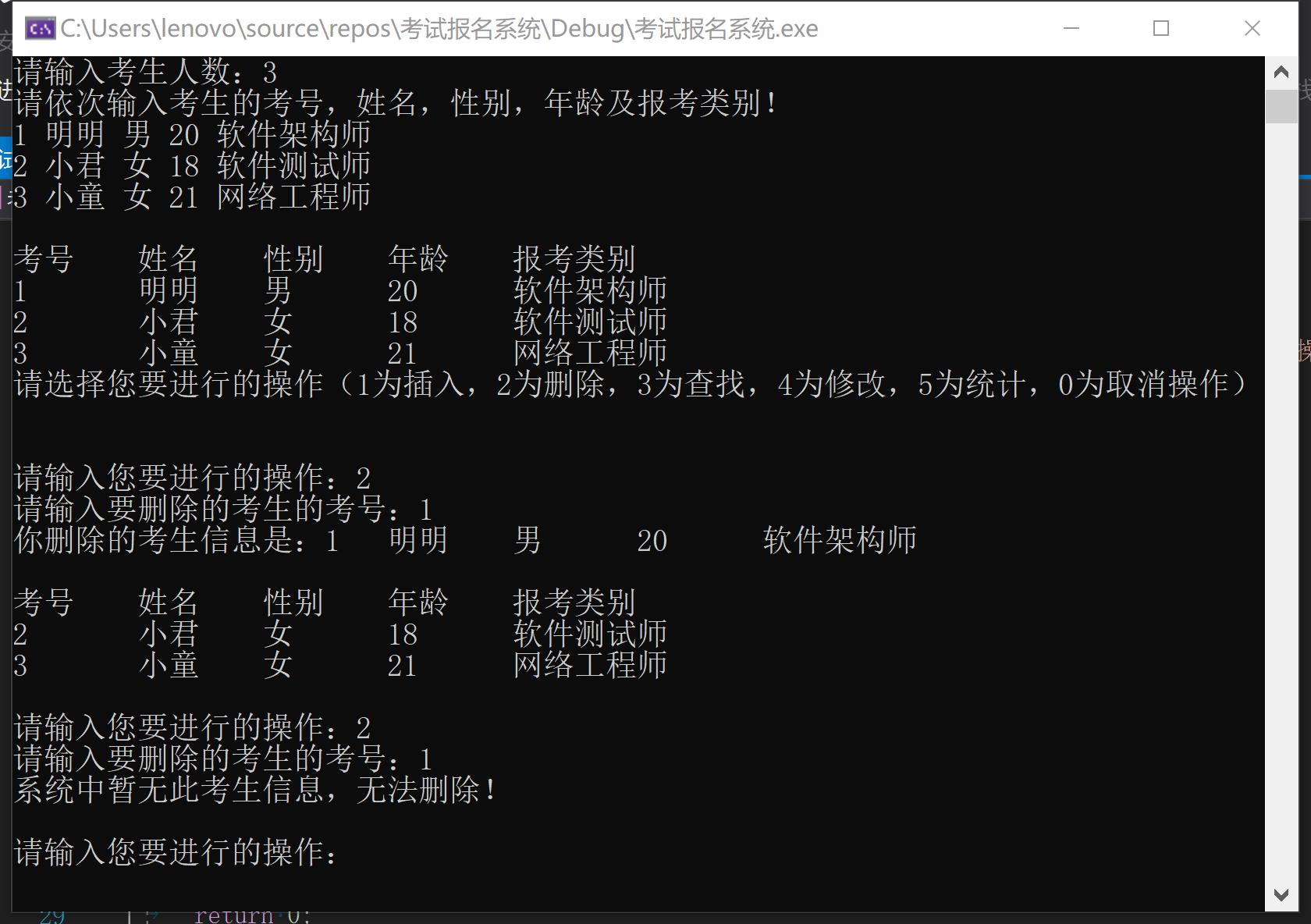
### 3.3.1 删除功能流程图



### 3.3.2 删除功能核心代码

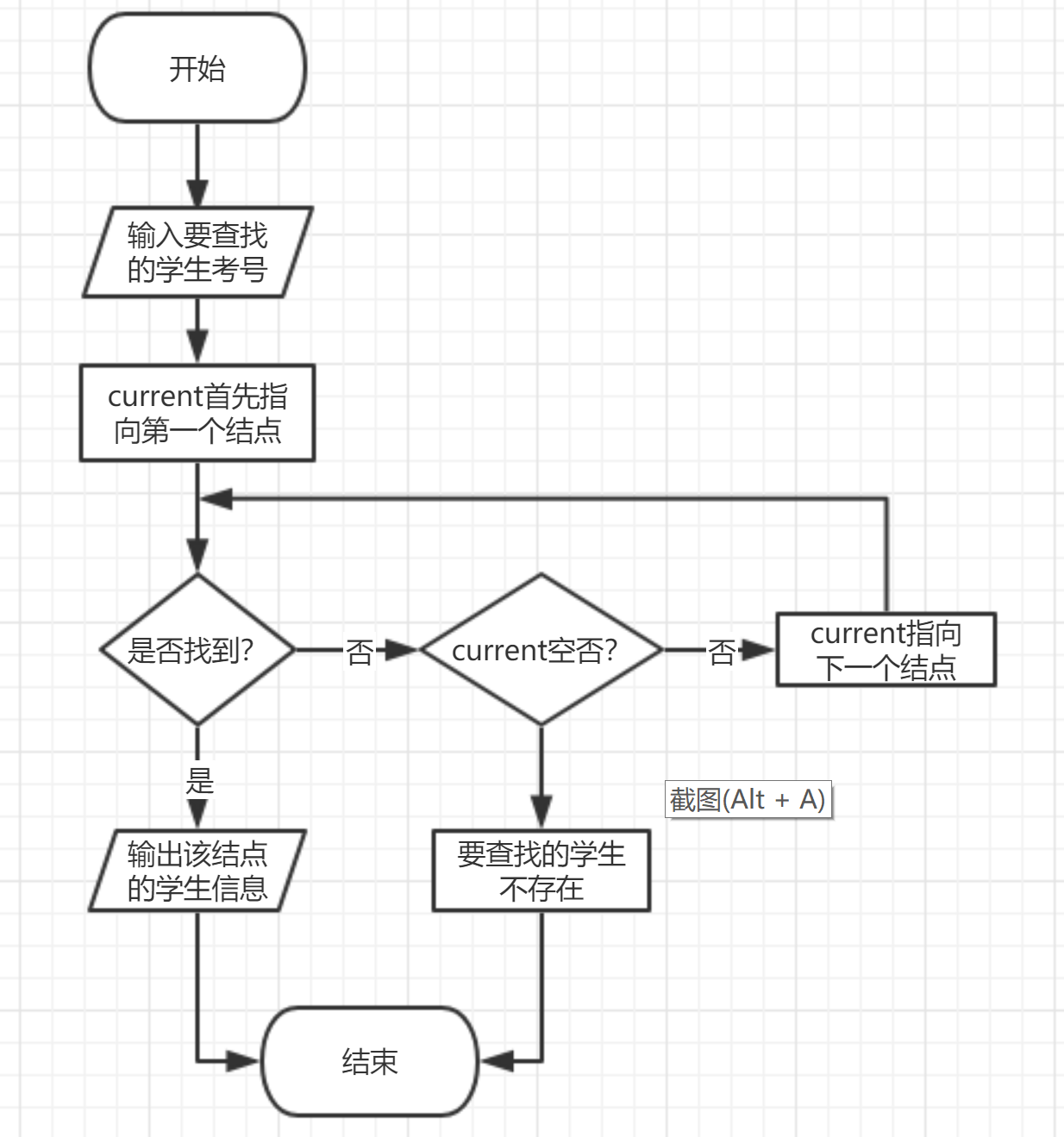


### 3.3.3 删除功能截屏示例

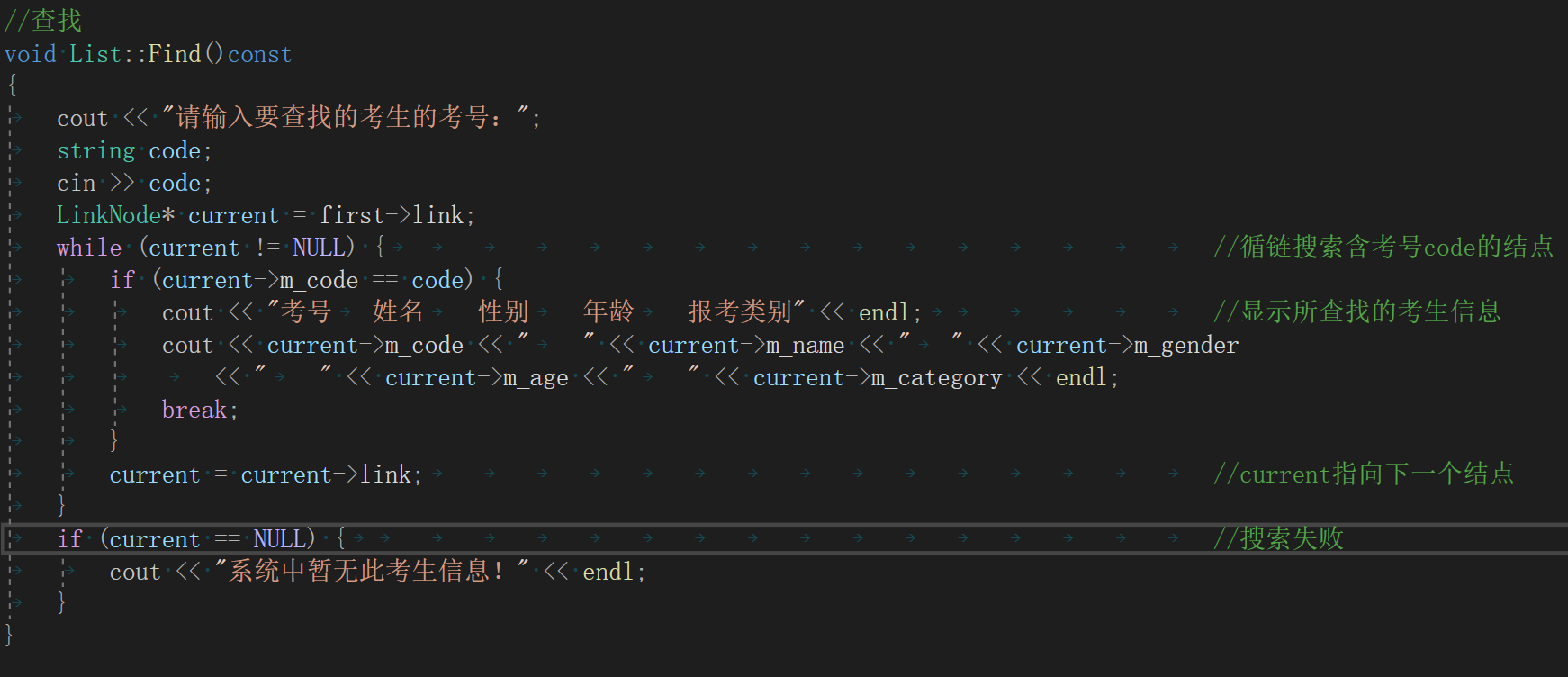


## 3.4 查找功能的实现

### 3.4.1 查找功能流程图

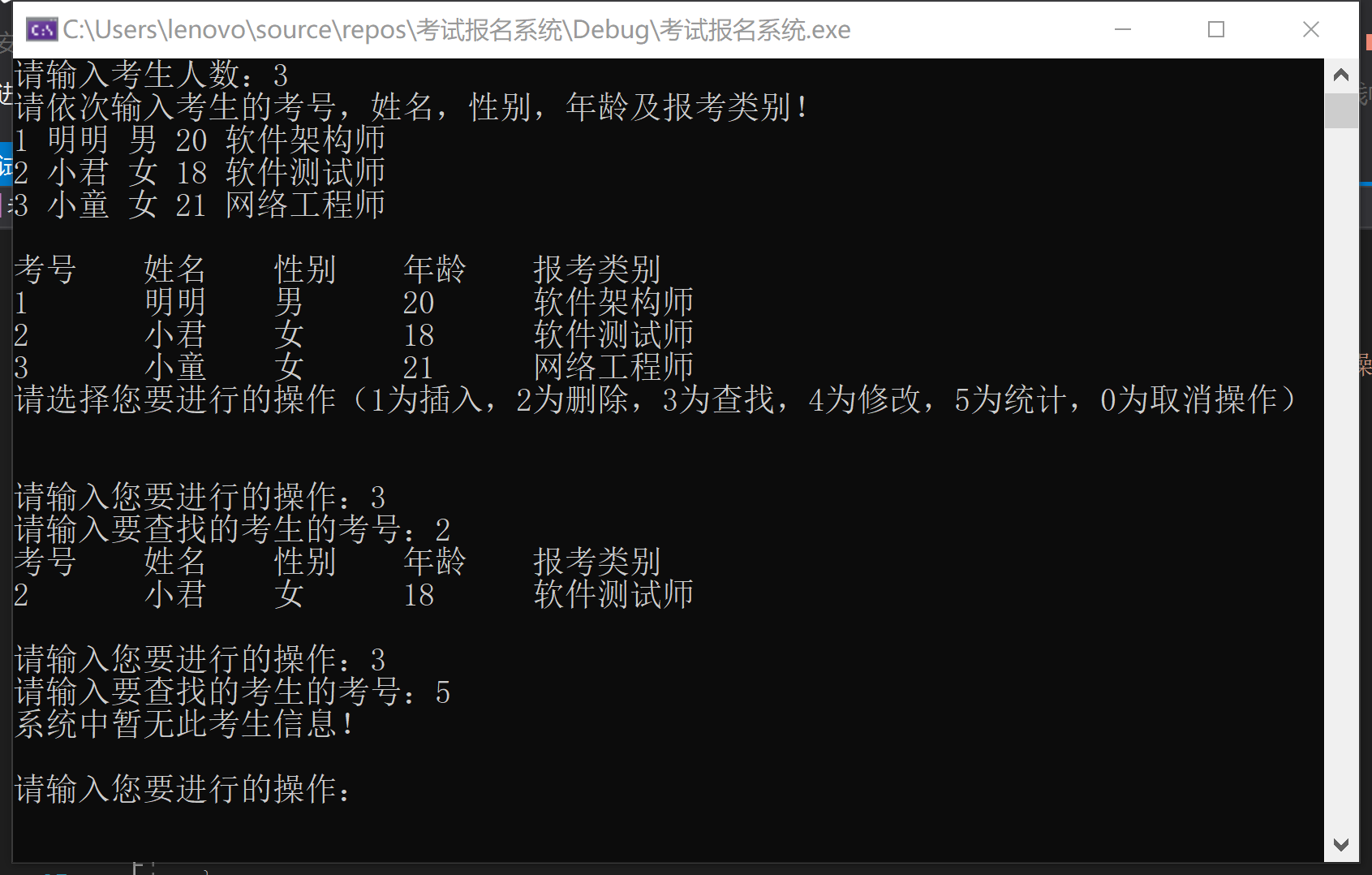


### 3.4.2 查找功能核心代码



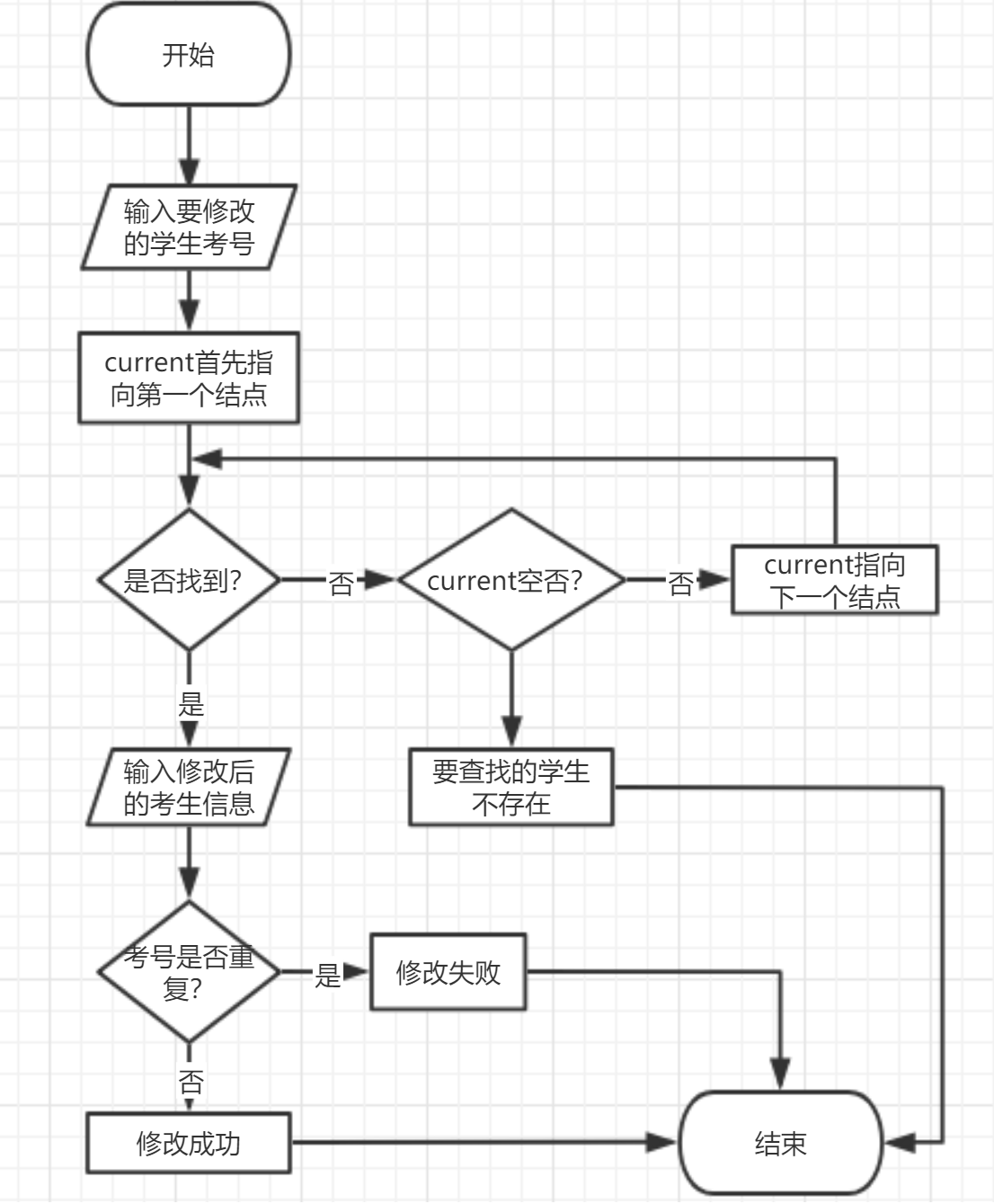
### 

### 3.4.3 查找功能截图示例

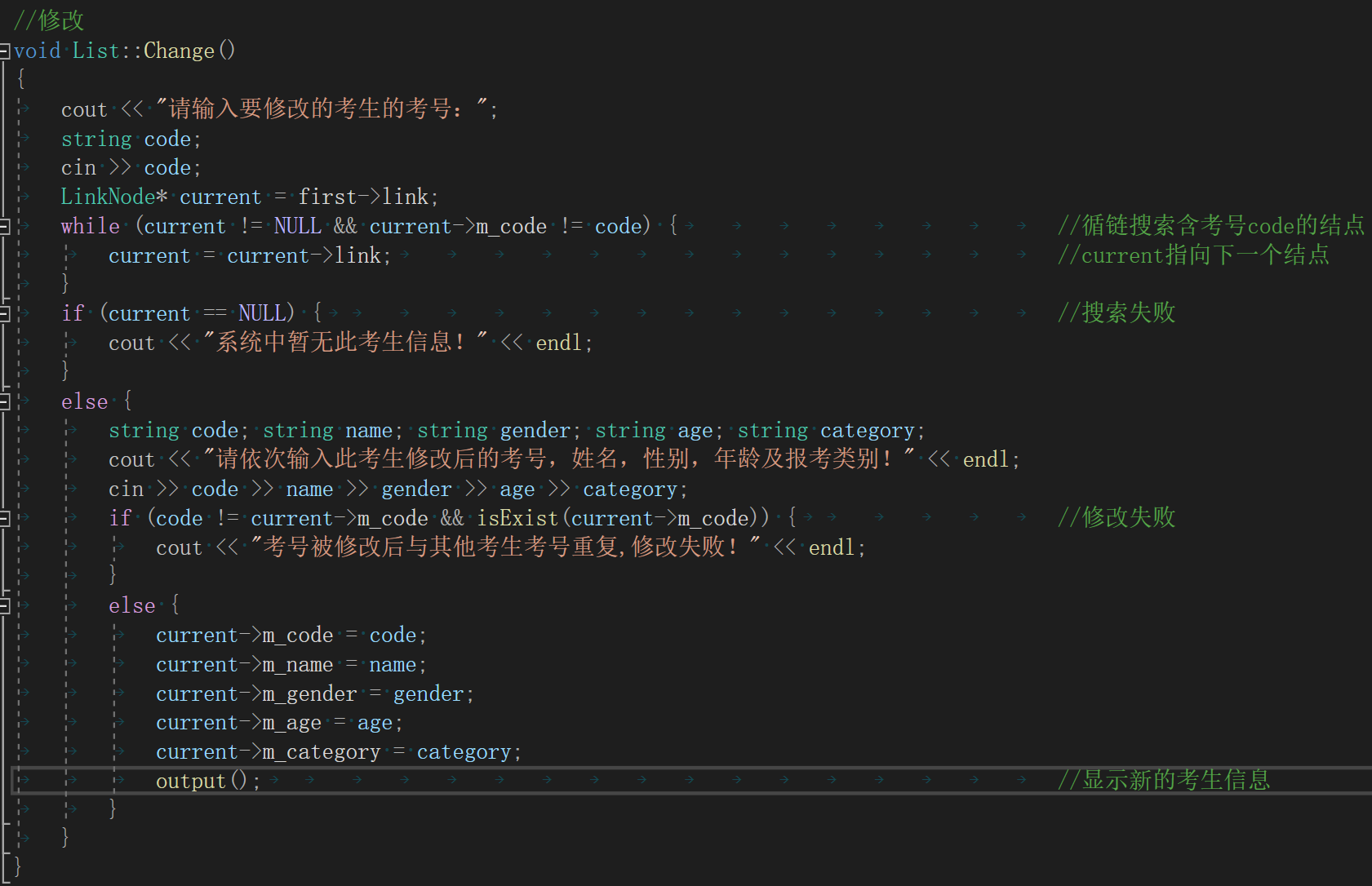


## 3.5 修改功能的实现

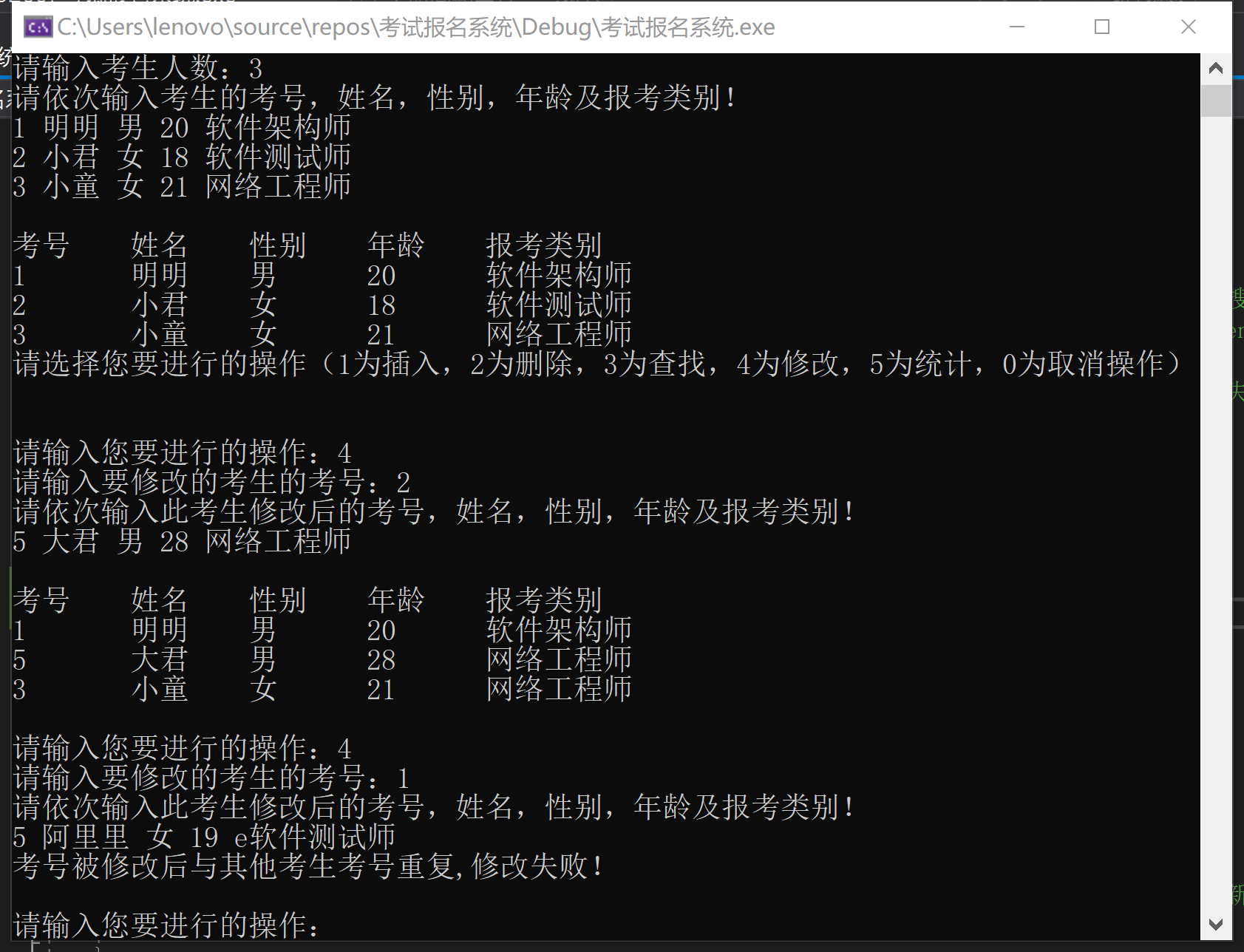
### 3.5.1 修改功能流程图



### 3.5.2 修改功能核心代码

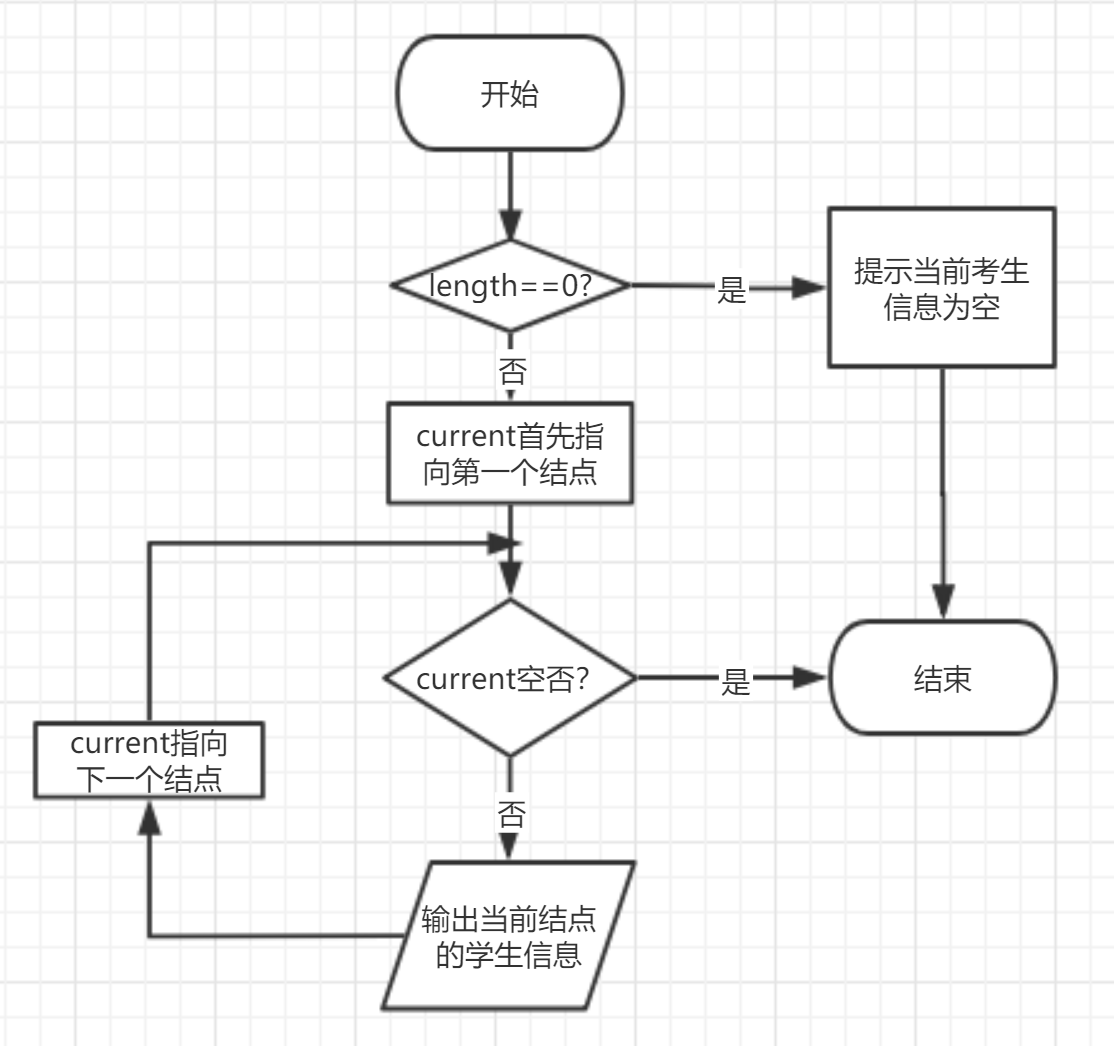


### 3.5.3 修改功能截屏示例



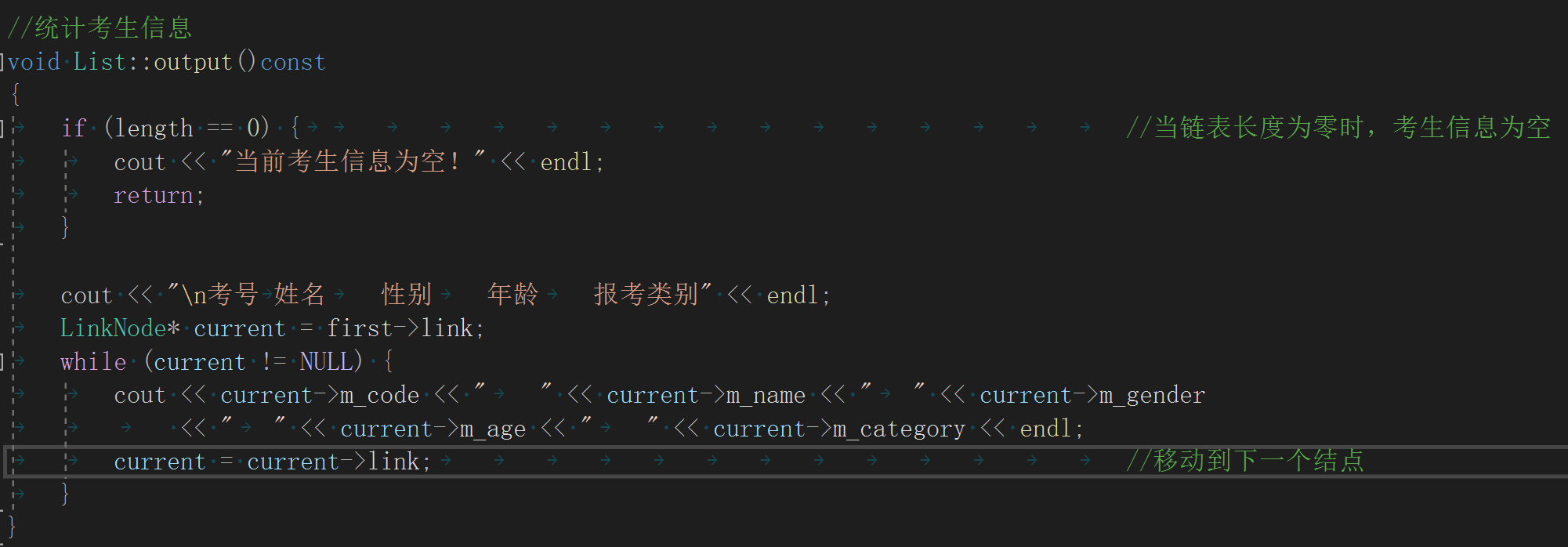
## 3.6 统计功能的实现

### 3.6.1 统计功能流程图



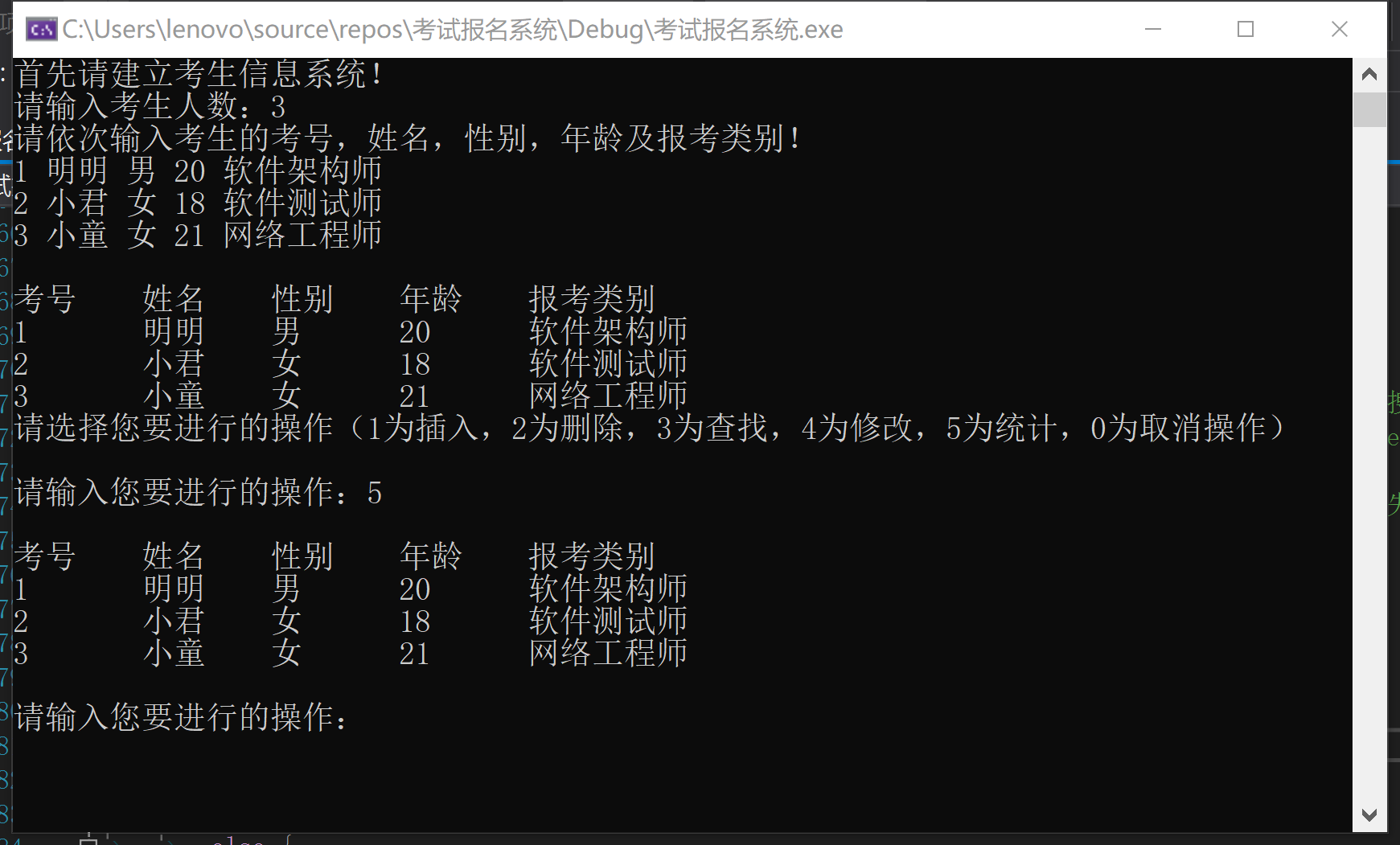
### 

### 3.6.2 统计功能核心代码



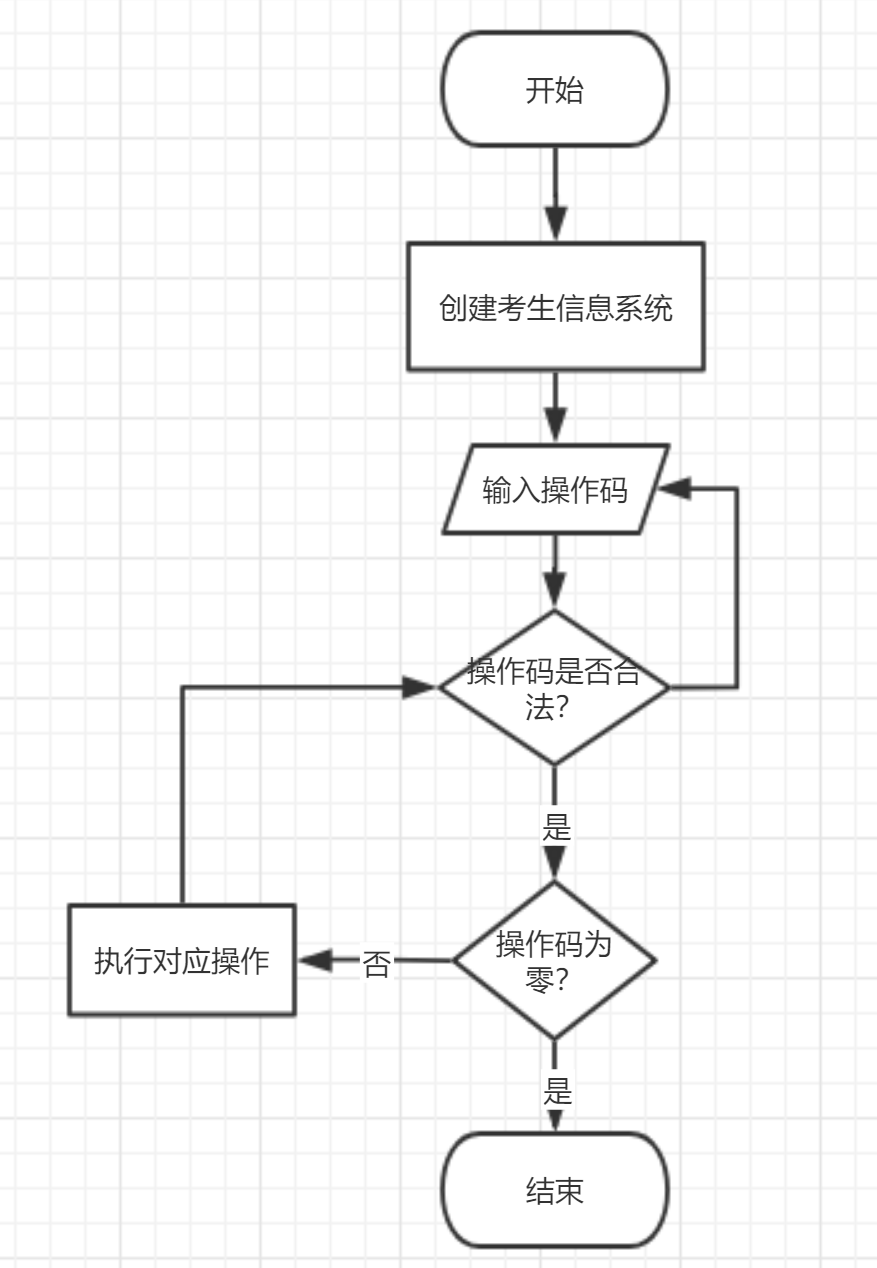
### 

### 3.6.3 统计功能截屏示例

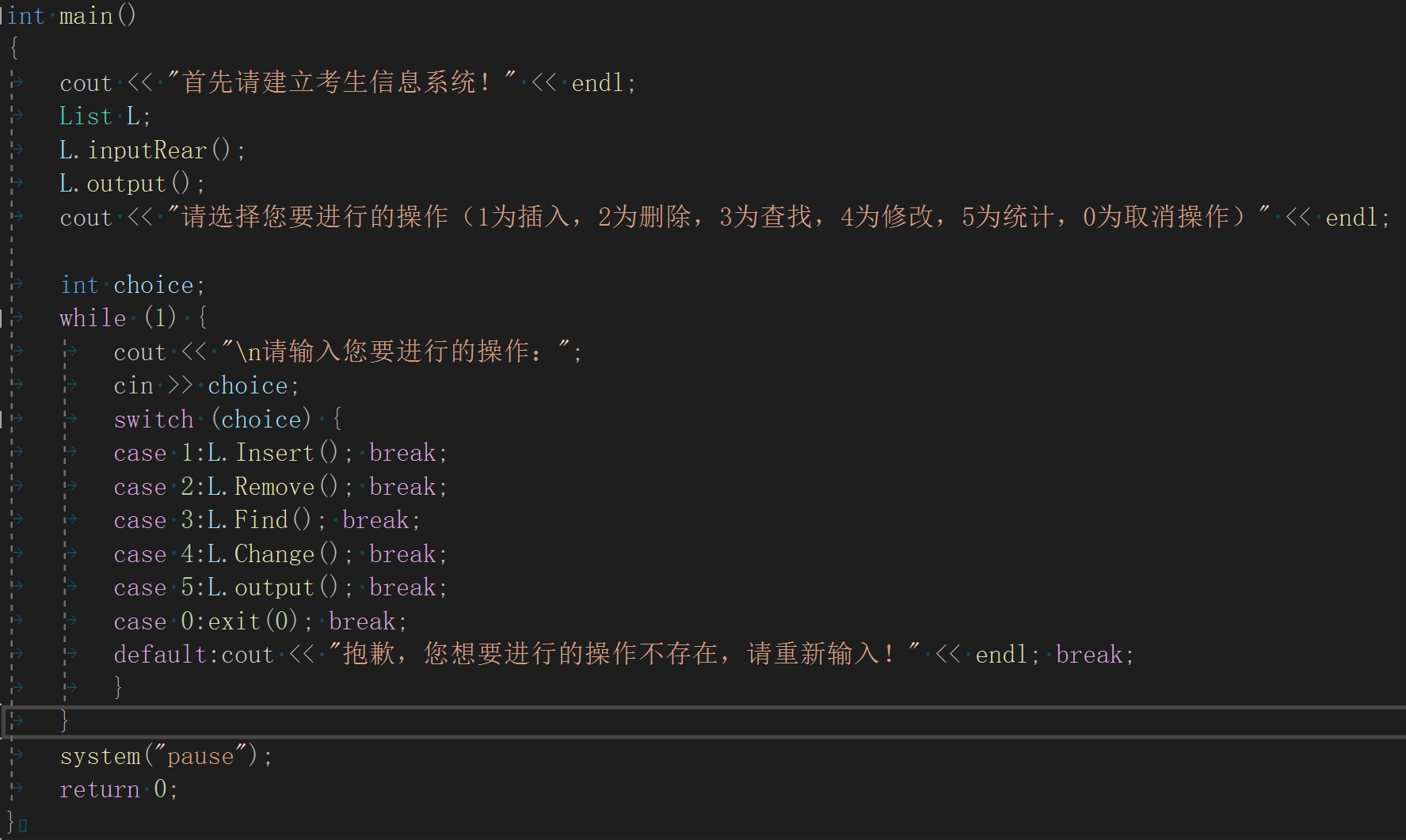


## 3.7 总体系统的实现

### 3.7.1 总体系统流程图



### 3.7.2 总体系统核心代码



# 

# 4 测试

## 4.1 功能测试

### 4.1.1 插入功能测试

**测试用例**：4 stu4 女 22 网络工程师

**预期结果**：

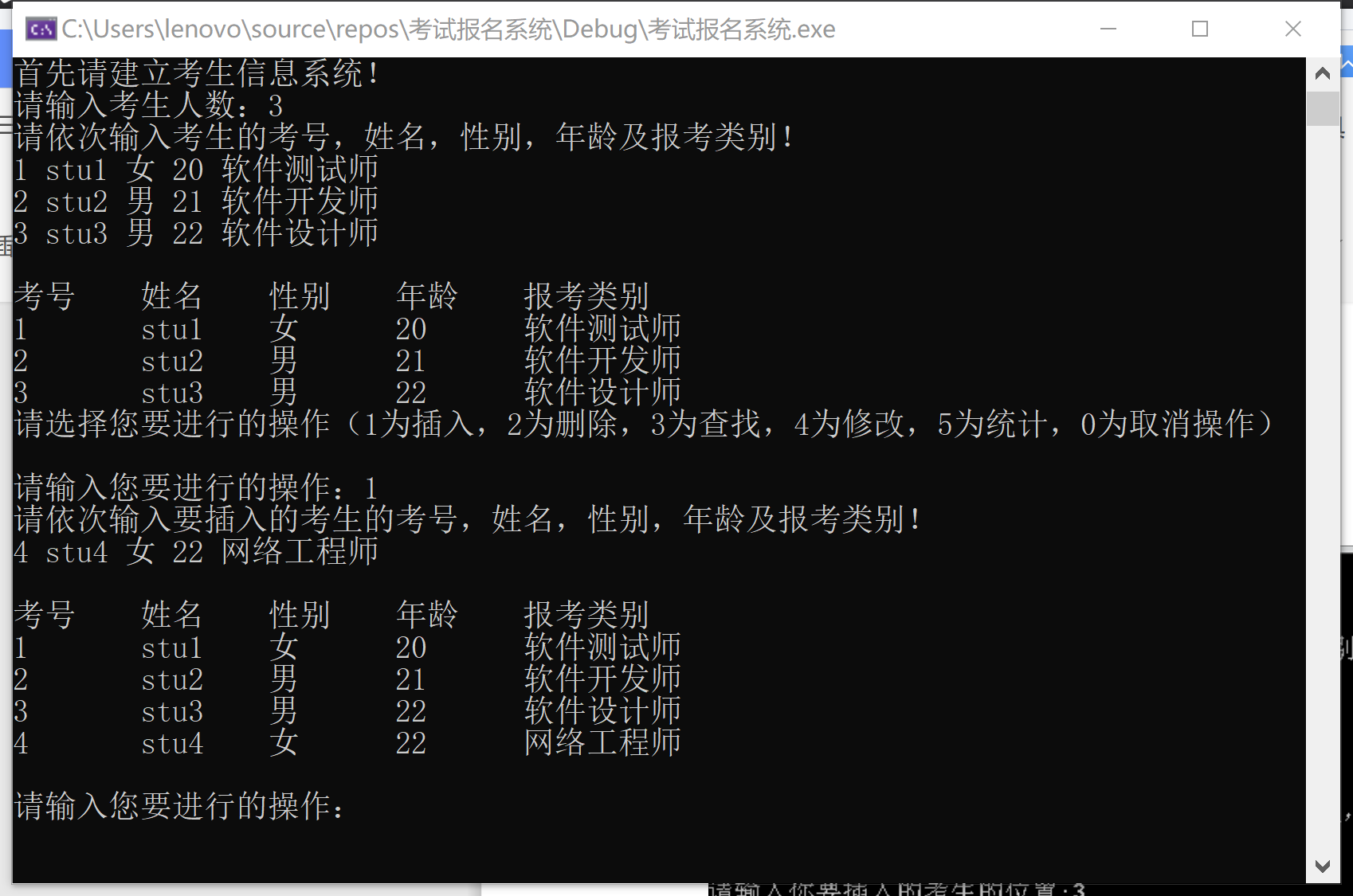
1 stu1 女 20 软件测试师

2 stu2 男 21 软件开发师

3 stu3 男 22 软件设计师

4 stu4 女 22 网络工程师

**实验结果**



### 4.1.2 删除功能测试

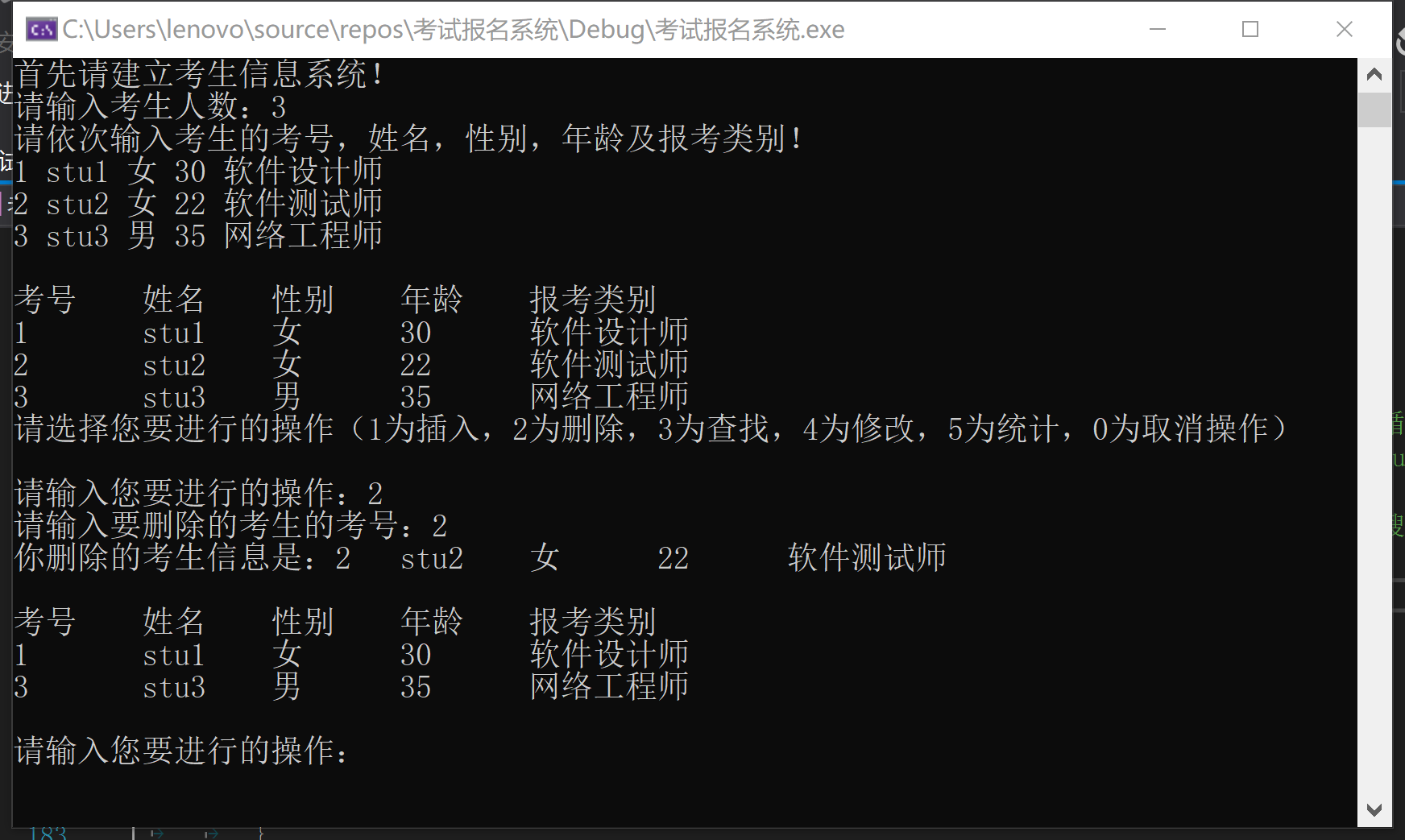
**测试用例：**删除考号为2的考生

**预期结果：**

1 stu1 女 30 软件设计师

3 stu3 男 35 网络工程师

**实验结果：**



### 

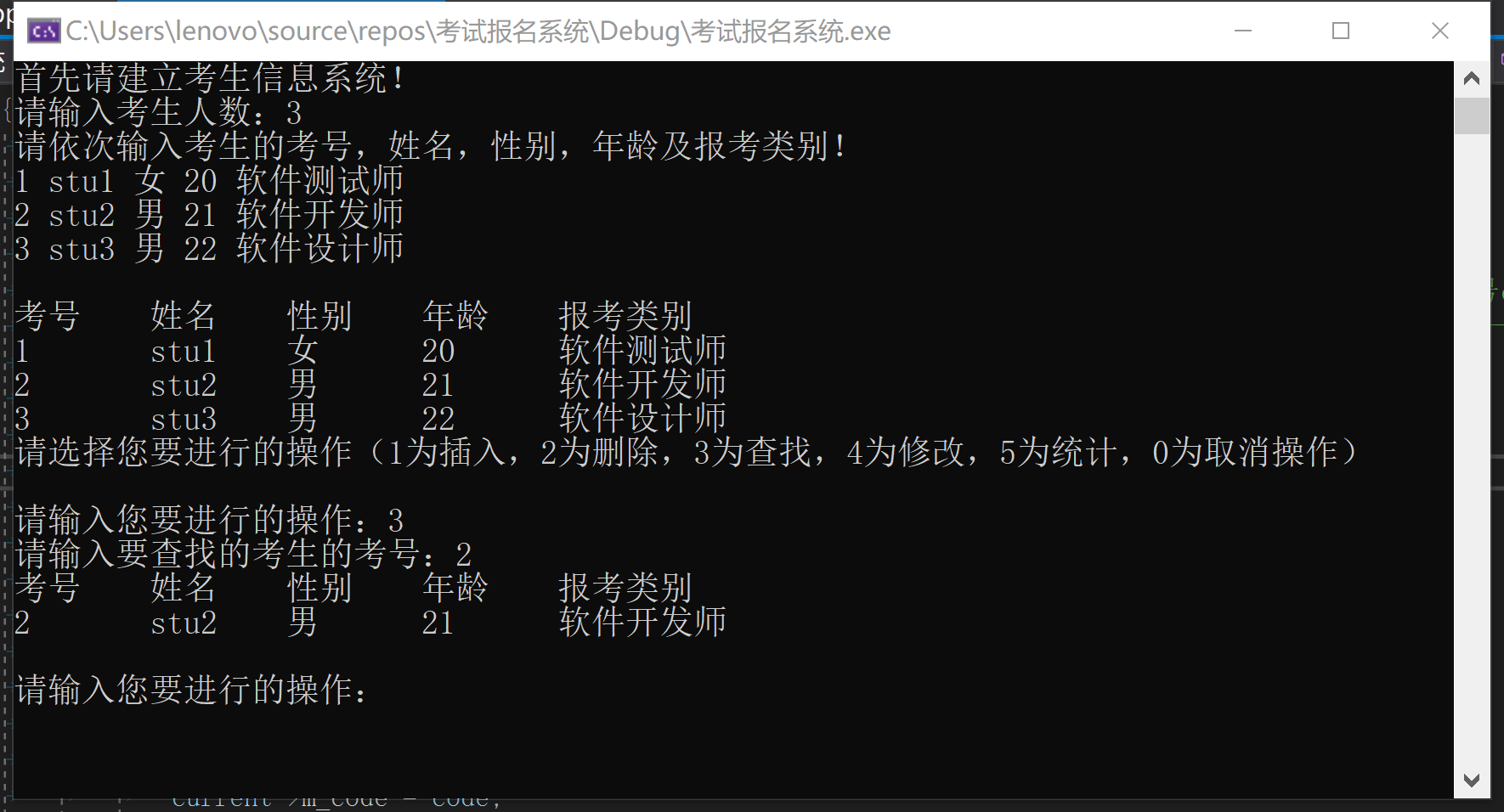
### 4.1.3 查找功能测试

**测试用例：**查找考号为2的考生

**预期结果：**

2 stu2 男 21 软件开发师

**实验结果：**



### 

### 4.1.4 修改功能测试

**测试用例：**将考号为2的考生修改为：考号5，姓名stu5，性别女，年龄26，报考类别网络工程师。

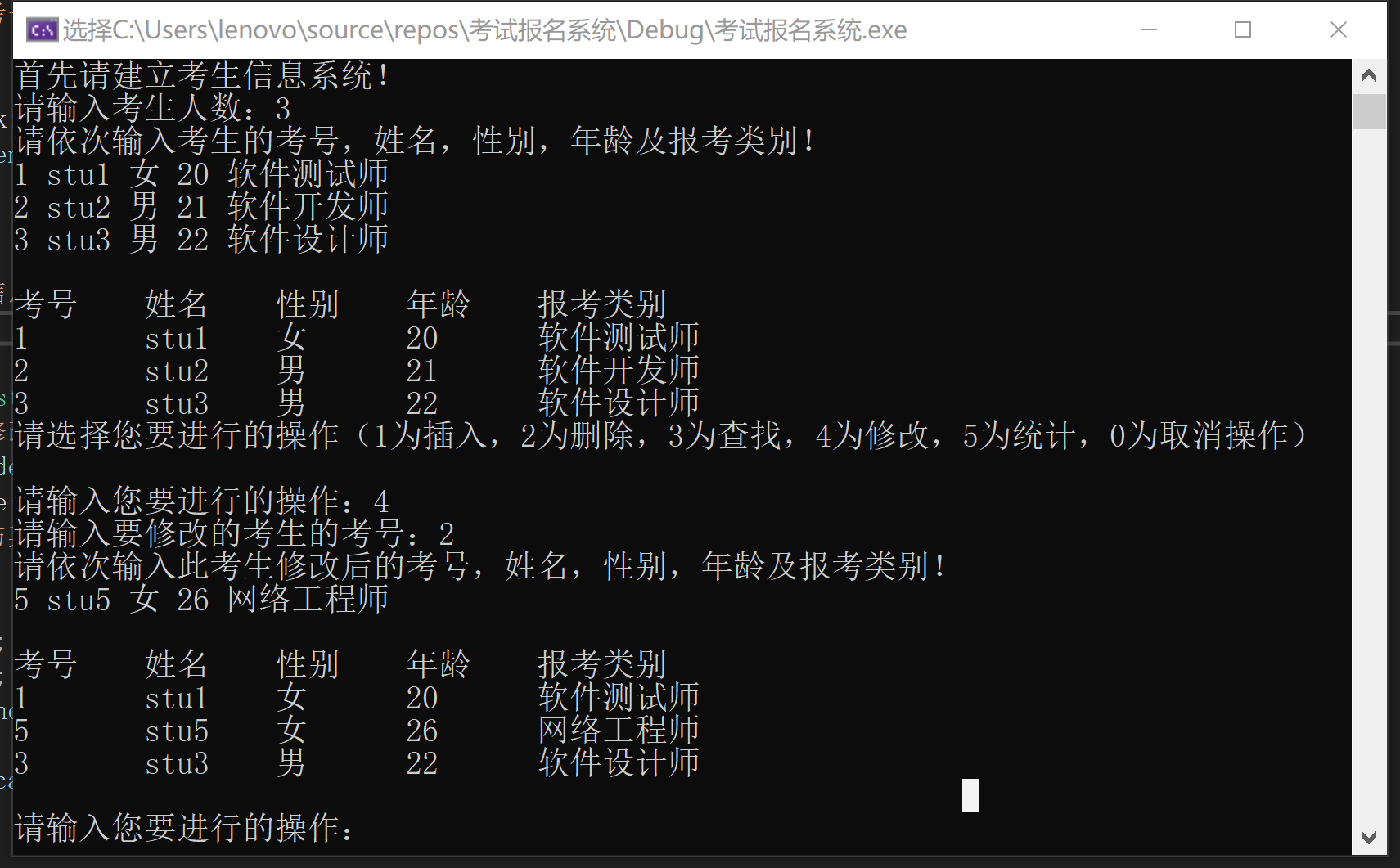
**预期结果：**

1 stu1 女 20 软件测试师

5 stu5 女 26 网络工程师

3 stu3 男 22 软件设计师

**实验结果：**



### 

### 4.1.5 统计功能测试

**测试用例：**统计当前数据

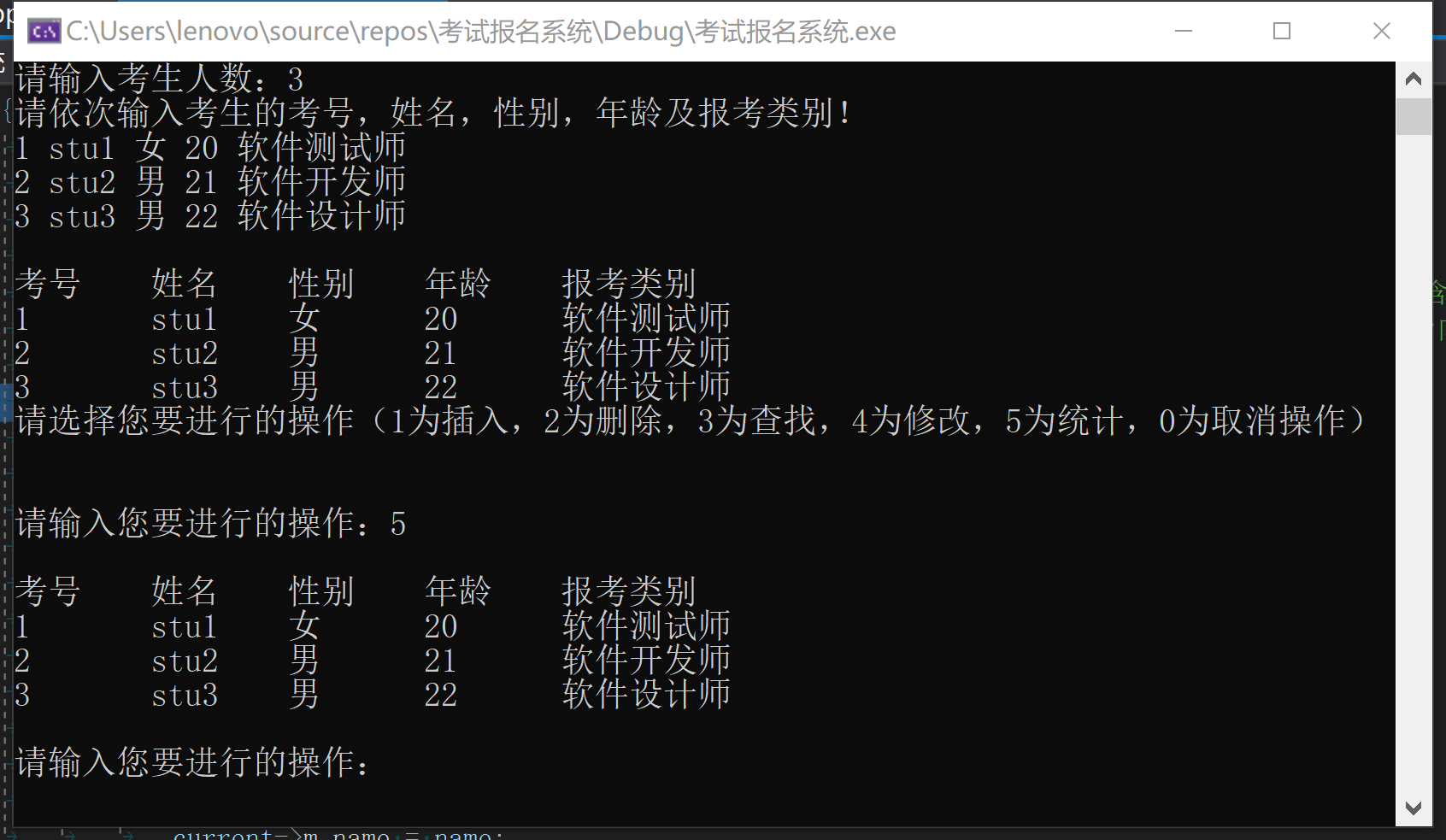
**预期结果：**

1 stu1 女 20 软件测试师

2 stu2 男 21 软件开发师

3 stu3 男 22 软件设计师

**实验结果：**



## 

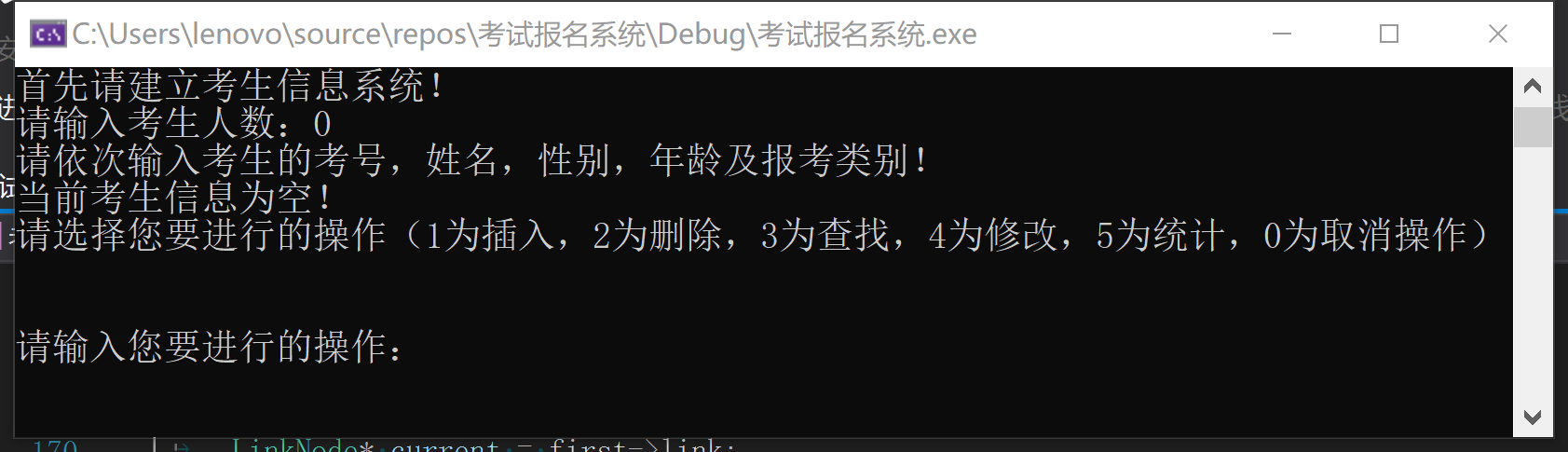
## 4.2 边界测试

## 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：**给出考生信息为空的提示，程序正常运行。

**实验结果：**

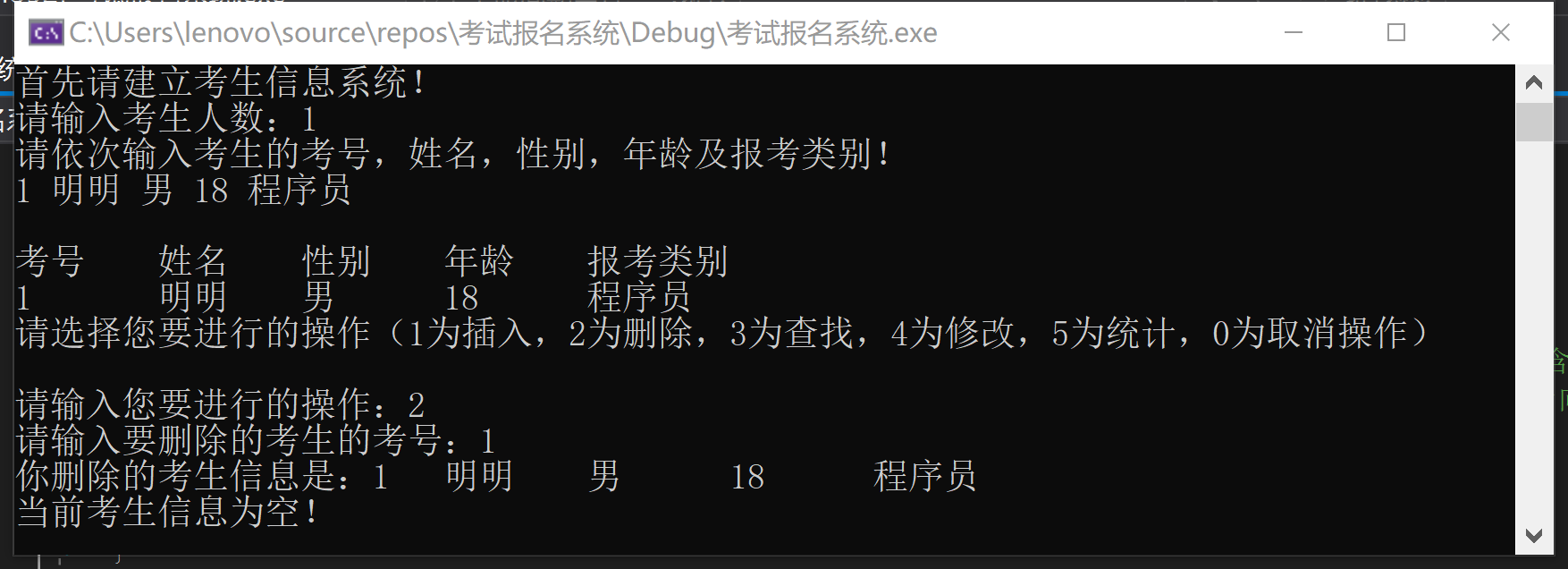


### 4.2.2 删除后链表为空

**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：**



## 

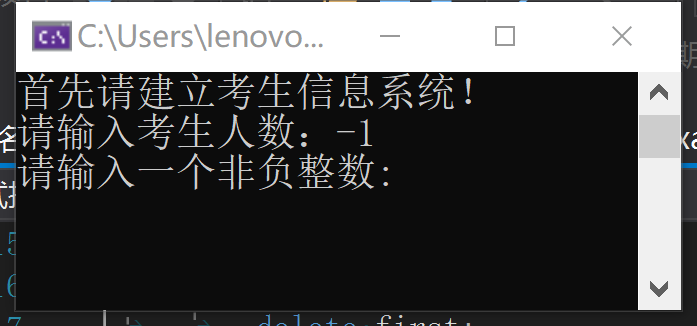
## 4.3 出错测试

### 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：**程序正常运行，给出提示让用户重新输入。

**实验结果：**

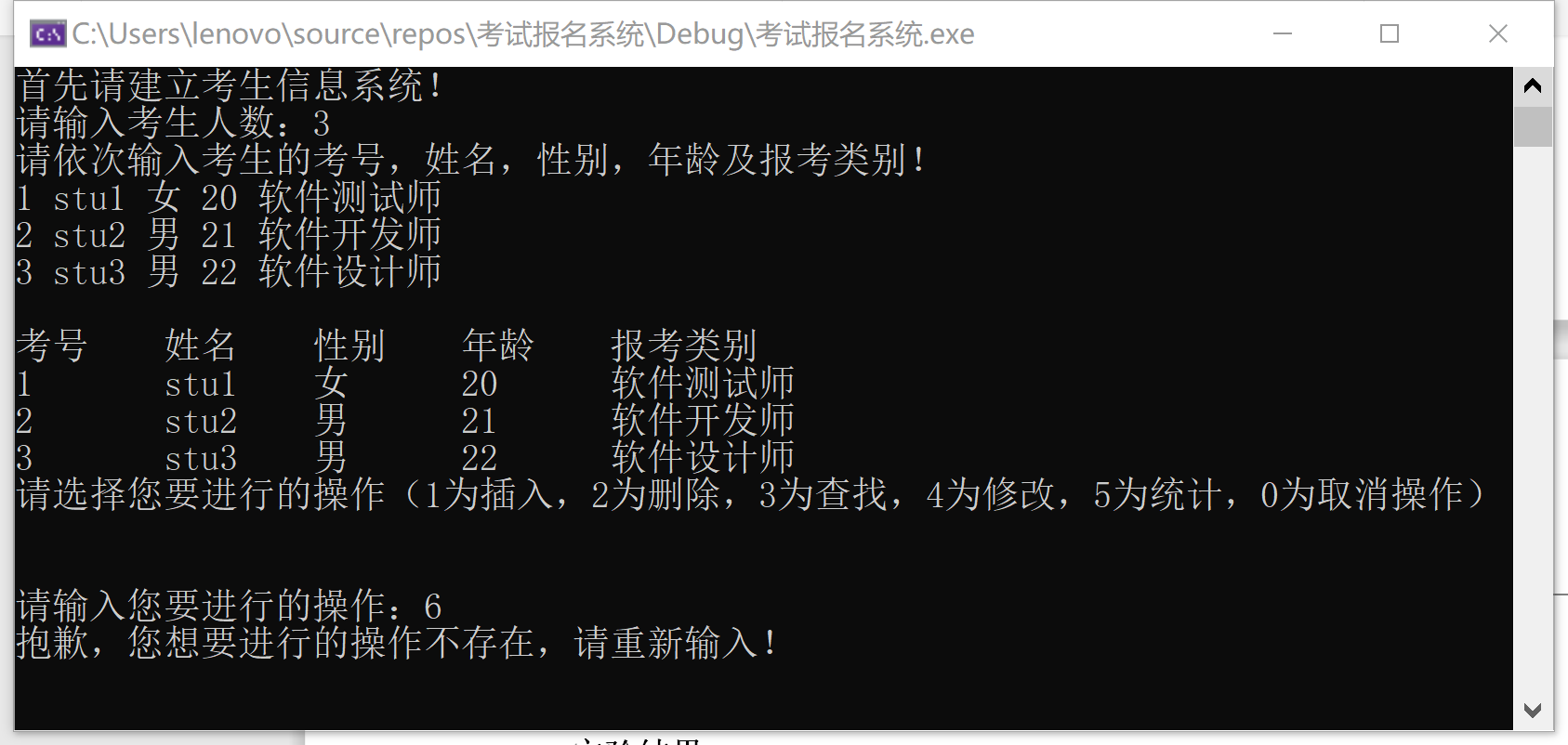
****

### 4.3.2 操作码错误

**测试用例：**输入的操作码错误

**预期结果：**程序正常运行，给出提示让用户重新输入。

**实验结果：**

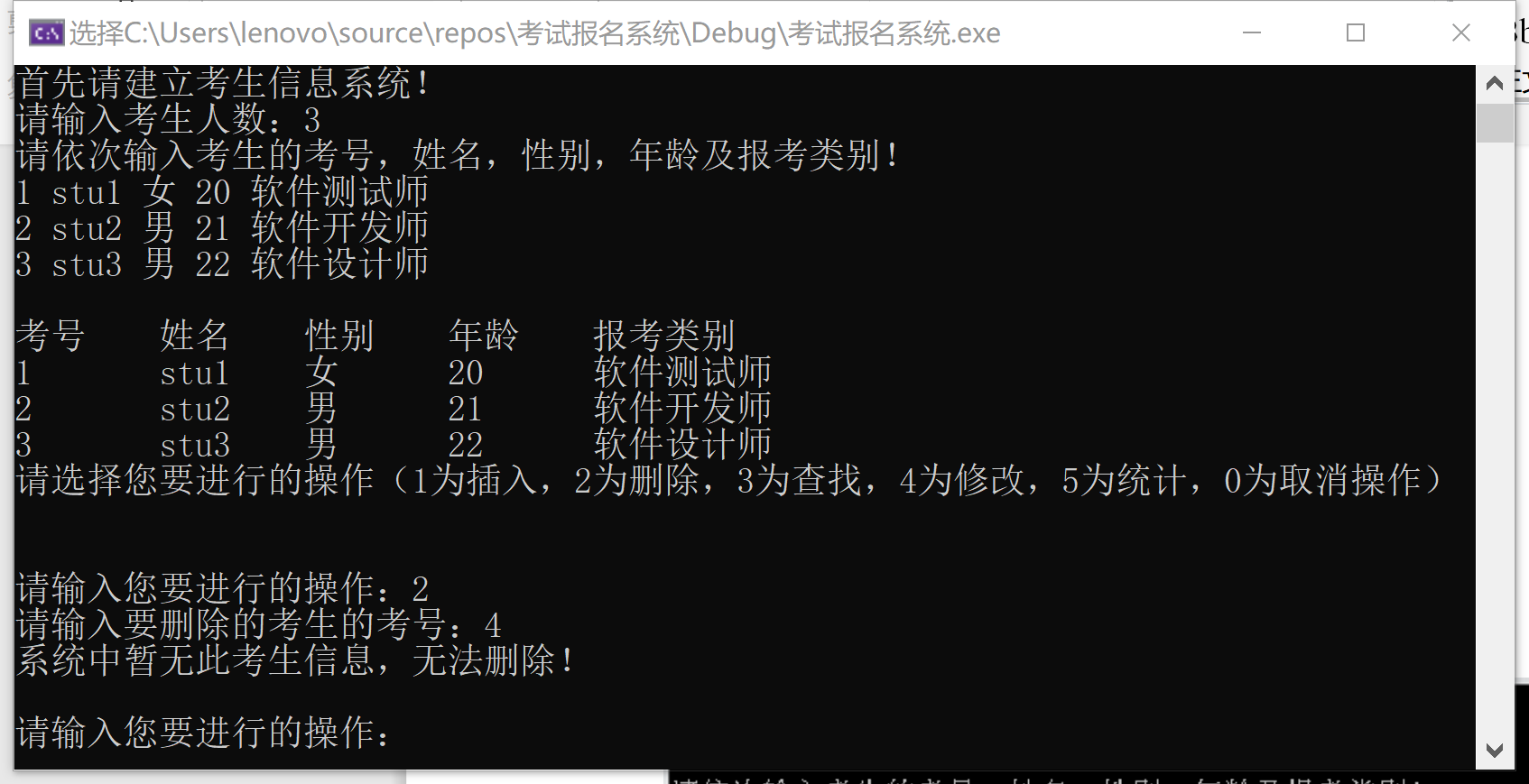


### 4.3.4 删除考号不存在

**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：**

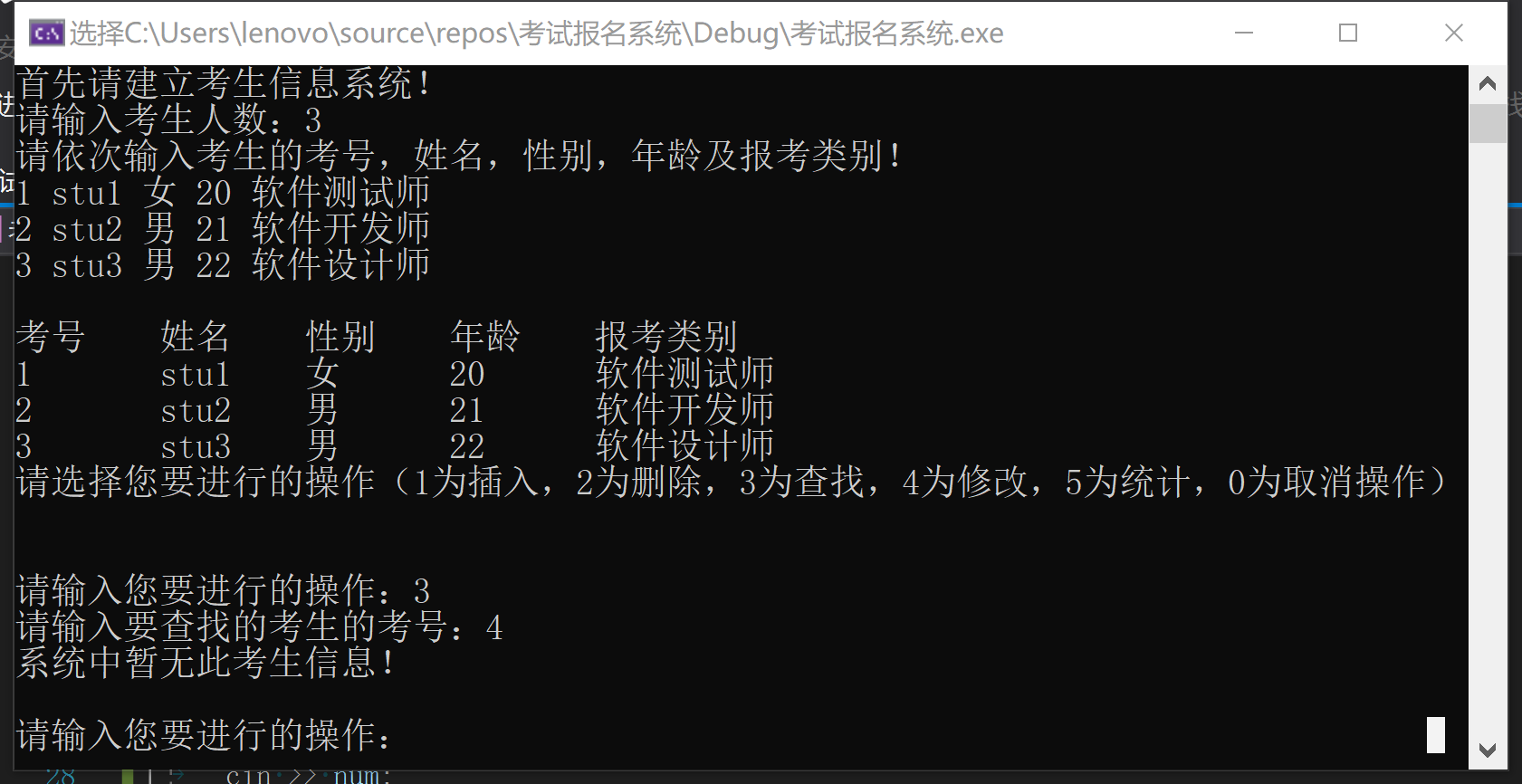
****

### 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：**

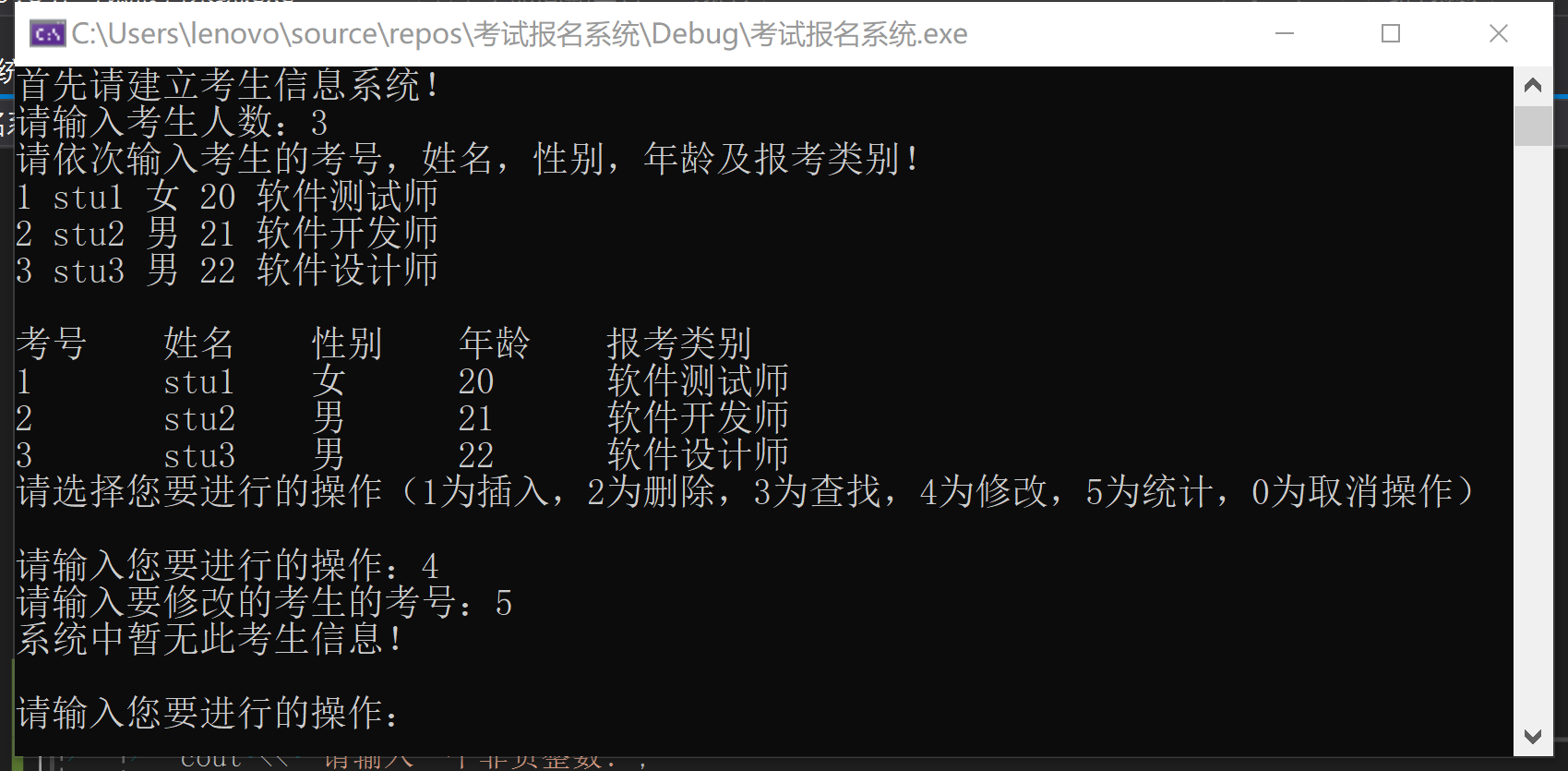
****

### 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：**



# 4.3.7 创建系统时输入错误

**测试用例：**创建系统时输入了同一考号两次或多次。

**预期结果：**程序正常运行不崩溃，提示用户补充未成功插入系统的考生信息。

**实验结果：**

