

题目一 考试报名系统

一.设计思路

考试报名系统要求建立一个存储学生信息的数据库。每个学生的信息包括，考号、姓名、性别、年龄、报考类型。

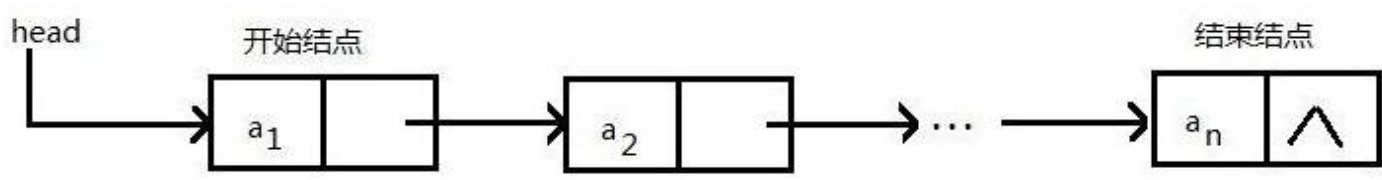
为了方便管理数据库中的信息，还需要具备输入考生信息、输出考生信息、查询考生信息、添加考生信息、修改考生信息、删除考生信息等功能。

如果采用普通的数组来存储学生信息，存在以下问题：

- 学生信息数量未知，数组初始大小不能确定
- 数据规模动态变化，数组大小不能动态变化
- 进行频繁的插入、删除操作时，数组性能差

综合几方面因素，决定采用单链表来存储学生信息。

其原理可表示如下：



每个节点有两个域，一个存储数据，另一个指向邻接的下一个节点。

用一个头指针来管理整个链表。

在实际使用链表来构建考试报名系统时，其数据结构比上图所示复杂。

需要定义许多辅助的函数来方便对于链表的操作。

节点中的数据也更为复杂。

因此，需要分别定义节点类型 `Student` 与单链表类型 `StudentDatabase`。

实现高内聚，低耦合的特点。

二.数据结构实现

1.链表结点类型（Student）

自定义了类 `class Student` 作为链表的结点，即数据库中每个学生信息。

1.1类成员

```
string _TestID;           // 考号
string _Name;             // 姓名
string _Gender;           // 性别
int _Age;                 // 年龄
string _Major;            // 报考类别

Student* _Next;           // 下一个学生节点
```

`_TestID`，`_Name`，`_Gender`，`_Major` 均为 C++ STL 容器的 `string` 类，分别代表学生考号、姓名、性别、报考类型。

`_Age` 为 `int` 类型，代表学生年龄。

`_Next` 为 `Student` 类，代表指向数据库中下一个学生信息的指针。

1.2构造函数

```
Student():_Age(0),_Next(NULL){}
```

默认构造函数，用于一般新建学生信息，默认将年龄置为0，将指向邻接节点的指针置为空。

```
Student(string id, string name, string gender, int age,string major)
:_TestID(id),_Name(name),_Gender(gender),_Age(age),_Major(major){}
```

构造函数，传入学生各个数据，直接初始化学生各项信息。

1.3信息输入函数

```
void InputInfo(string id, string name, string gender, int age,string major);
```

如果学生节点的信息在建立时未给出，可调用该函数后续输入。

1.4信息打印函数

```
void PrintInfo();
```

将该学生的信息按格式顺次输出。

1.5友元类

```
friend class StudentDatabase;
```

定义了友元类 `StudentDatabase`，即单链表类。因为单链表需要调用各个节点的内部函数，还会直接访问其成员，因此将其定义为友元类。

2.单链表类型（StudentDatabase）

自定义了类 `class StudentDatabase` 作为单链表类，即学生信息数据库。

1.1类成员

```
Student* _Data;
```

单链表只有一个成员 `_Data` 为 `Student*` 类，代表指向数据库中第一个学生信息的指针。实际起到头指针的作用，用于管理整个数据库。

1.2构造函数

```
StudentDatabase():_Data(NULL){}
```

默认构造函数，用于新建一个学生信息数据库，将 `_Data` 指针置为空。

1.3建立数据库函数

```
void BuildDatabase(string id, string name, string gender, int age,string major  
);
```

建立数据库函数，用于在程序刚开始时建立数据库。
函数的参数传入学生所有信息。

1.4成员插入函数

```
void Insert(string id, string name, string gender, int age,string major, int p  
os);
```

插入函数，用于数据库初始建立好后，后续动态插入学生信息。
将信息插入制定位置。

1.5成员删除函数

```
void Delete(Student* target);
```

删除函数，删除数据库中指定位置的节点。

1.6成员查找函数

```
Student* Find(string id);
```

查找函数。根据学号查找学生。与其他多个函数配合使用，实现多种功能。

1.7成员信息修改函数

```
void Modify(Student* target, Student info);
```

修改函数，将学生信息更新为指定值。

1.8信息输出函数

```
void Print(Student* target);
```

打印函数，输出单个成员的信息。

1.9数据库输出函数

```
void Print(Student* target);
```

数据库信息函数。将数据库中全部信息输出。

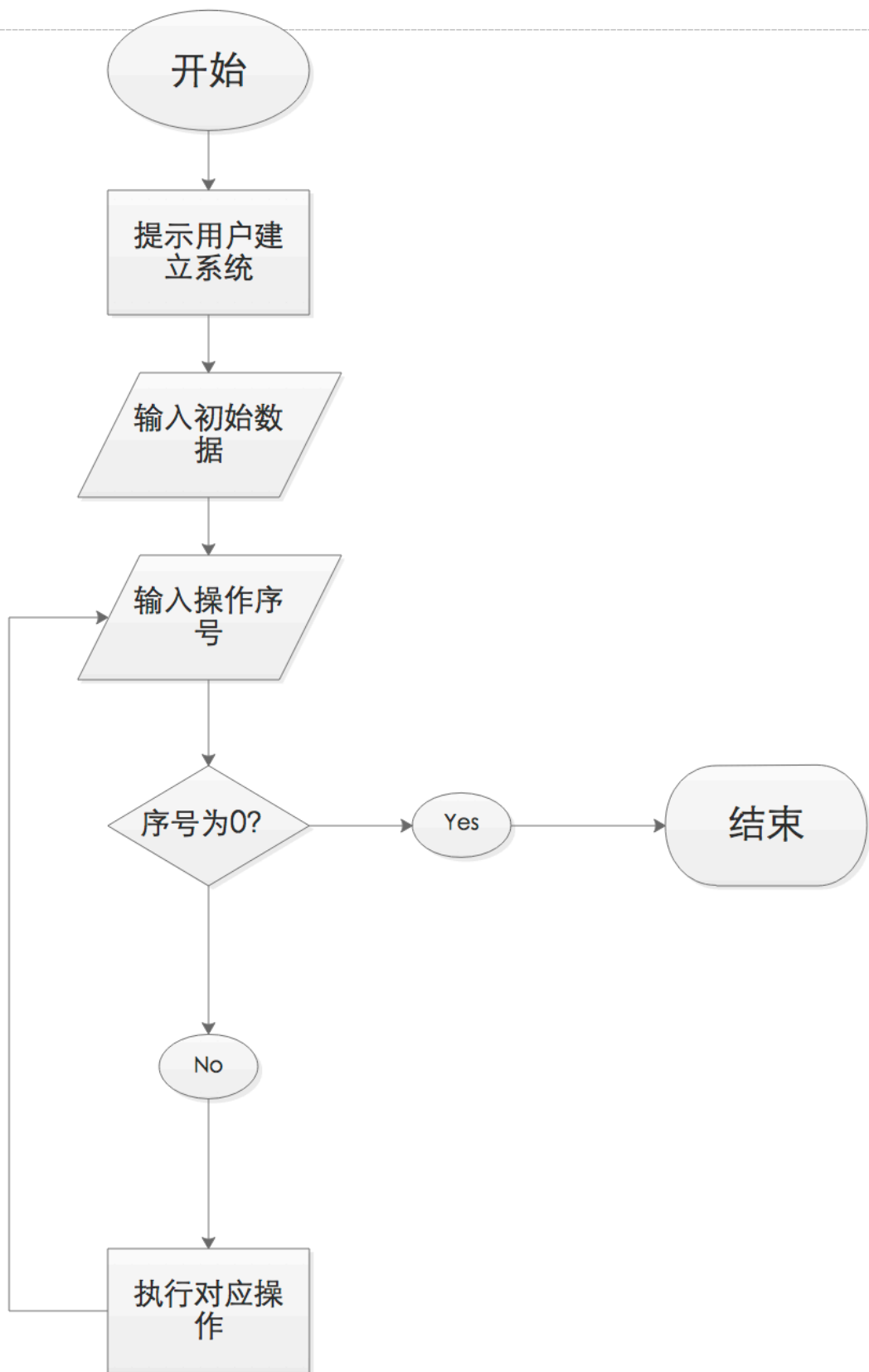
1.10计数函数

```
int count();
```

计数函数，统计学生信息的数量。

三.系统实现

1.系统执行框架



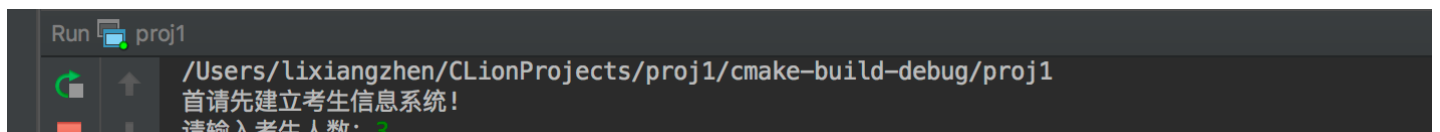
第一步先要建立一个数据库。

提示用户输入数据库学生数量，建立一个考试信息数据库。

核心代码如下：

```
/*  
 * 新建一个学生数据库  
 * */  
cout << "首请先建立考生信息系统!" << endl;  
cout << "请输入考生人数: ";  
int StudentsNum;  
cin >> StudentsNum;          // 学生人数输入
```

程序执行情况如下：



```
Run proj1  
/Users/lixiangzhen/CLionProjects/proj1/cmake-build-debug/proj1  
首请先建立考生信息系统!  
请输入考生人数: 
```

对于输入的人数要进行合法性检查。

核心代码如下：


```

/*
 * 处理考生人数非法输入
 * */
while(StudentsNum <= 0){           // 排除负数或0

    cout<<"学生人数需要是正整数!"<<endl;
    cout << "请输入考生人数";
    cin >> StudentsNum;
}

```

如果输入的学生数量不合理，循环回前一步，直到输入合适的数量。
接着要求用户输入数据库初始的信息。

核心代码如下：

```

cout << "请依次输入考生的考号，姓名，性别，年龄及报考类别!" <<endl;

/*
 * 处理输入信息
 * 建成数据库
 * */
StudentDatabase studentInfo;           // 新建一个数据库类型

for(int i = 0; i < StudentsNum; ++i){           // 循环向数据库中插入成员
    string id;
    string name;
    string gender;
    int age;
    string major;
    cin>>id>>name>>gender>>age>>major;
    studentInfo.BuildDatabase(id, name, gender, age, major);           // 调用数据库类型内部建立数据库函数
}

```

程序执行情况如下：



数据库初始建立成功后，进入功能界面。为例方便后续的操作，需要定义一批辅助临时变量，以读入和用户交互过程中的数据。

核心代码如下：

```
cout<<"请选择您要进行的操作（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）\n\n";

/*
 * 定义一系列临时变量辅助读入用户输入
 * */
int tag;           // 操作序号
string id;         // 学号
string name;       // 姓名
string gender;     // 性别
int age;          // 年龄
string major;      // 专业
int pos;          // 插入位置
Student* temp;    // 待操作节点指针
```

随后进入功能执行阶段。用户每次输入要执行的功能序号。利用 `switch - case` 语句对与用户的输入进行选择。然后跳转到对应的地方执行相关功能。

核心代码如下：

```
while(cin>>tag&&tag != 0){          // 循环读入用户操作
    switch (tag)
    {
        case 1:
            cout<<"请输入你要插入的考生位置：";
            cin>>pos;                  // 输入插入位置
            cout << "请依次输入你要插入的考生的考号，姓名，性别，年龄及报考类别!" <<endl
;

            cin>>id>>name>>gender>>age>>major;
            if(studentInfo.Find(id) != NULL){          // 排除学号重复
                cout<<"考号已经存在! \n";
                break;
            }
            studentInfo.Insert(id, name, gender, age, major, pos);          // 插入新节点

            cout<<endl;
            studentInfo.Display();
            break;
        case 2:
            cout<<"请输入要删除考生的考号：";
            cin>>id;
            if((temp = studentInfo.Find(id)) == NULL)          // 查找学生存在与否
                cout<<"没有该考生"<<endl;
            else{
                studentInfo.Delete(temp);          // 删除该节点
                cout<<endl;
                studentInfo.Display();
            }
            break;
        case 3:
            cout<<"请输入要查找的考生的考号：";
            cin>>id;
            temp = studentInfo.Find(id);          // 查找学生
            if(temp != NULL){          // 输出信息
```

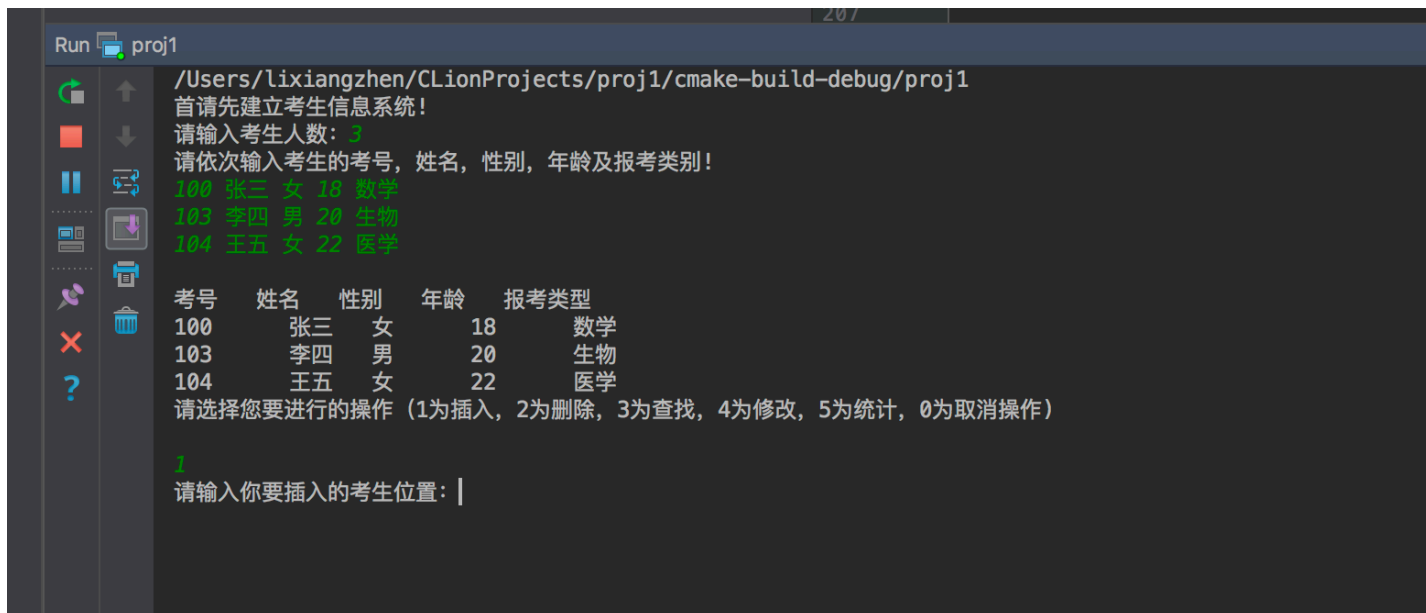
```

        cout<<"考号      "
            <<"姓名      "
            <<"性别      "
            <<"年龄      "
            <<"报考类型"<<endl;
        studentInfo.Print(temp);
    }
    else
        cout<<"没有该考生"<<endl;
    break;
case 4:
    cout<<"请输入要修改的考生的考号: ";
    cin>>id;
    temp = studentInfo.Find(id);          // 查找学生是否存在
    if(temp != NULL){
        cout<<"请依次输入修改后的考生的考号, 姓名, 性别, 年龄及报考类别:\n";
        cin>>id>>name>>gender>>age>>major;
        studentInfo.Modify(temp, Student(id, name, gender, age, major)
);
    }
    else                                // 不存在输出提示
        cout<<"没有该考生"<<endl;
    cout<<endl;
    studentInfo.Display();
    break;
case 5:
    studentInfo.Display();
    cout<<"数据库中一共有"<<studentInfo.count()<<"个学生\n";          //
输出统计信息

    break;
}
cout<<"请选择您要进行的操作 (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)
\n\n";
}

```

程序执行情况如下:



2.新建数据库功能

核心代码如下：

```

/*
 * 建立数据库函数
 * 将数据插入链表尾部
 * */
void StudentDatabase::BuildDatabase(string id, string name, string gender, int age
,string major){

    if(_Data == NULL){    // 处理数据库为空的情况
        _Data = new Student(id, name, gender, age, major);
        _Data->_Next = NULL;
    }
    else{
        Student* tail = _Data;
        while(tail->_Next != NULL)    // 找到尾节点
            tail = tail->_Next;

        tail->_Next = new Student(id, name, gender, age, major);    // 插入尾部
        tail->_Next->_Next = NULL;
    }
}
}

```

该函数实现新建了一个空的信息管理数据库后向其中插入初始数据的功能。首先插入第一个数据时，进入判断空数据库的分支，插入数据的同时，初始化头指针 `_Data` 。后续插入的数据被顺序的安排在链表尾部。

关联调用情况如下：

```

    string gender;
    int age;
    string major;
    cin>>id>>name>>gender>>age>>major;
    studentInfo.BuildDatabase(id, name, gender, age, major);    // 调用数据库类型内部建立数据库函数
}

if(studentInfo._Data == NULL){    // 处理为合法输入数据的情况

```

3.插入功能

核心代码如下：

```
cout<<"请输入你要插入的考生位置：";
cin>>pos;                                // 输入插入位置
cout << "请依次输入你要插入的考生的考号，姓名，性别，年龄及报考类别！" <<endl
;

cin>>id>>name>>gender>>age>>major;
if(studentInfo.Find(id) != NULL){        // 排除学号重复
    cout<<"考号已经存在！\n";
    break;
}
studentInfo.Insert(id, name, gender, age, major, pos);    // 插入

cout<<endl;
studentInfo.Display();
```

- 要求用户输入插入数据的信息。
- 判断学号是否存在
- 调用 `void StudentDatabase::Insert()` 插入新节点信息

函数实现如下：

```

/*
 * 节点插入函数
 * 根据参数给出的位置插入数据
 * */
void StudentDatabase::Insert(string id, string name, string gender, int age, string
major, int pos){

    if(pos == 1){                // 处理头部插入
        Student* temp = new Student(id, name, gender, age, major);
        temp->_Next = _Data;
        _Data = temp;
        return;
    }

    Student* pre = _Data;
    pos -= 2;
    while(pos != 0 && pre != NULL){        // 找到插入位置前一个节点指针
        pre = pre->_Next;
        --pos;
    }

    if(pos != 0){                // 处理非法插入位置
        cout<<"非法插入位置! \n";
        return;
    }
    else{
        Student* temp = new Student(id, name, gender, age, major);        // 插入对应
位置
        temp->_Next = pre->_Next;
        pre->_Next = temp;
    }

}

```

在已经建立好数据库的情况下，向其中插入新的数据节点。

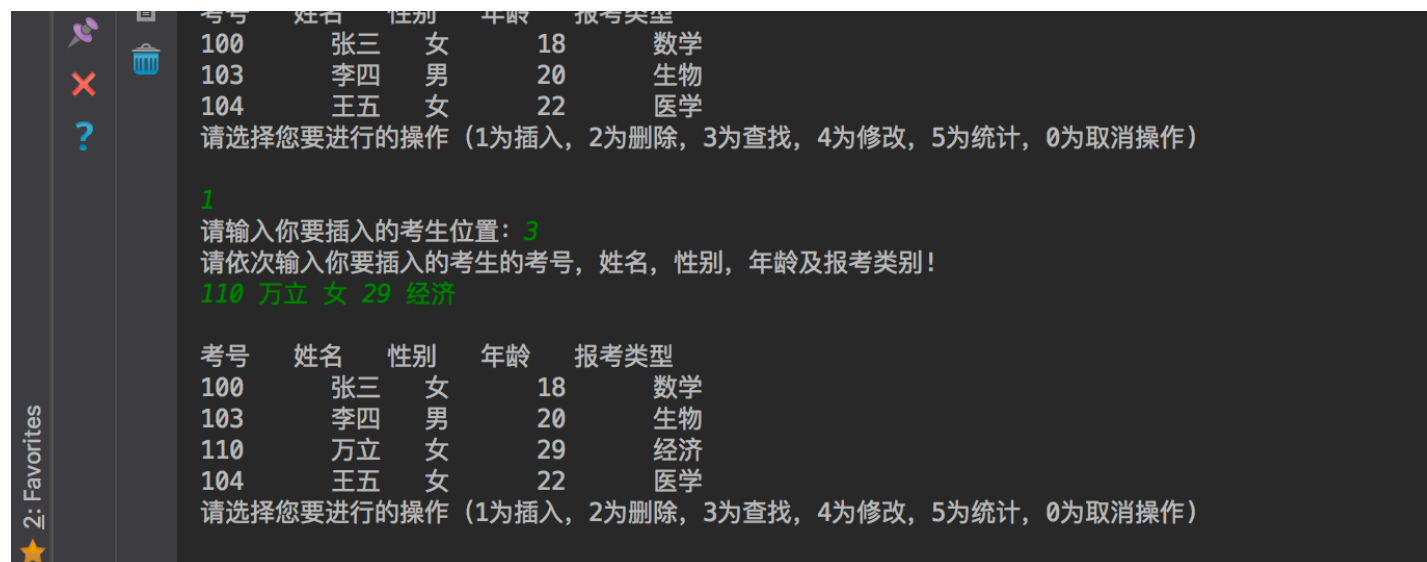
插入位置有参数 `int pos`，传入。

其他参数传入数据的具体信息。

- 先处理在头部插入的情况。

- 随后寻找插入位置指针
- 判断位置合法性
- 执行插入

程序执行情况如下：



4.删除功能

核心代码如下：

```
cout<<"请输入要删除考生的考号: ";
cin>>id;
if((temp = studentInfo.Find(id)) == NULL)           // 查找学生存在与否
    cout<<"没有该考生"<<endl;
else{
    studentInfo.Delete(temp);                       // 删除该节点
    cout<<endl;
    studentInfo.Display();
}
```

- 调用查找函数搜索要删除节点
- 判断搜索结果
- 调用 `void StudentDatabase::Delete()` 删除节点

函数实现如下：

```

/*
 * 删除函数
 * 将参数指针对应节点删除
 * */
void StudentDatabase::Delete(Student* target){

    if(target == NULL)        // 处理删除对象为空的情况
        return;

    if(target == _Data){
        _Data = _Data->_Next;        // 处理头节点删除
    }
    else{
        Student* pre = _Data;
        while(pre->_Next != target)    // 找到被删节点前一个位置
            pre = pre->_Next;

        pre->_Next = target->_Next;    // 被删节点出链
    }

    delete target;        // 释放空间
}

```

- 处理被删对象为空的情况
- 找到前驱指针
- 删除节点

程序执行情况如下：

```
Run: proj1 proj1 proj1
/Users/lixiangzhen/CLionProjects/proj1/cmake-build-debug/proj1
首请先建立考生信息系统！
请输入考生人数: 3
请依次输入考生的考号, 姓名, 性别, 年龄及报考类别!
100 张三 女 18 数学
103 李四 男 20 生物
104 王五 女 22 医学

考号  姓名  性别  年龄  报考类型
100   张三  女    18    数学
103   李四  男    20    生物
104   王五  女    22    医学
请选择您要进行的操作 (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)

请选择您要进行的操作: 2
请输入要删除考生的考号: 100
您要删除的考生的信息是: 100      张三   女      18      数学

考号  姓名  性别  年龄  报考类型
103   李四  男    20    生物
104   王五  女    22    医学
请选择您要进行的操作 (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)

请选择您要进行的操作:
```

5.查找功能

核心代码如下:

```

        cout<<"请输入要查找的考生的考号: ";
        cin>>id;
        temp = studentInfo.Find(id);           // 查找学生
        if(temp != NULL){                     // 输出信息
            cout<<"考号      "
                <<"姓名      "
                <<"性别      "
                <<"年龄      "
                <<"报考类型"<<endl;
            studentInfo.Print(temp);
        }
        else
            cout<<"没有该考生"<<endl;
    }
}

```

- 调用 `Student* StudentDatabase::Find()` 函数查找该节点
- 判断查找结果
- 格式化输出

其中, `Find()` 函数实现如下:

```

/*
 * 查找函数
 * 根据参数的学号查找
 * 返回对象节点指针
 * */
Student* StudentDatabase::Find(string id){
    Student* temp = _Data;

    while(temp != NULL&&temp->_TestID != id)    // 遍历查找
        temp = temp->_Next;

    return temp;
}

```

- 遍历数据链表查找信息
- 查到返回信息指针
- 查不到指针为NULL

程序执行情况如下：

```
你要删除的考生的信息是： 100      张三      女      18      数学

考号   姓名   性别   年龄   报考类型
103     李四   男     20     生物
104     王五   女     22     医学
请选择您要进行的操作（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）

请选择您要进行的操作： 3
请输入要查找的考生的考号： 103
考号   姓名   性别   年龄   报考类型
103     李四   男     20     生物
请选择您要进行的操作（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）

请选择您要进行的操作： 3
请输入要查找的考生的考号： 140
没有该考生
请选择您要进行的操作（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）

请选择您要进行的操作：
```

此外， `Find()` 函数作为基础功函数，被其他功能函数多次调用。

关联调用情况如下：

```
cin>>pos; // 输入插入位置
cout << "请依次输入你要插入的考生的考号，姓名，性别，年龄及报考类别!" <<endl;
cin>>id>>name>>gender>>age>>major;
if(studentInfo.Find(id) != NULL){ // 排除学号重复
    cout<<"考号已经存在! \n";
    break;
}
```

```

cout<<"请输入要删除考生的考号：";
cin>>id;
if((temp = studentInfo.Find(id)) == NULL)    // 查找学生存在与否
    cout<<"没有该考生"<<endl;
else{
    cout<<"你要删除的考生的信息是：";
    temp->PrintInfo();
}

```

```

35         break;
36     case 4:
37         cout<<"请输入要修改的考生的考号：";
38         cin>>id;
39         temp = studentInfo.Find(id);    // 查找学生是否存在
40         if(temp != NULL){
41             cout<<"请依次输入修改后的考生的考号，姓名，性别，年龄及报考类别:\n";

```

6.修改功能

核心代码如下：

```

        cout<<"请输入要修改的考生的考号：";
        cin>>id;
        temp = studentInfo.Find(id);    // 查找学生是否存在
        if(temp != NULL){
            cout<<"请依次输入修改后的考生的考号，姓名，性别，年龄及报考类别:\n";
            cin>>id>>name>>gender>>age>>major;
            studentInfo.Modify(temp, Student(id, name, gender, age, major)
        );
        }
        else    // 不存在输出提示
            cout<<"没有该考生"<<endl;
        cout<<endl;
        studentInfo.Display();
    }
}

```

- 调用查找函数搜索要修改的考生指针
- 判断修改结果
- 执行修改

函数实现如下：

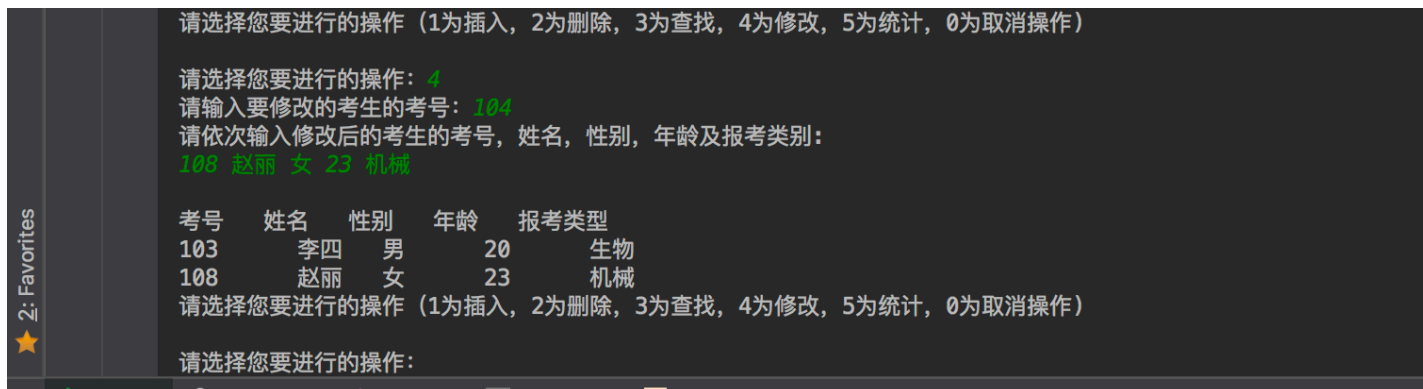
```
/*
 * 修改函数
 * 将参数指针对应节点信息修改为参数内容
 * */
void StudentDatabase::Modify(Student* target, Student info){

    target->_TestID = info._TestID;      // 依次修改对应信息
    target->_Age = info._Age;
    target->_Gender = info._Gender;
    target->_Major = info._Major;
    target->_Name = info._Name;

}
```

依次将新的信息赋值到位。

程序执行情况如下：



7.统计功能

核心代码如下:

```
studentInfo.Display();  
cout<<"数据库中共有"<<studentInfo.count()<<"个学生\n";           //
```

输出统计信息

- 输出数据库中所有信息
- 输出数据库中学生数量

主要函数实现如下:

```

/*
 * 展示函数
 * 将数据库中所有信息输出
 * */
void StudentDatabase::Display(){

    if(_Data == NULL){          // 处理数据库为空情况
        cout << "空表! ";
        return;
    }

    cout<<"考号   "
        <<"姓名   "
        <<"性别   "
        <<"年龄   "
        <<"报考类型"<<endl;

    Student* temp = _Data;      // 循环依次输出所有节点信息
    while(temp != NULL){
        temp->PrintInfo();
        temp = temp->_Next;
    }

}

```

展示函数，遍历格式化输出数据库中所有信息。

```

/*
 * 计数函数
 * 统计数据库中 学生数量
 * */
int StudentDatabase::count() {
    int ans = 0;          // 记录学生数量
    Student* temp = _Data;
    while(temp != NULL){   // 遍历统计
        ++ans;
        temp = temp->_Next;
    }

    return ans;
}

```

计数函数，统计学生数量。

程序执行情况如下：

请选择您要进行的操作（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）

请选择您要进行的操作：5

数据库中一共有3个学生

考号	姓名	性别	年龄	报考类型
100	张三	女	18	数学
103	李四	男	20	生物
104	王五	女	22	医学

请选择您要进行的操作（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）

请选择您要进行的操作：

四.测试

1.基本功能测试

测试用例

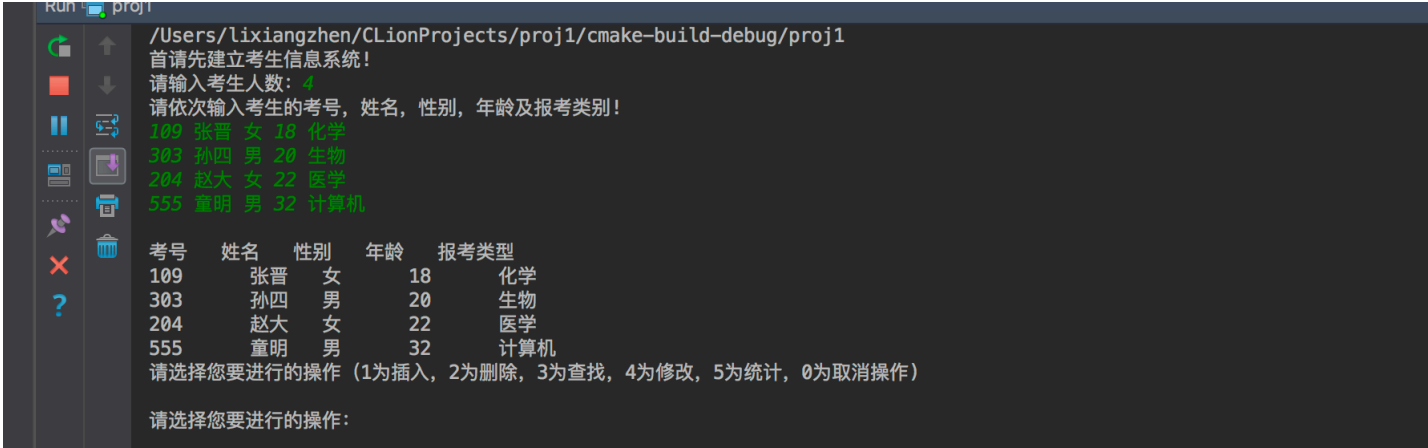
初始考生数量：

初始考生信息：

109	张晋	女	18	化学
303	孙四	男	20	生物
204	赵大	女	22	医学
555	童明	男	32	计算机

建立初始数据库

程序执行情况如下：



插入新考生

测试用例

插入位置：

插入考生信息：

206 王丽 女 17 经济

程序执行情况如下：

✖

?

考号	姓名	性别	年龄	报考类型
109	张晋	女	18	化学
303	孙四	男	20	生物
204	赵大	女	22	医学
555	童明	男	32	计算机

请选择您要进行的操作（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）

请选择您要进行的操作：1

请输入你要插入的考生位置：2

请依次输入你要插入的考生的考号，姓名，性别，年龄及报考类别！

206 王丽 女 17 经济

考号	姓名	性别	年龄	报考类型
109	张晋	女	18	化学
206	王丽	女	17	经济
303	孙四	男	20	生物
204	赵大	女	22	医学
555	童明	男	32	计算机

请选择您要进行的操作（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）

请选择您要进行的操作：

删除考生信息

删除考生考号：

303

程序执行情况如下：

考号	姓名	性别	年龄	报考类型
109	张晋	女	18	化学
206	王丽	女	17	经济
303	孙四	男	20	生物
204	赵大	女	22	医学
555	童明	男	32	计算机

请选择您要进行的操作（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）

请选择您要进行的操作：2
请输入要删除考生的考号：303
您要删除的考生的信息是：303 孙四 男 20 生物

考号	姓名	性别	年龄	报考类型
109	张晋	女	18	化学
206	王丽	女	17	经济
204	赵大	女	22	医学
555	童明	男	32	计算机

请选择您要进行的操作（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）

请选择您要进行的操作：

查找考生

查找考生考号：

204

程序执行情况如下：

您要删除的考生的信息是：303 孙四 男 20 生物

考号	姓名	性别	年龄	报考类型
109	张晋	女	18	化学
206	王丽	女	17	经济
204	赵大	女	22	医学
555	童明	男	32	计算机

请选择您要进行的操作（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）

请选择您要进行的操作：3
请输入要查找的考生的考号：204
考号 姓名 性别 年龄 报考类型
204 赵大 女 22 医学
请选择您要进行的操作（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）

请选择您要进行的操作：

修改考生信息

修改考生考号：

109

修改后考生信息：

109 周吴 女 20 算法

程序执行情况如下：

2: Favorites

★

4: Run

6: TODO

CMake

Terminal

Q: Messages

```
考号      姓名      性别      年龄      报考类型
109      张晋      女      18      化学
请选择您要进行的操作（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）

请选择您要进行的操作：4
请输入要修改的考生的考号：109
请依次输入修改后的考生的考号，姓名，性别，年龄及报考类别：
109 周吴 女 20 算法

考号      姓名      性别      年龄      报考类型
109      周吴      女      20      算法
206      王丽      女      17      经济
204      赵大      女      22      医学
555      童明      男      32      计算机
请选择您要进行的操作（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）

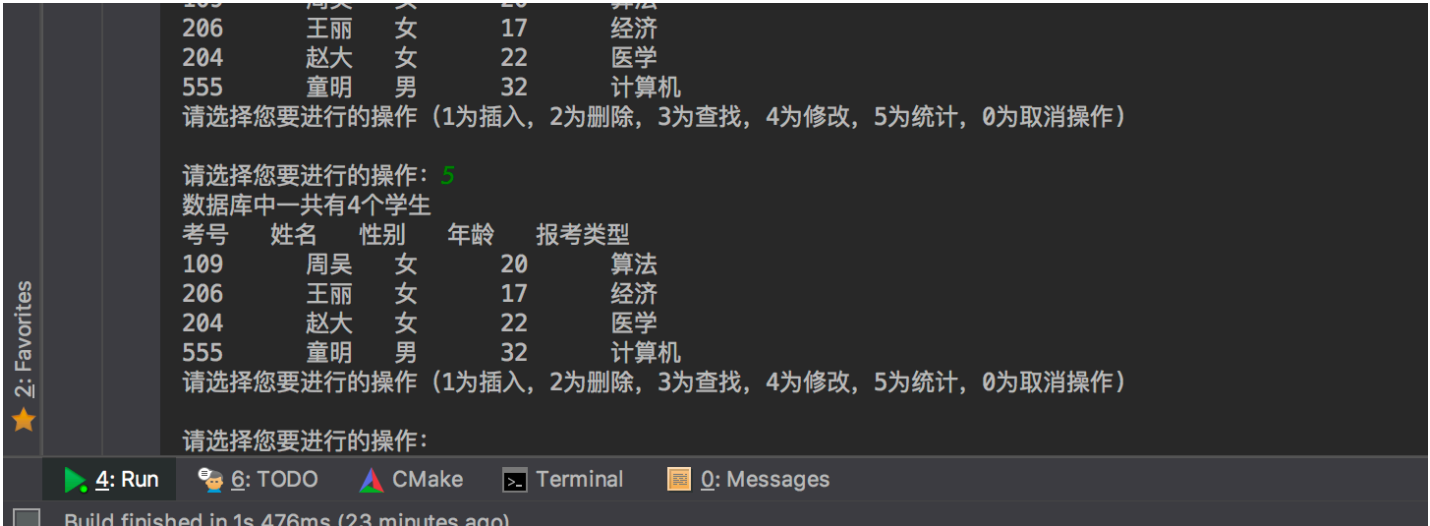
请选择您要进行的操作：
```

信息统计

当前考生信息：

109	周吴	女	20	算法
206	王丽	女	17	经济
204	赵大	女	22	医学
555	童明	男	32	计算机

程序执行情况如下：

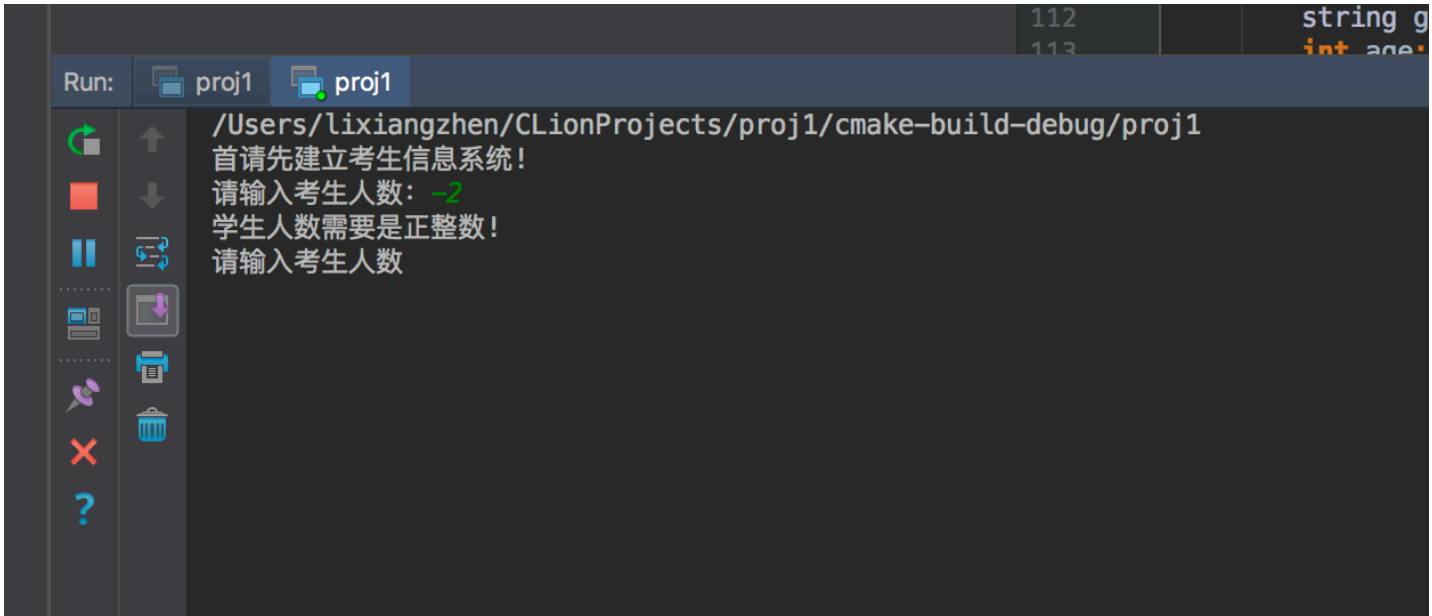


2.边界测试

2.1输入的考生人数非法

考试人数：

程序执行情况如下：



2.2插入重复学号

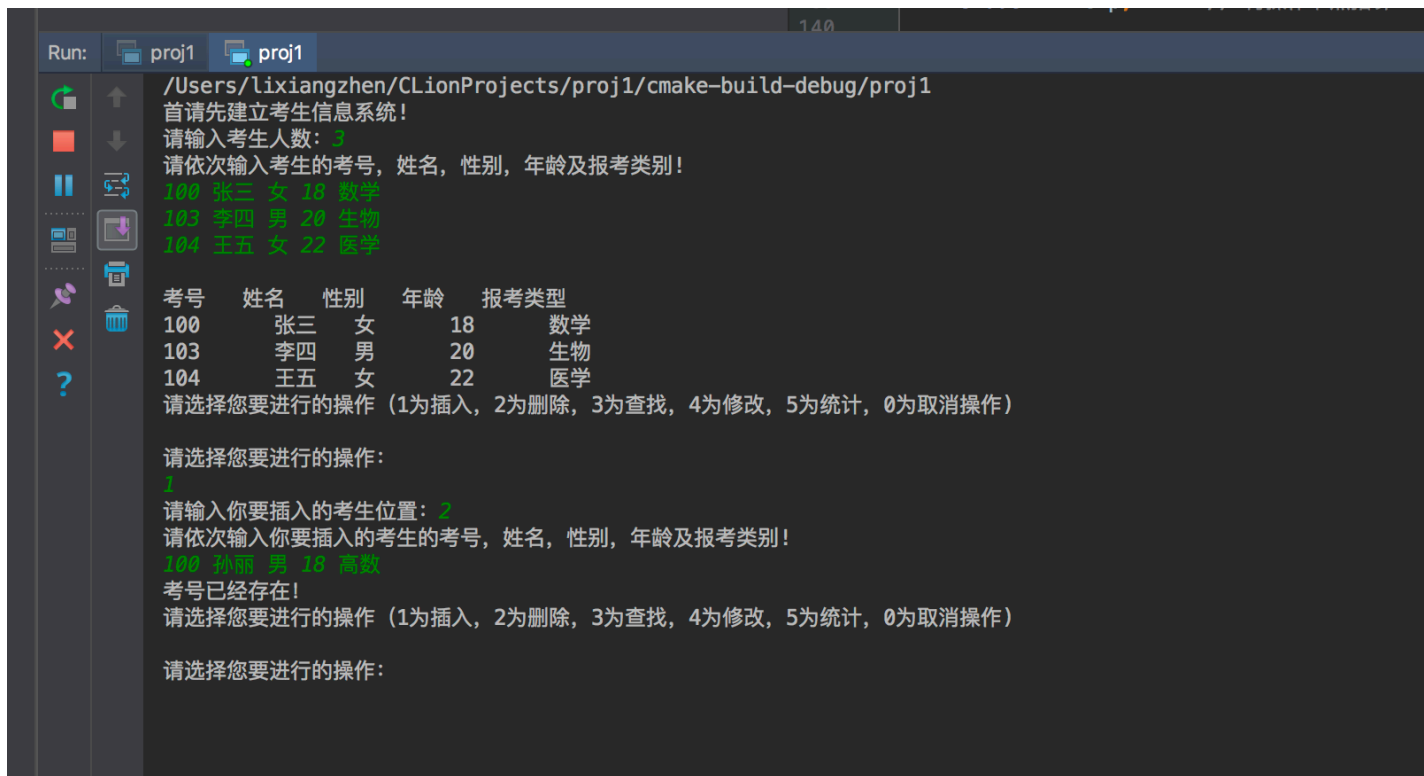
当前考生信息：

100	张三	女	18	数学
103	李四	男	20	生物
104	王五	女	22	医学

新插入考试信息：

100	孙丽	男	18	高数
-----	----	---	----	----

程序执行情况如下：



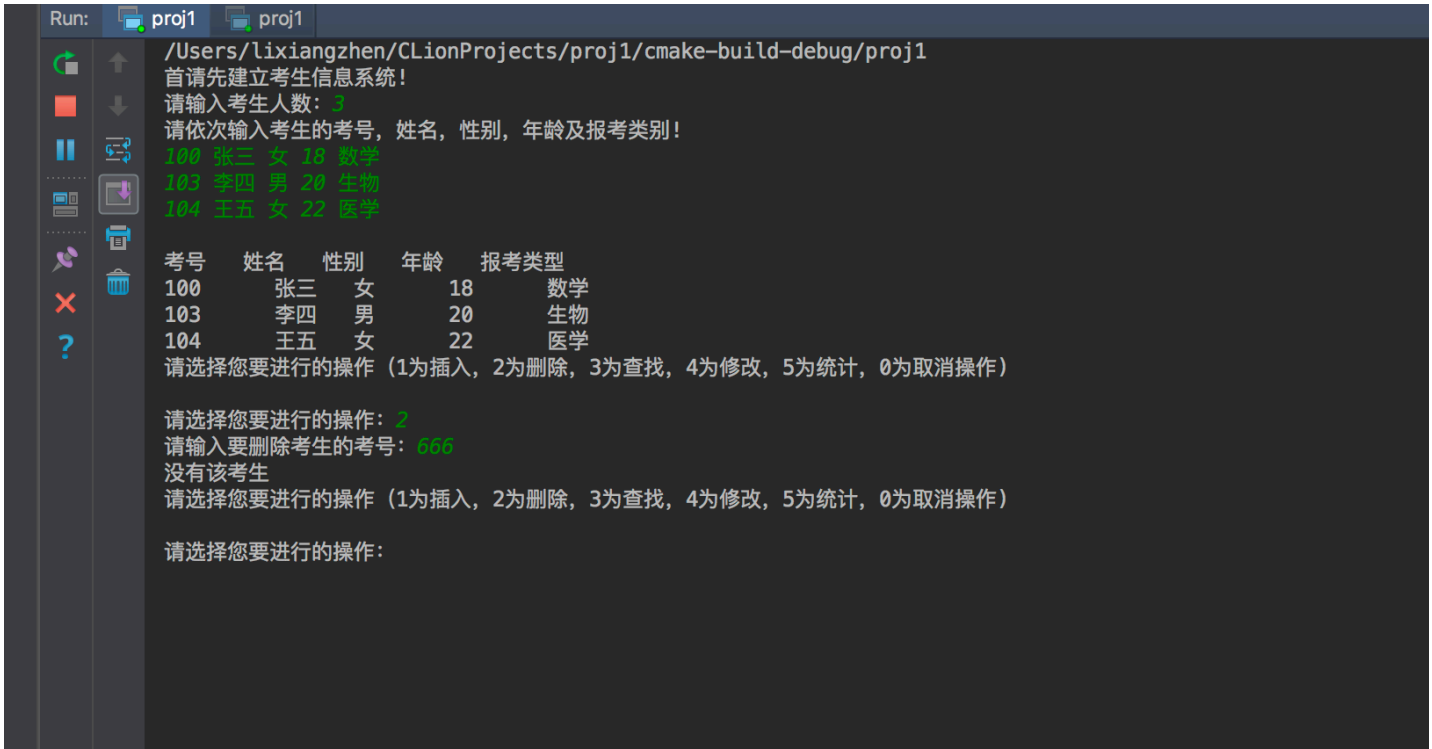
2.3被操作考生不存在

当前考生信息:

100	张三	女	18	数学
103	李四	男	20	生物
104	王五	女	22	医学

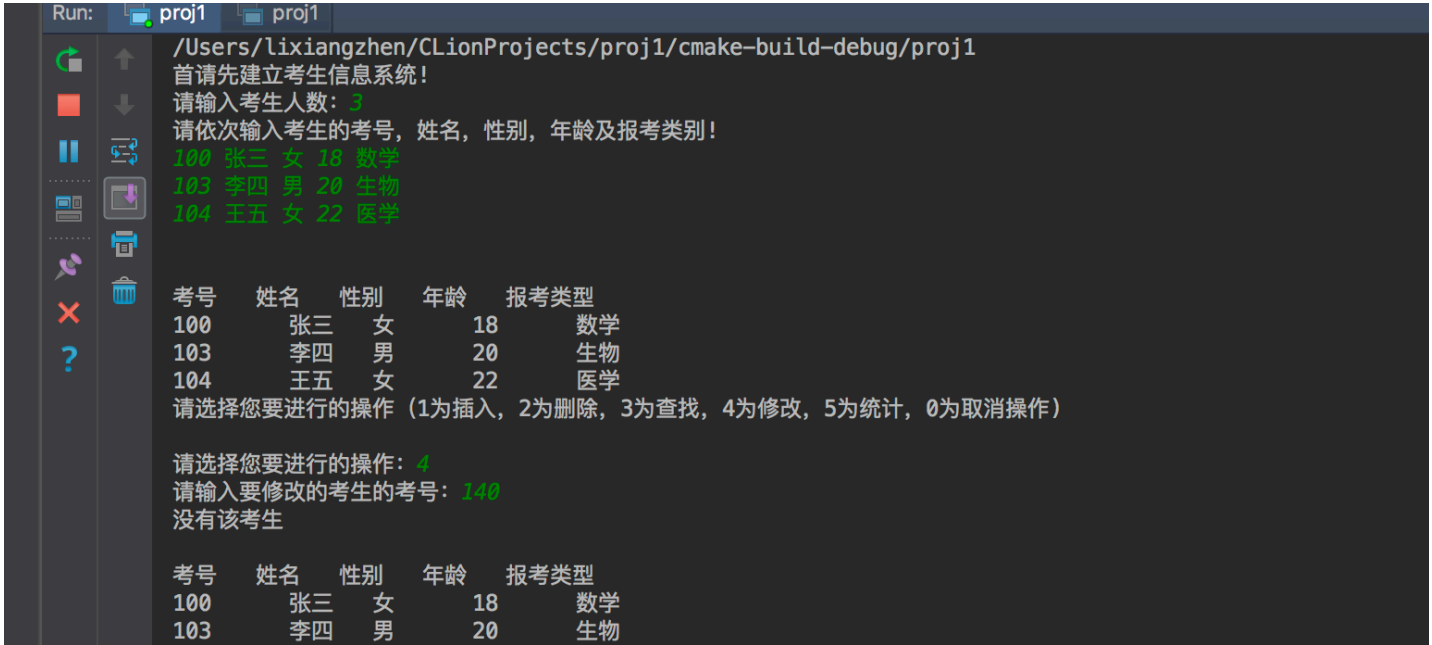
待删除考生考号: 666

程序执行情况如下:



待修改考生考号: 140

程序执行情况如下:



2.4在头部插入考生信息

当前考生信息：

100	张三	女	18	数学
103	李四	男	20	生物
104	王五	女	22	医学

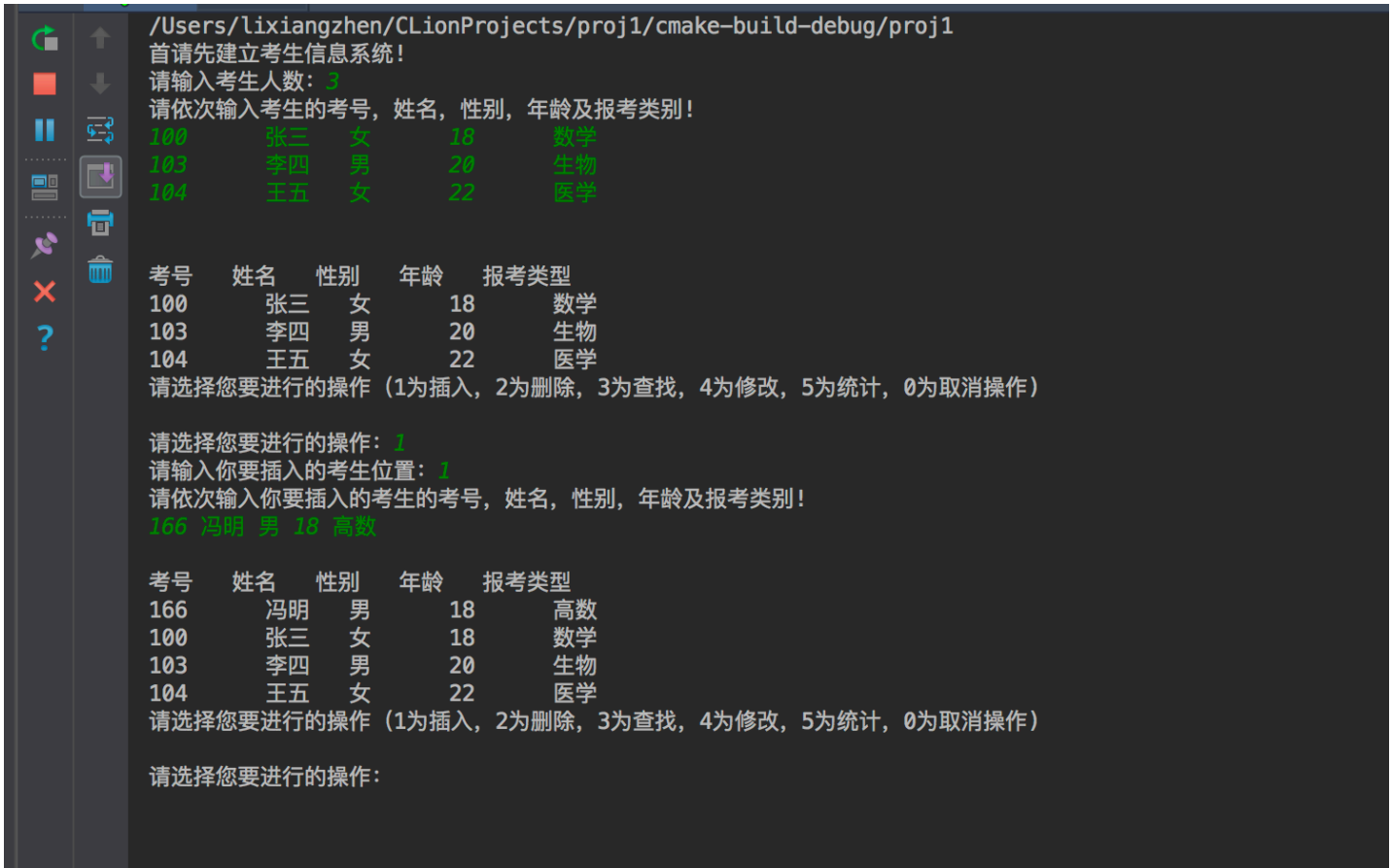
新插入考试信息：

166	冯明	男	18	高数
-----	----	---	----	----

新插入位置：

1

程序执行情况如下：



2.5表头删除

当前考生信息：

100	张三	女	18	数学
103	李四	男	20	生物
104	王五	女	22	医学

删除考生考号：

程序执行情况如下：

```
/Users/lixiangzhen/CLionProjects/proj1/cmake-build-debug/proj1
首请先建立考生信息系统！
请输入考生人数: 3
请依次输入考生的考号，姓名，性别，年龄及报考类别！
100    张三    女    18    数学
103    李四    男    20    生物
104    王五    女    22    医学

考号    姓名    性别    年龄    报考类型
100      张三    女      18      数学
103      李四    男      20      生物
104      王五    女      22      医学
请选择您要进行的操作（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）

请选择您要进行的操作: 2
请输入要删除考生的考号: 100
你要删除的考生的信息是: 100      张三    女      18      数学

考号    姓名    性别    年龄    报考类型
103      李四    男      20      生物
104      王五    女      22      医学
请选择您要进行的操作（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）

请选择您要进行的操作:
```

2.6空数据库

当前考生信息：

103	李四	男	20	生物
104	王五	女	22	医学

程序执行情况如下：

