CS 101 – Problem Solving & Programming 1
Fall 2012
Program 7: The Six Degrees of George Clooney
Algorithm due Thursday, Oct 25
Program due Sunday, Oct 28

If Alice lives next door to Bob, and Bob went to college with Charlie, and Charlie works with Eva, and Eva is related to Frank, we say there are four *degrees of separation* between Alice and Frank; that is, we can get from Alice to Frank in 4 steps. Note that Alice and Frank may never have heard of each other; the separation between them is a function of how connections can be traced, not whether people have actually met.

This leads to some interesting properties. For example, on a fully-loaded passenger plane full of randomly-chosen Americans, where none of the passengers know each other, there is greater than a 75% probability that two people on the plane have a mutual acquaintance. The popular *six degrees* hypothesis claims that it's possible to connect any two people in the world in no more than 6 steps. It's been shown this is probably not true, mostly because of some very isolated rural populations. But if you restrict the sample to urban dwellers, it appears much more likely to be true.

Unsurprisingly, some people have more connections than others; these people can serve as bridges between otherwise unconnected networks. Perhaps the best-known example is the party game *Six Degrees of Kevin Bacon*: Starting with an arbitrary actor, draw a connection between that actor and Kevin Bacon in no more than six steps. (Or, for a more challenging game, *exactly* six steps.) For this program, we're going to solve a similar problem, using George Clooney.

Program Specification: You are given a text file containing a list of movies, with actors appearing in that movie. The name of the movie is listed first, followed by the actors, separated by commas. Each movie is on a separate line, and each movie has a unique title. No movie title contains a comma, so you can separate the line into distinct items by splitting on the comma.

Your program will ask the user for the file name, read the input file, and produce 2 output files. The first lists all actors in the file, sorted by name, with the *Clooney number* (defined below) for each. The second file lists the actors sorted by Clooney number (and sorted by name within each list). The attached output files show the format to use. Name the output files 'actors.txt' and 'numbers.txt'. (It is not necessarily to match the formatting *exactly*; don't bother with counting individual spaces, etc. But the formatting for the file should show some forethought, and the output should have a neat, well-organized appearance.)

Definition of Clooney Number: The Clooney number is the smallest integer showing the number of connections between an actor and George Clooney. George Clooney, and only George Clooney, has a Clooney Number of 0. Any actor appearing in a movie with George Clooney has a Clooney Number of 1. Any actor who has not appeared with George Clooney, but has appeared with an actor who has a Clooney Number of 1, has a Clooney Number of 2. In general, if an actor has appeared with an actor with a Clooney number of k, and has not appeared with anyone with a smaller Clooney Number, then that actor has a Clooney Number of $k+1$. If the actor has not appeared with any actor with a Clooney Number, then the Clooney Number does not exist for that actor.

Development notes:

- Think about how you are going to store the data. It may be helpful to store it in more than one

way, perhaps indexed by Clooney Number, or by actor. You may want to store data about actors and movies separately.

- This assignment is going to make heavy use of dictionaries and sets. Remember that the key to a dictionary can be any immutable type, which includes strings and integers. Values can be anything. For example, you could have a dictionary where the keys are movie titles and the values are lists or sets of the actors appearing in that movie. Or a dictionary where the keys are integers (Clooney numbers) and the values are the set of actors having that number.
- If an actor has a Clooney number of *n*, then any actors appearing with that actor that don't already have Clooney numbers will have Clooney numbers of *n* + 1. Can this be used to iterate through, for example, all movies or all actors?
- Once an actor has been assigned a Clooney number, there is no possibility of being assigned a higher number. You may want to consider starting with a data structure holding all actors who haven't been assigned Clooney numbers; as they are processed, remove them from that structure and put them into another. This continues until all actors have been assigned Clooney numbers, or no more numbers can be assigned. (Any actors remaining don't have a Clooney number.)
- If you had a dictionary where the keys were Clooney numbers and the values were lists or sets of actors, you could iterate through it and build up a list of (actor, number) tuples, which could then be sorted by actor name.
- Some actors don't have a Clooney number. Rather than dealing with handling the special case of not having a Clooney number, it may be more convenient to assign a 'dummy' value that can't be a real Clooney number—for example, -1. For output, you should still use “None” or tell the user it doesn't exist, but that doesn't mean you have to store it that way inside your program.
- No, this input file is not comprehensive. That doesn't matter for this assignment. Likewise, don't worry about inconsistent spellings of actors' names. If the file has 'Sean Connery' in one movie and 'Bean Connery' in another, your program can treat them as different actors.

Deliverable: Your python source file. You do not need to upload the input or output files. Your program will be graded using a different input file of the same format.

Sample run: Console output is minimal for this program; the primary output of this program is the 2 text files. Here's the console output from running my implementation in WING 101:

```
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)]
Type "help", "copyright", "credits" or "license" for more information.
[evaluate Program07.py]
Name of movie data file? movies.txt
Numbers file written.
Actors file written.
All done. Have an ordinary day.
```