

大类基础课程大型实验

**2018/2019(2)**



实验题目 考勤管理系统

学生姓名 金明瀚

学生学号 201806062407

学生班级 软工1802班

任课教师 廖峰锋

提交日期

**计算机科学与技术学院**

**学生考勤管理系统 实验报告**

1. **大型实验内容**

学生考勤管理系统用于对学生缺课情况的记录，要求完成的主要功能包括录入学生缺课情况，包括：缺课日期、第几节课、课程名称、学生姓名、缺课类型（迟到、早退、请假及旷课）。功能要求：

1、录入学生的缺课记录；

2、修改某个学生的缺课记录；

3、查询某个学生的缺课情况；

4、统计某段时间内，旷课学生姓名及旷课次数，按旷课次数由多到少排序；

5、统计某段时间内，有学生旷课的课程及旷课人次，按旷课人次由多到少排序；

6、系统以菜单方式工作；

7、所有的增加、修改、删除能同步到文件；也从文件读取数据到程序。

要求通过学习过的C/C++程序设计知识完成学生考勤管理系统的设计与实现。

1. **运行环境**

学生考勤管理系统在dev-C++平台下开发，操作系统：Windows 10。

硬件环境：

处理器：Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30GHz

内存：8.00GB

系统类型：64位操作系统

1. 实验课题分析

**3.1学生考勤管理系统**

学生考勤管理系统主要功能为：学生缺课信息管理、可以实现缺课信息的录入、缺课信息的查找、时段内学生缺课次数统计、时段内被旷课程统计。详细结构图如下：

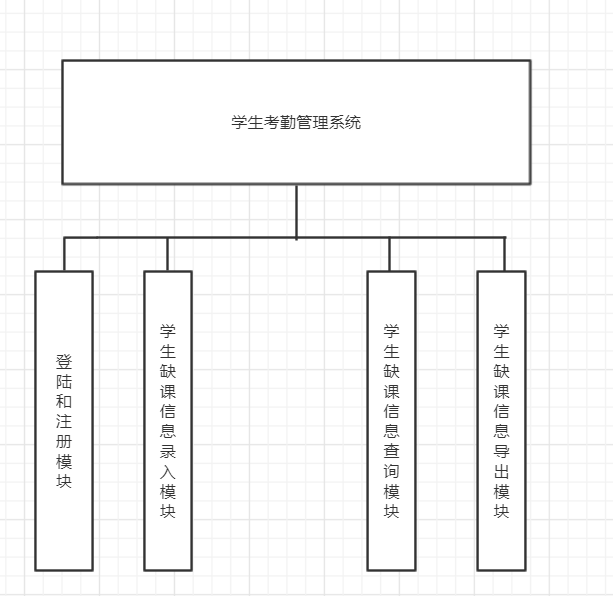


图1 系统结构图

系统各模块的功能具体描述为：

1. 登录模块

先输入操纵系统的总密码（管理员密码），选择登陆已有教师账号，或注册新的教师账号，成功后进入系统。从文件读入已有账号信息如果输入的账号不存在、账号对应的密码不对，管理员密码错误，立刻结束系统运行。创建的账号添加入文件中。退出时保存所有已经确认的操作。

1. 学生缺课信息录入模块

可以对缺课信息进行操作例如录入、修改、增加、删除

录入、增加：按需要先输入每次录入的缺课信息的条数，按提示输入相应的信息。

修改、删除：对于某一条信息出错的，可以进行修改，或删除，输入需要修改信息的学生信息的id、时间和第几节课这三个信息对该学生缺课记录进行精确定位，并重新录入信息修改，对于需要删除的信息，直接将值设为0。意为删除，并在后续的操作中逐步覆盖。

1. 学生缺课信息查询模块

可以通过输入具体的初末时间段，和学生的学号来查询这个学生所有符合条件的缺课记录。

1. 学生缺课情况汇总查询模块

可以通过输入某个时间区间，实现对该时间内所有学生的旷课情况统计汇总，并按次数降序输出的功能。也可以按课程名称查询。输入时间，输出课程被旷次数，降序输出。

**3.2系统分析及设计**

系统涉及对象有一个类：学生类。涉及的功能操作归纳为如下表1所示:

表1 学生类涉及的操作

|  |  |
| --- | --- |
| **对象** | **涉及的对象操作** |
| 老师 | 录入/增加学生缺课信息 |
| 查找学生缺课信息 |
| 查看课程被旷情况 |
| 修改学生缺课信息 |

另外还有基于结构体safesystem的密码系统主要功能有：1、管理员登陆 2、教师注册/登陆。

设计学生类实现系统要求的功能。

用txt文本对数据进行保存，主要需要保存的数据包括：学生数据、账号数据等信息。

|  |
| --- |
| Student类 |
|  |
|  |

|  |
| --- |
| 结构体students |
| 学生统计信息 |
| 课程统计信息 |

|  |
| --- |
| 结构体safesystem |
| 教师账号信息 |
|  |

图二 系统主要类结构和结构体

**3.3系统的实现**

**（1）类的编写**

系统project名为考勤管理系统。包含了student类，在类中包含了对于符合条件的对象的数据，例如：名字、学号、时间等等信息。同时还有对应的函数调用这些数据，对应的函数实现相应的功能。

具体的类结构声明如下

**Student类**

|  |
| --- |
| class student  {  public:  void get\_name()  {  cout<<"姓名：" ;  cin>>name;  }  void get\_id()  {  cout<<"\n学号：（浙工大计算机专业学号）";  cin>>id;  }  void get\_time()  {  cout<<"\n缺课时间：(例如1999 11 11)";  cin>>year>>month>>day;  }  void get\_order()  {  cout<<"\n课时：" ;  cin>>order;  }  void get\_classname()  {  cout<<"\n课程名称：";  cin>>classname;  }  void get\_abstype()  {  cout<<"\n缺课类型：";  cin>>abstype;  }  string check\_name()  {  return name;  }  long long check\_id()  {  return id;  }  int check\_year()  {  return year;  }  int check\_month()  {  return month;  }  int check\_day()  {  return day;  }  int check\_order()  {  return order;  }  string check\_classname()  {  return classname;  }  string check\_abstype()  {  return abstype;  }  void chuanru\_name(string n)  {  name = n;  }  void chuanru\_id(long long i)  {  id = i;  }  void chuanru\_date(int y,int m,int d)  {  year = y;  month = m;  day = d;  }  void chuanru\_order(int orde)  {  order = orde;  }  void chuanru\_classname(string cla)  {  classname = cla;  }  void chuanru\_abstype(string abs)  {  abstype = abs;  }  void get\_info();  void modify();  void search(long long n);  int check(long long id,int year,int month,int day,int order);  private:  string name;//姓名  long long id;//学号  int year,month,day; //缺课日期  int order; //第几节课  string classname; //课程名称  string abstype;//缺课类型（迟到、早退、请假、旷课）    }; }  void chuanru\_abstype(string abs)  {  abstype = abs;  }  void get\_info();//录入  void modify();//修改  void search(long long n);//搜索id  int check(long long id,int date,int order);//确认输入信息  private:  string name;//姓名  long long id;//学号  int date; //缺课日期  int order; //第几节课  string classname; //课程名称  string abstype;//缺课类型（迟到、早退、请假、旷课）  }; |

**Students结构体：**

|  |
| --- |
| struct students  {  string classnames;//被旷的课程  string namee;//旷课学生name  int countname;//旷课次数统计  int countclass;//该课程被旷次数统计  }stu[10000]; |

**结构体safesystem：**

|  |
| --- |
| struct safesystem  {  string zhanghao;  string mima;  }tea[2000]; |

**补充说明：**

在类中实现的各种功能例如学生缺信息的查找、修改、添加和删除都直接以遍历类数组的形式实现。通过计算机强大的算力将变得非常快捷简便。

通过此程序实现的各个功能如下：

1. **管理员登录功能：**

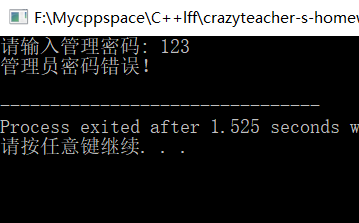
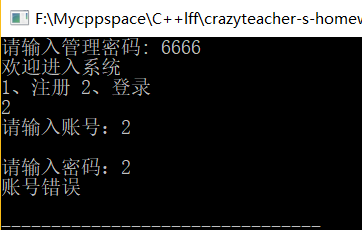
在打开程序时，需要输出管理员密码才可以进入，设计这个功能的目的是为了尽可能的保障操作的安全，尽管出发点在与安全，但是这样的账号密码系统还可以拓展很多功能由于时间关系并未过多拓展，具体如下：

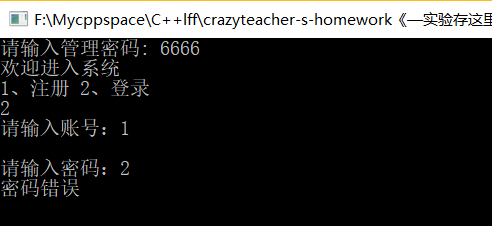
* 1. 首先输入管理员密码进入登陆界面可以选择注册、登陆等功能。
  2. 不管是管理员密码错误还是账号密码错误都直接结束应用进程。
  3. 注册的新的账号都存到对用的文件中保存，程序打开时导入到程序中。

详细代码请见源文件

具体见图三：

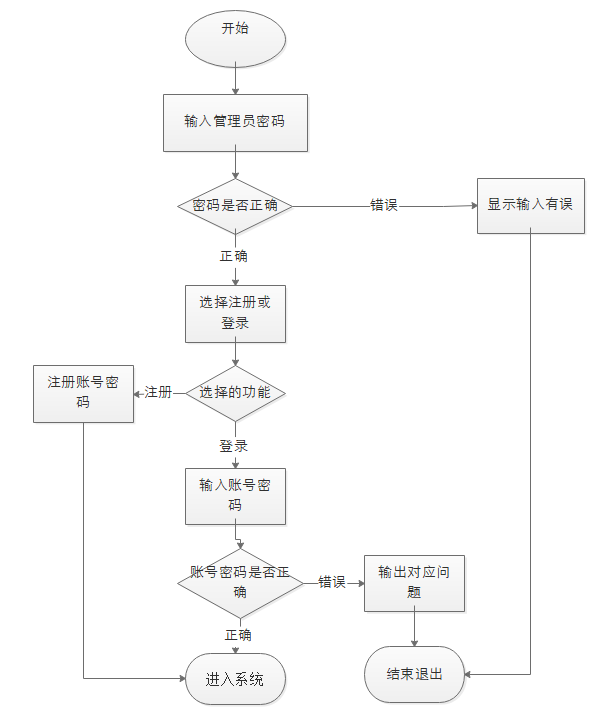
图三 登录成功





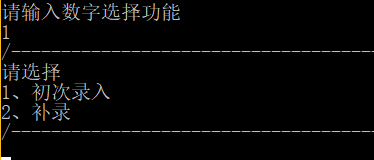
图三 登录失败

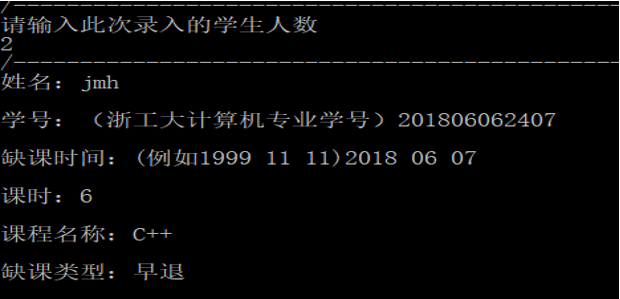
流程图如下



登录功能流程图

1. **学生缺课信息的录入：**

在录入的时候有两种选择，首先是首次录入，就直接在没有任何基础数据的情况下输入数据。其次是补录功能，是在已经有数据的基础上在添加数据。两个不同的选项均为先选择要录入的缺课情况条数，依次输入后各条缺课记录。同时，在输入结束的时候，管理员可以选择是否保存到硬盘文件中。详情请见图4以及图5：



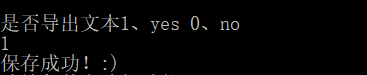


图4 录入功能图

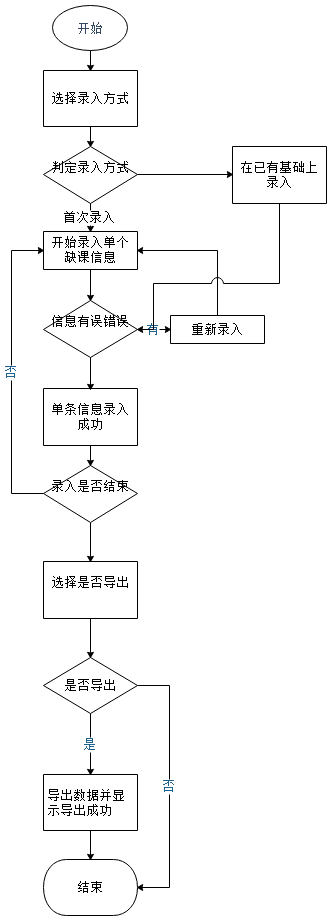


图5 录入功能流程图

**缺课信息的修改或删除功能：**

对于已有的录入错误的信息，或是文件中已有的录入信息进行修改，具体修改方式为先输入学号、时间、课时三个信息对缺课记录进行精确查找，倘若缺课记录不存在，则输出缺课信息不存在的提示，若找到对应缺课信息，则提示用户修改数据，在这些操作之前，会从文件中导入存在的缺课信息。查找结束后，用户可以自行选择是否要保存修改。确定保存后将数据重新写入文件，提示用户保存成功，结束操作。详情见图5、6：

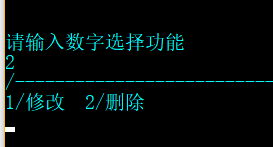


图6 运行示意图

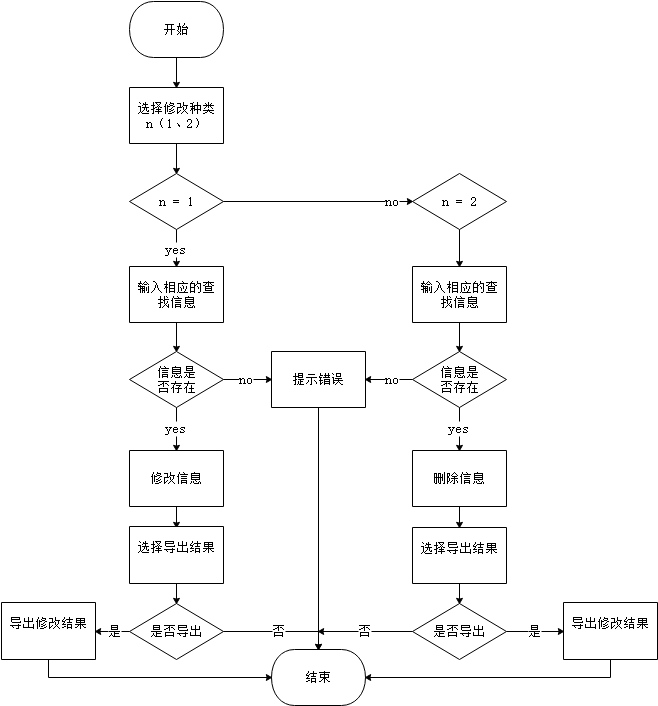


图7修改删除功能图

**查询学生缺课信息功能:**

在这个功能中可以输入学生的学号对学生的缺课信息进行搜索并输出，首先将文件中的缺课信息导入到创建的类对象中，再在类对象里进行遍历搜索，符合的对象随机输出全部信息，不符合的对象直跳过。

详情见图8，9：

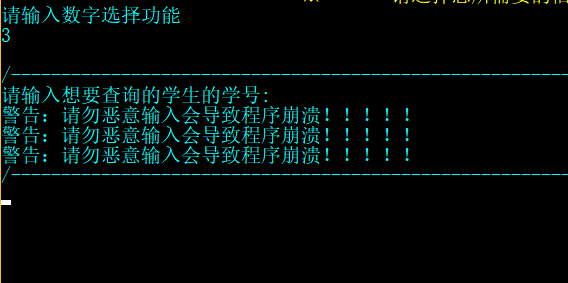


图8 功能运行图

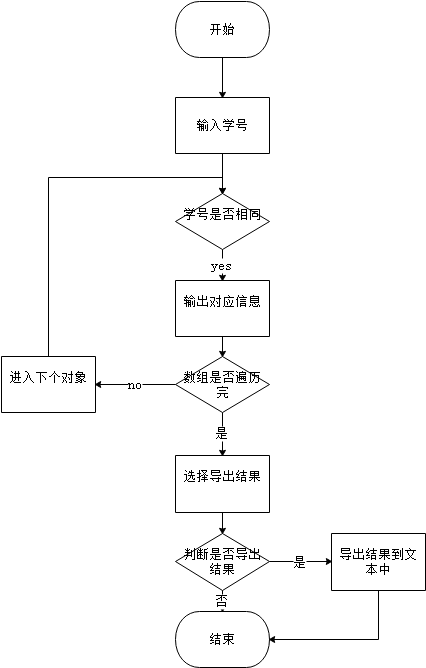


图9 学生缺课缺课搜索流程图

**缺课学生情况总统计功能（姓名与次数）：**

在选择该功能后，首先从文件中导入所有的缺课记录，再又键盘输入起始时间和终止时间，若时间输入有误，将有提示时间输入有问题，并由键盘重新输入。在利用结构体students里的namee变量记录旷课学生姓名，countname变量记录旷课次数，遍历数组后，所有的时间范围内的缺课记录都将别记录到对应建立的结构体数组中去，并由用户决定是否由程序导出到文件中去。

详情如图10、11所示：

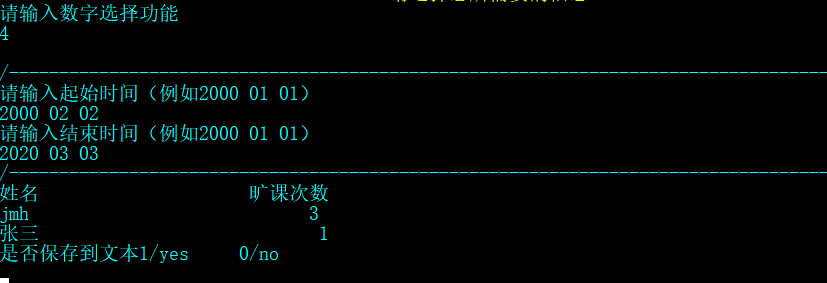


图10 功能运行图

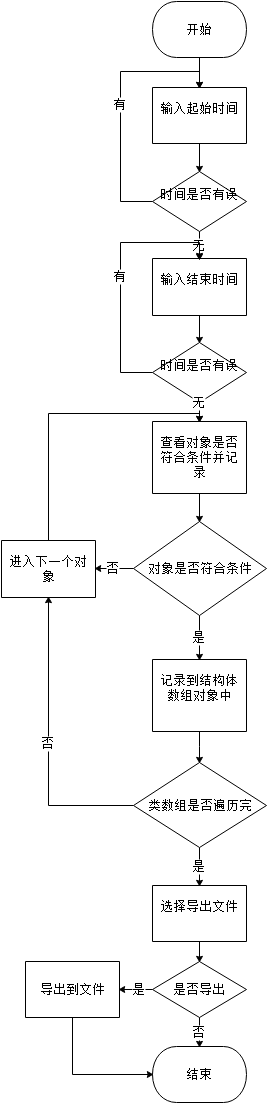


图11 学生缺课汇总功能流程图

**课程被旷情况统计功能：**

类似于对学生缺课情况汇总功能，对一段时间范围内的课程被旷情况进行统计，首先从文件中导入所有的缺课记录，再又键盘输入起始时间和终止时间，若时间输入有误，将有提示时间输入有问题，并由键盘重新输入。在利用结构体students里的classname变量记录被旷课课程名字，countclass变量记录被旷次数，遍历数组后，所有的时间范围内的缺课记录都将别记录到对应建立的结构体数组中去，并由用户决定是否由程序导出到文件中去。

详情如图12、13所示：

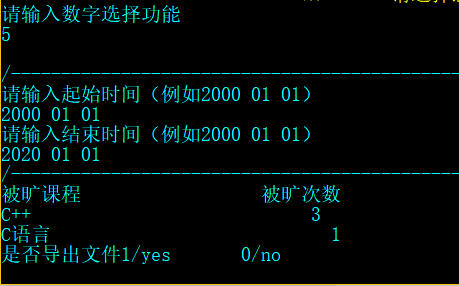


图12 功能运行图

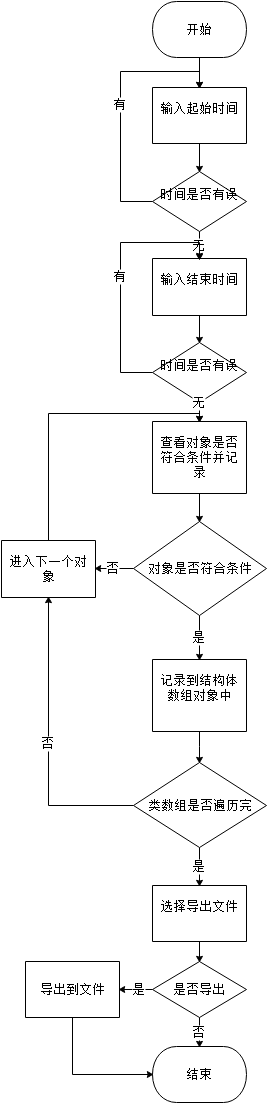


图13 课程被旷统计流程图

**交互界面以及登录菜单的实现：**

系统运行开始的界面如图14所示：

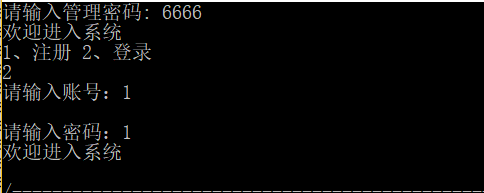
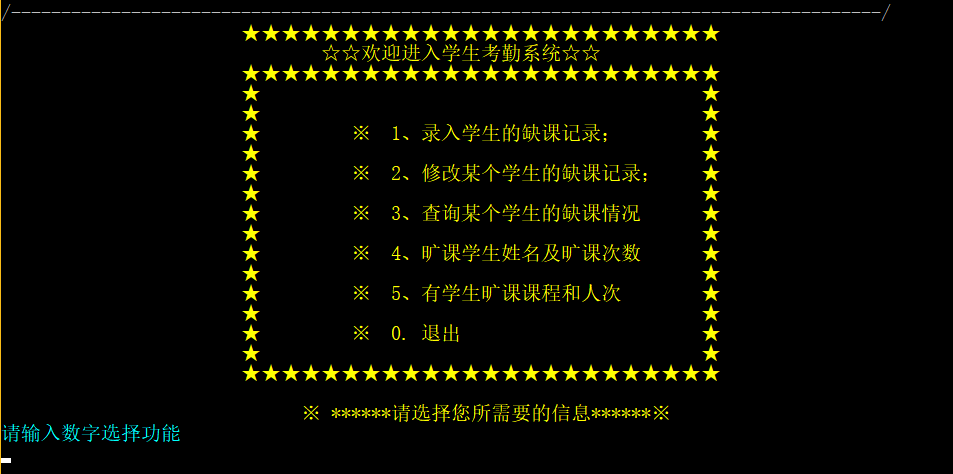


图14 开始的登录界面

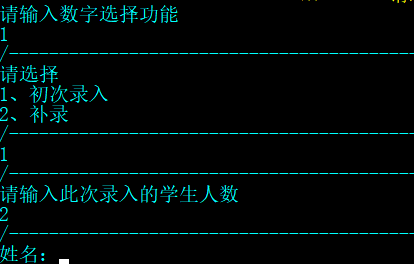
主要通过选择结构和循环结构实现界面的前进和回退。例如在登录成功后，会出现如下界面：



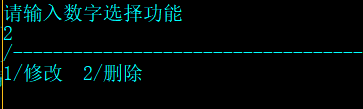
为了显得交互界面不那么冗杂，每一个功能使用结束后，都使用清屏函数对之前操作进行清除。尝试着用改变颜色美化界面。

1. **实验调试、测试、运行记录及分析**

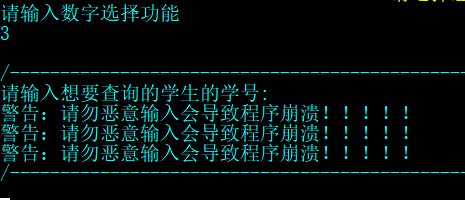
功能1：



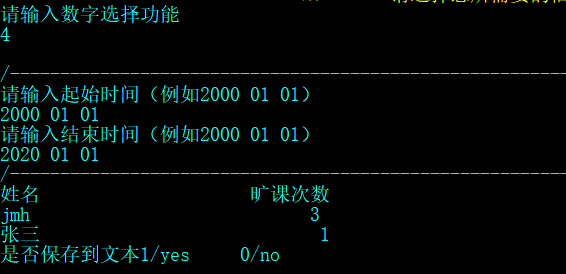
功能2：



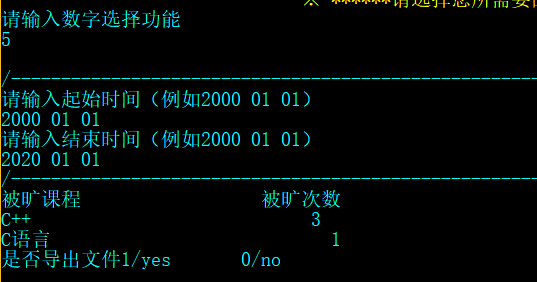
功能3：



功能4：



功能5：



总的来讲，用户可以按照交互界面的提示就可进行操作，不存在不知道如何输入的问题。

**遇到的问题以及解决方法：**

**问题一：**

**问题描述：**

在输入学号时，如果输入数字过长，会导致程序崩溃，但是当时已经改成了long long 数据类型，理论上18位以下的数字不会出现问题。

**解决方法：**

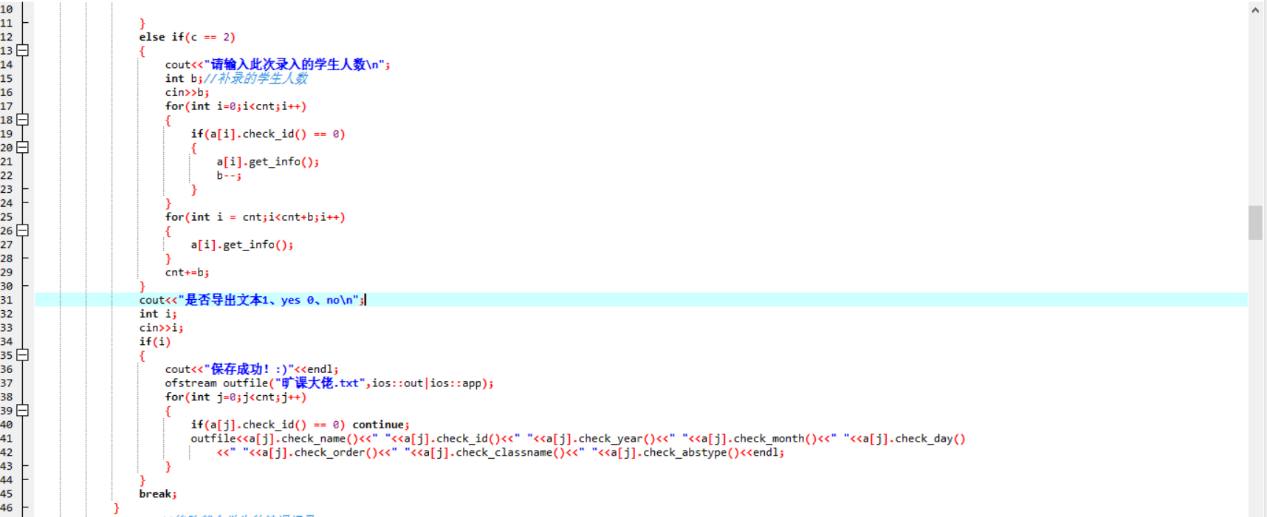
经过对程序的不断复查，发现，虽然自己的学号的数据类型已经是 long long 类型，但是在一个有关学号的类函数的返回值的类型仍为int 型，导致最终传入的学号被限制成了int型，学号长度超过12位时就会导致程序崩溃进入死循环。

**问题二：**

**问题描述:**

在录入功能中的补录，如果之前有过初次输入，就会出现在导出时，缺课记录重复的问题。

原代码如下图：



**解决方法：**

将导入功能分别放入两个录入方式中，若为补录，则从上一个标记点开始导出到文件中。

1. **实验总结（优点、不足、收获及体会）**

我设计的考勤管理系统基于任务书的要求，类的结构设计清晰，功能完善，在储存数据的方式上，基于现实需求选择了用类对象数组储存，操作上难度较小，支持增删改查等等功能，很好的满足的实验设计的需求。

存在的缺点是对于密码系统和实际功能的结合上不是很完美，实际上对应不同的管理员账号可以标记录入的缺课信息的操作人，也可以加入对缺课信息操作的权限等功能，让密码系统更加有意义与实际可用性，但是由于时间问题，以及实践操作上的难度较大，并没有付诸实践，这也是这次实验的遗憾之处，在设计之初并没有考虑周当，对需求没有很深的认识，导致后期对代码反复修改，拖长的工作时间。再一个就是再对储存方式的使用上，简单的使用的类数组存储信息，在查询或是搜索的时候，一旦数据量变大，效率不高，可以使用二分或是排序后搜索等算法提高效率，也希望日后能够学到更多的有用的算法。

这次C++大型实验的设计，让我体验到了实践的重要性，也让我体验到了自己设计程序的快乐，对于专业产生了更加浓厚的兴趣，同是在实验过程中也让我意识到了自己在编程上的一些不足，就比如对边界情况的考虑欠妥，会导致程序出现各种奇怪的错误，对于各种变量的命名和对作用域的把控不是很好，也为我在接下来的学习指明了方向。

1. **附录：源代码**