|  |  |
| --- | --- |
| **学 号：** | 0121503490118 |

****

**课 程 设 计**

|  |  |
| --- | --- |
| **题 目** | QQ群重复成员查找 |
| **学 院** | 管理学院 |
| **专 业** | 信息管理与信息系统 |
| **班 级** | 1501 |
| **姓 名** | 李令新 |
| **指导教师** | **燕翔** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2017 | 年 | 07 | 月 | 03 | 日 |

**课程设计任务书**

**学生姓名： 李令新 专业班级： 信管1501**

**指导教师： 燕翔**  **工作单位： 管理学院**

**题 目：QQ群重复成员查找**

**初始条件：**

QQ群是聚集相似人群非常有用的工具。然而群容量是有限的，当一个群人数满了以后，群主往往需要开设一个新群来容纳更多的人。当一个群主拥有了至少两个或以上的相同主题的QQ群，并且人数达到一定规模的时候，问题就会显现出来。

当一个普通群员发现群主还有另外一个相同主题的群，总是会产生进去探索一番的心理，于是就会出现多个QQ群里有很多重复成员的情形。有些情况下，重复成员是必要的，比如管理员、特邀嘉宾等；但更多时候，群主可能并不希望同一个人用相同的QQ号同时占据多个群的名额，因此他需要把这些“不小心”放进去的人找出来，然后……。

找出这些重复人员并不是一件容易的事情，尤其是当两个群的人数都很多时。现在你的任务就是帮助苦恼的群主解决这个难题。

必须提醒的是，导出的文件并没有使用自动换行，但是从**错误!未找到引用源。**中可以看出，某些成员的QQ号被强制分成两行显示了。你得首先弄清楚这是怎么回事，并找到解决对策。特别提醒，该文件的内容不允许被更改！亦即你的程序必须要有办法找到第一个群成员开始的地方并正确辨别需要的信息，答辩时将由老师使用自备数据检验。

**要求完成的主要任务:** （包括课程设计工作量及其技术要求、说明书撰写等具体要求）

1. 找出的重复成员名单输出到一个文本文件中保存，文件名自定。
2. 如何定位第一个群成员可以参考串的模式匹配算法，要求必须使用KMP算法。
3. 对于任意一个群，都能按照根据群名片和QQ号两种方式从小到大排序输出全体成员，且结果保存在文本文件中便于查看。

**时间安排：**

|  |  |  |
| --- | --- | --- |
| **序号** | **设计内容** | **所用时间** |
| 1 | 问题分析和任务定义 | 0.5天 |
| 2 | 数据类型和系统设计 | 0.5天 |
| 3 | 编码实现和静态检查 | 3天 |
| 4 | 上机准备和上机调试 | 2天 |
| 5 | 总结和整理设计报告 | 1天 |
| 合 计 | | 7天 |



**C:\Users\muz\Documents\Tencent Files\1195843739\Image\C2C\QF0AXPRP4L)XZR]RI1[8ZDB.jpg指导教师签名： 2017年 06月07日**

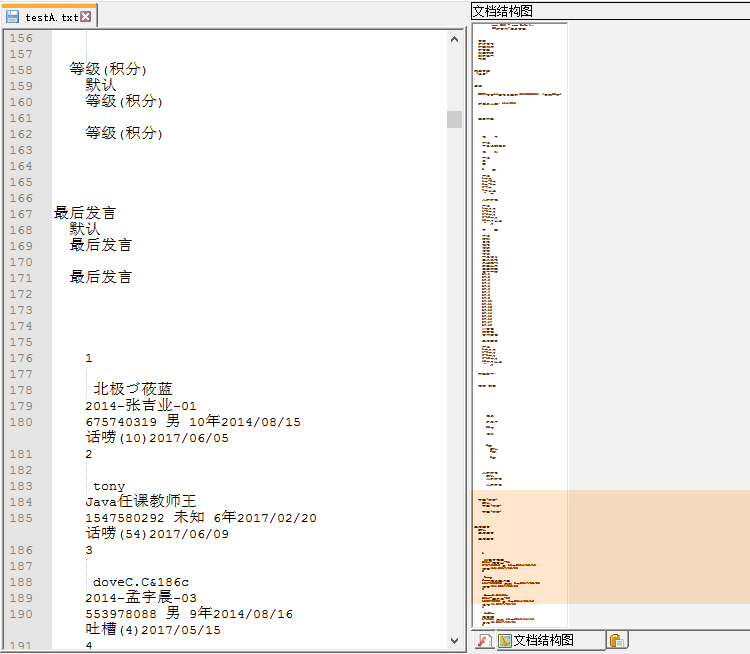
**系主任（或责任教师）签名： 2017年 06 月 07日**

QQ群重复成员查找

# 需求分析

* 1. 输入的形式和输入值的范围

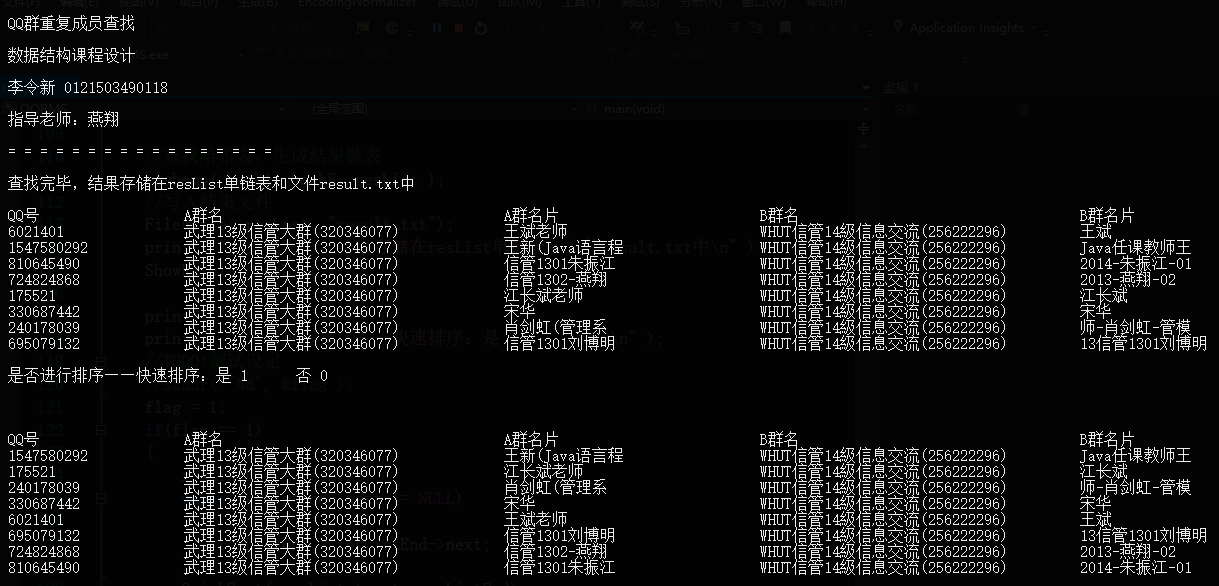
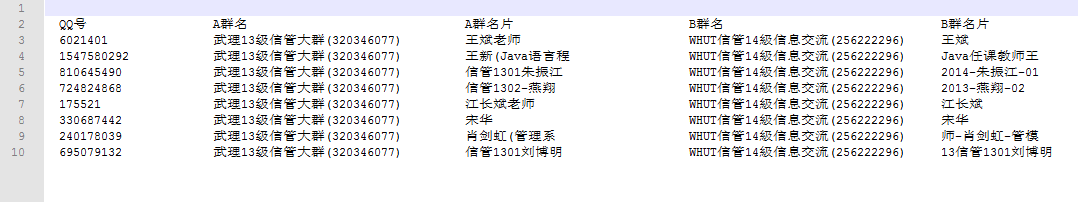
1. 原始数据的获取（参考课设指导书）
2. 打开浏览器，输入<http://qun.qq.com/>，点击“登录”，进入群空间。
3. 在群空间页面上点击下图所示的“群管理”图标，再点击“成员管理”，选择想查看的群，例如“武汉理工大学信管师生”群，进入群成员清单界面。
4. 由于网页元素的特殊性，需要按住Ctrl+End键不放，一直到所有成员都显示出来，然后在当前界面下按Ctrl+S键，弹出保存网页窗口，更改保存类型为文本文件，保存之。

这样，我们就得到了一个群的成员名单文件。里面的内容大体上如下：

1. 数据形式及范围
2. 文件编码格式

文件应以windows中文简体版默认格式编码ANSI（中文简体GBK）

* 1. 输出形式

1. 控制台输出
2. 文本输出
   1. 程序功能
3. 读入QQ群成员名单，并提取每个成员的群内名片，QQ号
4. 将每个成员数据项按照哈希表组织
5. 检查两个哈希表重复项，并把结果存入结果单链表
6. 对结果单链表执行快速排序
7. 输出结果
   1. 测试数据

项目文件夹内

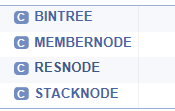
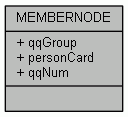
正确格式数据：testA.txt testB.txt testC.txt

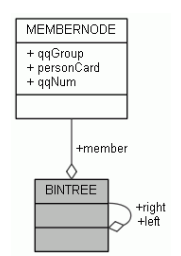
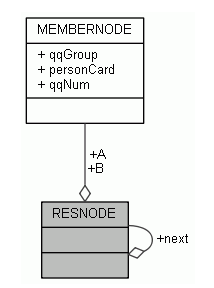
正确结果输出：“无重复”或结果表格

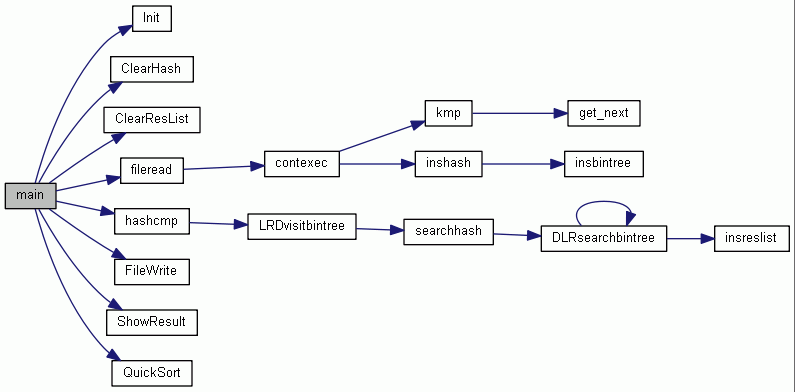
错误格式数据：testERR.txt（文件编码格式错误）

错误结果提示：指出错误地方，并停止执行

# 概要设计

2.1 抽象数据类型



2.2 主函数流程及函数调用关系

# 详细设计

3.1数据类型

1. MEMBERNODE

数据对象：

typedef struct BINTREE

{

struct MEMBERNODE \* member;//QQ成员信息结构体

struct BINTREE \* left; //小值在左

struct BINTREE \* right; //大值在右

}binTreeNODE;

基本操作：创建。

1. BINTREE

数据对象：

typedef struct BINTREE

{

struct MEMBERNODE \* member;//QQ成员信息结构体

struct BINTREE \* left; //小值在左

struct BINTREE \* right; //大值在右

}binTreeNODE;

基本操作：插入结点insBinTree、后序遍历LRDVisitBinTree、先序遍历DLRVisitBinTree，快速排序QuickSort

1. RESNODE

数据对象：

typedef struct BINTREE

{

struct MEMBERNODE \* member;//QQ成员信息结构体

struct BINTREE \* left; //小值在左

struct BINTREE \* right; //大值在右

}binTreeNODE;

基本操作：追加结点insResList、遍历结点ShowResult

* 1. 函数声明

void Init ()

int fileread (char \*filename, binTreeNODE \*hash[])

int contexec (char \*str, binTreeNODE \*hashT[])

int isChinese (char \*)

long kmp (char \*master, char \*pattern, long pos)

int searchhash (binTreeNODE \*bintree, binTreeNODE \*hashB[], resNODE \*resList)

int insreslist (resNODE \*list, resNODE \*p)

int insbintree (binTreeNODE \*, binTreeNODE \*)

int hashcmp (binTreeNODE \*hashA[], binTreeNODE \*hashB[], resNODE \*resList)

int LRDvisitbintree (binTreeNODE \*bintree, binTreeNODE \*hashB[], resNODE \*resList)

int DLRsearchbintree (binTreeNODE \*bintree, binTreeNODE \*item, resNODE \*resList)

int FileWrite (resNODE \*list, char \*filename)

int QuickSort (resNODE \*list, resNODE \*end)

int PushStack (STACK \*stack, STACK \*item)

STACK \* PopStack (STACK \*stack)

void ShowResult (resNODE \*list)

int ClearResList (resNODE \*list)

int ClearHash (binTreeNODE \*hash[])

int LRDClearBinTree (binTreeNODE \*bintree)

int inshash (binTreeNODE \*hashT[], binTreeNODE \*item)

# 调试分析

4.1 问题及解决方案

1. 文件编码格式问题

问题：开始调试时，使用群内老师提供的数据，通过编辑器查看编码均显示为UTF-8，但是处理时，却出现了错误。

解决：最终经过老师指导，重新生成测试数据用例，通过windows默认编码格式，最终避免了编码格式错误的问题。

同时，增加文件检测功能，若编码格式错误，则直接输出提示信息，暂停执行，并返回上一层。

1. Kmp匹配越界问题

问题：在匹配不到的时候，主串指针继续移动造成越界异常中断。

解决：增加指针位置判断功能。若大于主串尾位置，则停止匹配，返回上一层。

1. 单链表快速排序的实现

问题：快排的核心在于分割。数组的快排通常是首尾两端同时移动，但是单向链表的操作指针仅能向一个方向移动。

解决：

1. 每次轴心点元素，均取头指针数据。首先考虑使用栈来记录每次的分割位点。前半程，即分割点没有与头指针相遇时，分割位点逐步逼近轴心点（头指针数据），分割位点压入栈。相遇时，进入下半程，轴心点元素取分割位点，同时分割位点由出栈元素担任。栈空时即表明全部排序完成。由于经常需要操作指针元素，实现复杂，又由于指针的不安全性，故最终没有使用该方案。
2. 通过递归的方式进行，每次分割开的前后两部分，均当做子链表进行快排。当首尾指针想等，或为空时，返回。此种方法实现简便，最终以此方法实现。
3. 内存的释放

问题：每执行完一次均需要对上次操作结果进行清空，应释放内存。但是由于未知的内存溢出或是野指针问题，导致释放失败。由此引发了一系列运行不稳定问题。

解决：暂未解决。

# 4.2 算法

1. 哈希存储

[binTreeNODE](file:///D:\Data-structure\%EF%BC%B1%EF%BC%B1%E7%BE%A4%E9%87%8D%E5%A4%8D%E6%88%90%E5%91%98%E6%9F%A5%E6%89%BE\docs\doxygen\html\struct_b_i_n_t_r_e_e.html) \* hashA[[HASHLENGTH](file:///D:\Data-structure\%EF%BC%B1%EF%BC%B1%E7%BE%A4%E9%87%8D%E5%A4%8D%E6%88%90%E5%91%98%E6%9F%A5%E6%89%BE\docs\doxygen\html\rms_8c.html#a663a695f0855a54eae517d4b68e704a0)] = { NULL };//构造一个哈希表，表长为HASHLENGTH=500；

哈希冲突通过拉链法解决。拉链形为二叉查找树。

复杂度分析：一次插入或查找操作，由一次哈希和二叉查找树构成。

* 在由N个随机键构造的二叉查找树中，查找命中平均所需的比较次数为2ln(N)约等于1.39lg(N)。
* 在由N个随机数构造的二叉查找树中，插入操作和查找未命中平均所需的比较次数为2ln(N)约等于1.39lg(N)。
* 在一棵二叉查找树中，所有操作在最坏情况下所需时间和树的高度成正比。

**注：引用自网络博客**

1. 二叉查找树插入，先序后序遍历
2. 单链表插入，遍历

复杂度：O(n)

1. Kmp模式匹配

复杂度：O(m+n)

1. 单链表快速排序

复杂度：O(nlogn)

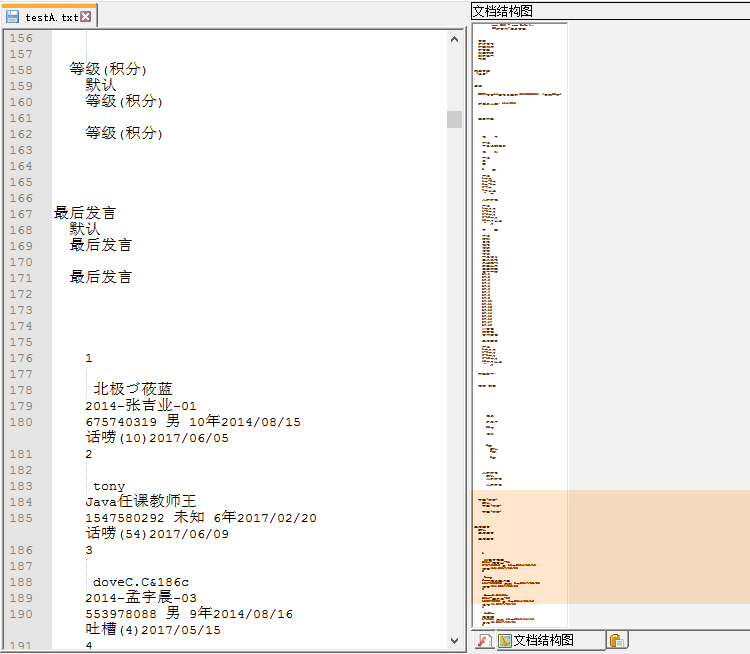
# 用户使用说明

5.1 启动程序

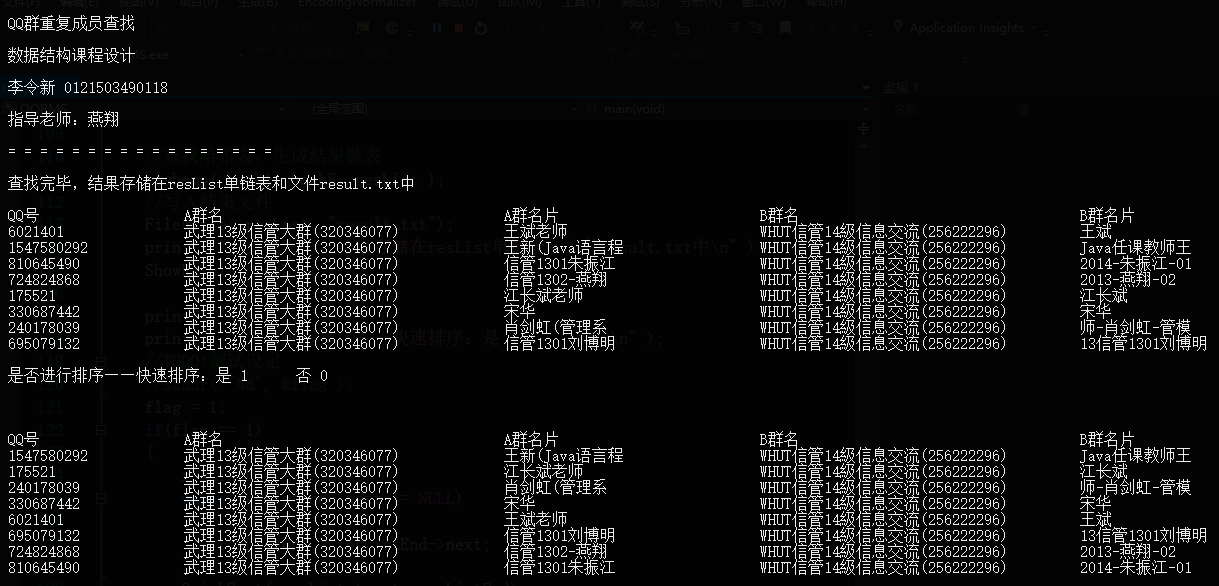
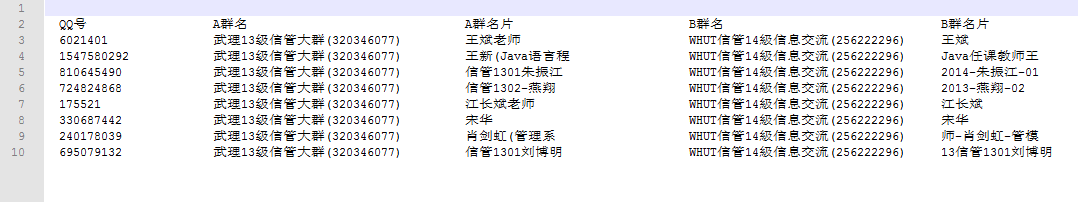
5.2 输入待处理数据文件名（建议将文本与程序文件放在同一路径下

5.3 根据屏幕提示，输入指令。

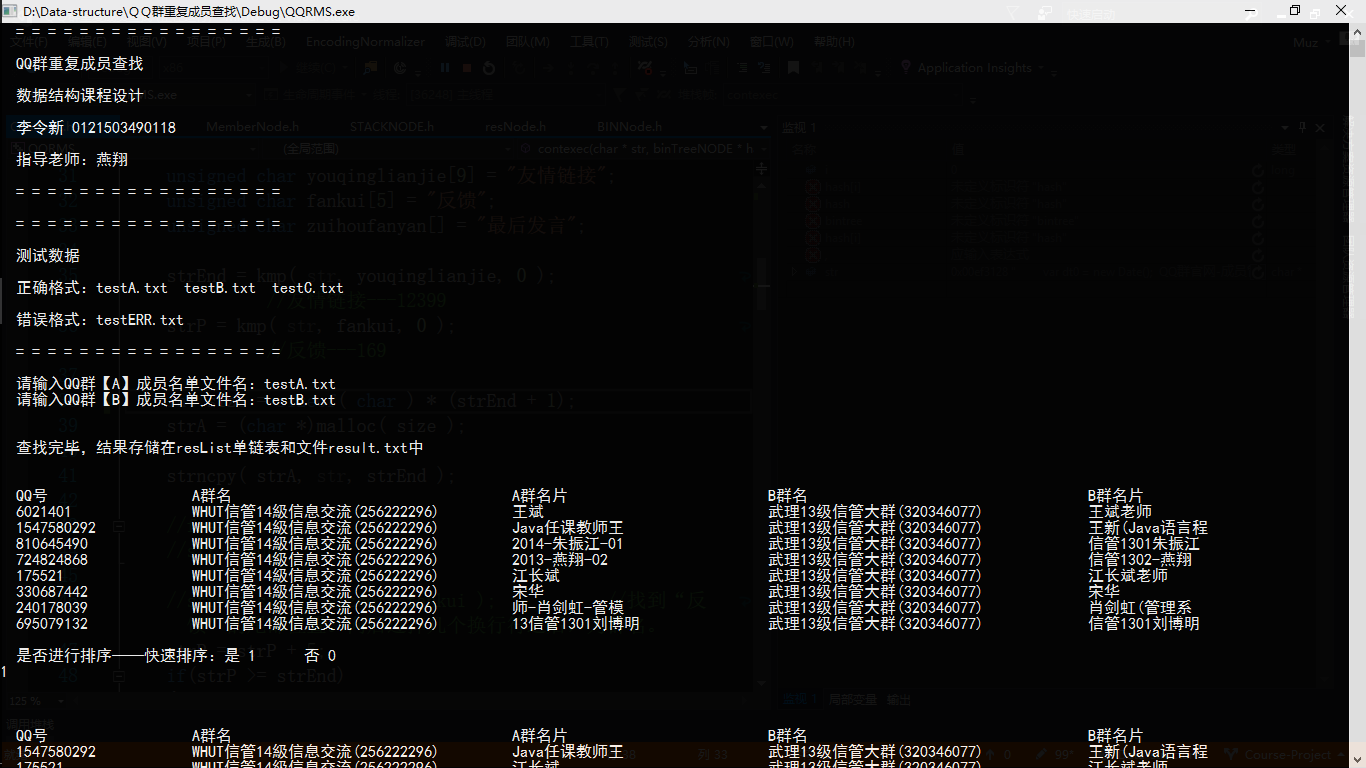
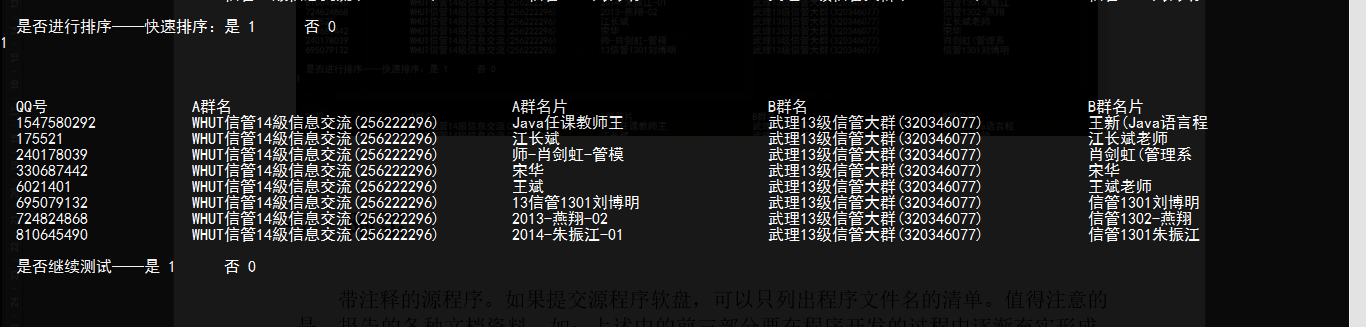
# 测试结果（截图）

6.1 输入

6.2 输出

1. 控制台输出
2. 文本输出

6.3 操作截图



# 附录

**本科生课程设计成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名** | 李令新 | **性 别** | **男** |
| **专业班级** | **信息管理与信息系统1501** | | |
| **课程设计题目：QQ群重复成员查找** | | | |
| **课程设计答辩或质疑记录：** | | | |
| **成绩评定依据：**   |  |  | | --- | --- | | **评 定 项 目** | **评分成绩** | | 1．设计原创性达到80%（30分） |  | | 2．程序执行结果实现指导书中的所有基本要求（20分） |  | | 3．设计结果（软件程序）正确（10分） |  | | 4．设计报告的规范化（10分） |  | | 5. 答辩（30分） |  | | 总分（100分） |  |   备注：成绩等级：优（90分—100分）、良（80分—89分）、中（70分—79分）、及格（60分—69分）、60分以下为不及格。若第一项低于20分，或第一项与第五项之和低于35分，则课程设计成绩不能及格。 | | | |
| **最终评定成绩（以优、良、中、及格、不及格评定）** | | | |



**指导教师签字：**

**2017年 07月 03日**