**丰收祭前的游戏**

**(初赛第二阶段)**

**情景描述：**

在某片遥远的大陆上，居住着两个世代友好的部落，分别是部落A和部落B。他们一起耕耘劳作，互相帮助，亲如一家。久而久之，部落里的每个人都在对方部落里找到了志趣相投，互相欣赏的好朋友。有的人性格热情开朗，好朋友很多；有的人性格沉稳内敛，好朋友相对少一些。

每到秋天丰收的季节，这两个部落的人民都会聚集在一起举行盛大的“丰收祭”，来祈祷下一年的风调雨顺。今年的丰收祭马上又要举行了。为了进一步增进两个部落的友谊，也为了明年能有一个好收成，这两个部落的祭司们商量后决定：在今年的丰收祭前举办一场特别的“击鼓传花”游戏。只不过游戏中并非有人真的击鼓，并且所传递的“花”也不是真的花，而是等待在丰收祭上献上的祭品。

游戏规则如下：

1. 两个部落的所有人都可以事先准备自己的祭品，且每个人的祭品样式都不同，每一个祭品都分别盛放在一个相对应的木托盘里；准备此祭品的人熟悉自己的祭品；
2. 每个人可以准备的祭品数量不限；祭品的最小不可分割单位是1份；
3. 游戏开始后，在整个游戏过程中，每个人都能且只能将祭品（包括木托盘）传递给自己在对方部落里的好朋友们，每个好友可以接收的祭品数量不限；
4. 收到祭品的人必须在盛放此祭品的木托盘上刻上自己的名字（代表留下自己美好的祝愿），随后按照上一条规则，继续传递；
5. 如果祭品回到最初准备此祭品的人手中，此人也在木托盘上刻上自己的名字之后，终止传递；
6. 木托盘上不允许出现重复的人名，如果无法满足此条件，则不再继续传递该祭品；
7. 当所有的祭品都不再传递后，游戏结束；

游戏开展得非常顺利。游戏结束后，祭司们将收集同时满足如下三个条件的祭品用于接下来的丰收祭活动：

1. 此祭品回到了最初准备它的人手中；
2. 盛放此祭品的木托盘上至少有4个名字，至多有14个名字；
3. 如果有多个木托盘上的名字完全一样（不区分名字的排列顺序），则从其中随机选择一个木托盘所对应的祭品。

**问题：**

已知这两个部落里的所有人都不重名，并且部落A的人和部落B的人之间的好朋友关系以附件的csv数据表格文件给出，其中行索引代表部落A中的人，列索引代表部落B中的人，表格中的数字“1”代表他们两人是好朋友，“0”代表他们两人不是好朋友。请问：

如果以木托盘上的名字的数量对用于丰收祭的祭品分类，每一类分别最多有多少个祭品？

**请参赛者答题：**

木托盘上有4个名字的祭品最多有（）个；木托盘上有6个名字的祭品最多有（）个；木托盘上有8个名字的祭品最多有（）个；木托盘上有10个名字的祭品最多有（）个；木托盘上有12个名字的祭品最多有（）个；木托盘上有14个名字的祭品最多有（）个；

请在每个（）中填写一个正整数答案；

网上初赛将分为两个阶段，其中第一阶段将于赛题发布的同时启动。第二阶段于4月30日启动，参赛者需参加全部两阶段的比赛。两阶段的结果准确性得分及排名互相独立。即第二阶段启动后，第一阶段的的得分及排名将不再更新。赛事方将根据参赛者在两阶段比赛中提交的结果的准确性得分和参赛者所设计的算法的效率给出参赛者的综合成绩。

**参赛者需注意以下事项：**

* 参赛者以个人为单位参赛，最后结果只关联到一个参赛自然人。
* 每天第一次登陆时参赛者会随机得到一个代表两个部落的人民之间好朋友关系的csv数据表格文件。参赛者需要在得到此表格文件后的四十分钟内提交答案，在此时间内参赛者最多可提交答案2次。超过该指定时间，则需等待至第二天再重新答题，重新答题后，参赛者将获得一份新的csv数据表格文件，并基于新得到的csv数据表格文件给出答案。
* 参赛者可使用C/C++/Python/Matlab 计算机编程语言，在自己的计算机上计算出结果，再把计算结果输入到系统中。
* 参赛者还需提供可在64位windows系统的PC个人电脑上使用CPU单进程和单线程执行的算法程序源代码文件和算法原理说明文档。使用C/C++的参赛者，还应提供编译后的exe可执行文件。参赛者需将源代码文件、编译后的exe可执行文件（如有）和算法原理说明文档打包压缩为zip文件后上传系统，压缩包文件的命名规则为“拼音姓名\_学校英文缩写\_版本号.zip”，其中版本号信息请用“v+阿拉伯数字”表示，版本号从v0开始，压缩包文件每更新一次，版本号信息中的阿拉伯数字增加1，例如ZhangSan\_NJU\_v1.zip，版本号信息不可省略。
* 参赛者提交的源代码应当可直接读取符合初赛两个阶段赛题形式的本地csv数据表格文件，并将计算结果保存在以“**result.txt**”命名的文本文件中。源代码应符合常见的代码书写规范，源代码文件中应有必要的注释信息，具备良好的可读性。输出结果的内容应清晰可读。赛事方要求参赛者的源代码仅可在注释部分使用中文字符，代码的其余部分请不要使用中文字符。例如：代码中请勿出现打开或保存含有中文字符的文件名及路径，或打印中文字符至文件及屏幕等操作。算法原理说明文档应使用中文文字清晰地说明所设计算法的思想。参赛者提交的说明文档不要和历史提交的文档关联，即提交的说明文档应是对算法原理的完整描述，而不是增量描述。算法原理的说明应不少于300字，说明文档请使用PDF或MS Word文件格式。
* 参赛者在系统中分别填写不同种类的祭品数量。系统将根据参赛者所提交的各项数值结果给出参赛者的当次得分和历史最高得分。该得分仅代表参赛者在规定时间内所提交的数值结果的准确性。系统理论上能给出的最高得分为100分；
* 初赛第二阶段开始后，参赛者需及时提交含有源代码，可执行exe文件（如有）和算法原理说明文档的压缩包文件；系统将对已提交上述文件的参赛者的本阶段历史最高得分排名，该排名数据公开；未提交上述文件的参赛者将不被计入初赛第二阶段的排行榜中；若参赛者在初赛第二阶段结束前补交了上述文件，则系统将根据参赛者第二阶段的历史最高得分（含补交前的得分）将参赛者计入排行榜；
* 赛事方将对参赛者最终提交的代码进行准确性和效率测试。代码准确性的优先级高于代码效率的优先级。在准确性相同的情况下，赛事方将根据算法效率的排序确定参赛者的最终名次；
* 参赛者提供的代码必须能够在以下测试环境中正常运行：

（1）操作系统：64位MS Windows；

（2）内存：不大于8G内存；

（3）CPU主频：不高于3GHz；

（4）进程和线程数：单进程和单线程；

（5）C/C++版本：c99/c++11；集成开发环境 visual studio 2015；参赛者需打包上传使用的外部依赖库/函数并说明使用方法（如有）。或者，

Python软件版本：Python 3.7，参赛者需说明使用的外部依赖库/函数（如有）。或者，

Matlab软件版本：Matlab 2019a，参赛者需打包上传使用的外部依赖库/函数并说明使用方法（如有）。

（6）其他说明：若参赛者使用Java语言 ，请使用JDK1.8+IntelliJ 2019 community,且参赛者需要打包上传使用的外部依赖库并说明使用方法（如有）；但赛事方不承诺Java代码测试的优先级。

* 在一般家用计算机配置条件下，使用CPU单线程，计算结果应当能在十分钟之内得到。如果参赛者的计算耗时明显高于此时间，则应考虑为所设计算法程序的效率问题，请对算法程序进行优化。留意观察数据的特征，参赛者也许可以找到提升效率的思路。