COMP3311 19s1 Database Systems

Assignment 2
PHP, SQL, PLpgSQL

[Assignment Spec] [Schema]

Downloads: a2.tgz, a2.zip
Note that `a2.tgz` and `a2.zip` contain the same material.
Each archive contains the IMDB database dump `a2.db`, a sample PHP code file, `a2.php` plus `updates.sql`.

Aims

This assignment aims to give you practice in

- further use of SQL and PLpgSQL
- writing scripts in PHP that interact with a database

The goal is to complete the functionality of some command-line tools via a combination of database code and PHP code.

Summary

Submission: Login to Course Web Site > Assignments > Assignment 2 > Assignment 2 Specification > Make Submission > upload required files > [Submit]

Required Files: a2.php, updates.sql, acting, title, toprank, similar, shortest, degrees

Deadline: Friday 19 April 2018 @ 23.59

Late Penalty: Late submissions will have marks deducted from the maximum achievable mark at the rate of 0.5% of the total mark per hour that they are late (i.e., 12% per day).

This assignment contributes **15 marks** toward your total mark for this course.
The mark for each question indicates (roughly) its level of difficulty.
The total marks for the questions sum to 15.

How to do this assignment:

- read this specification carefully and completely
- create a directory for this assignment
- unpack the supplied zip file into this directory
- login to `grieg` and run your PostgreSQL server
- remove your Assignment 1 database (to save space)
- create a new database called `a2` for your assignment 2
- load the schema and populate the data into the database by `psql -f a2.db a2`
- perform any required updates to the database by `psql -f updates.sql a2`
- re-acquaint yourself with the database (by browsing the data using psql and/or examine its schema through the link `[Schema]` above)
- familiarise (read the code) yourself with the provided sample PHP code file called `pg`
- make sure `pg` (by `chmod u+x pg`) and run it
- complete the tasks using `pg` as a reference example, you may also want to edit the supplied template files `a2.php` and `updates.sql`
- submit all these files (the Required Files) via WebCMS3 as described above

Details of the above steps are given below. Note that you can put the files wherever you like; they do not have to be under your `/srvr` directory. You can also edit the PHP and SQL files on hosts other than `grieg`. The only time that you need to use `grieg` is to manipulate your database or to run your PHP scripts.

You will probably not be able to fit both the database for Assignment 1 and the database for Assignment 2 into your PostgreSQL server on `grieg` (because of your quota on `/srvr`).

Introduction

A successful movie (or TV show) not only entertains audience, but also enables film companies to gain tremendous profit. A lot of factors (such as good directors, experienced actors) are important for creating good movies. Nevertheless, famous directors and actors usually bring an attractive box-office income, but they do not necessarily guarantee a highly rated imdb score. This assignment is based on an IMDB dataset to build several small PHP commands to show interesting results.

The dataset itself contains around 5000 movies, spanning across 100 years in 66 countries. There are more than 2000 movie directors, and thousands of actors/actresses. It also contains the IMDB rating score, numbers of votes and various facebook likes. To let you feel the kind of data that you are dealing with, a (unordered) glimpse of the dataset is included below:

```
a2=> select * from movie limit 10;
  id  |           title            | year | content_rating | duration |    lang    |       country        |  gross   |  budget  | director_id
------+----------------------------+------+----------------+----------+------------+----------------------+----------+----------+-------------
  407 | 102 Dalmatians             | 2000 | G              |      100 | English    | USA                  | 66941559 | 85000000 |        2174
 3699 | 10 Cloverfield Lane        | 2016 | PG-13          |      104 | English    | USA                  | 71897215 | 15000000 |        1327
 3016 | 10 Days in a Madhouse      | 2015 | R              |      111 | English    | USA                  |    14616 | 12000000 |        1881
 2846 | 10 Things I Hate About You | 1999 | PG-13          |       97 | English    | USA                  | 38176108 | 16000000 |        1458
 3421 | 10th & Wolf                | 2006 | R              |      107 | English    | USA                  |    53481 |  8000000 |        2252
 3645 | 11:14                      | 2003 | R              |       85 | English    | USA                  |          |  6000000 |         665
 2430 | 127 Hours                  | 2010 | R              |       94 | English    | USA                  | 18329466 | 18000000 |          57
 4823 | 12 Angry Men               | 1957 | Not Rated      |       96 | English    | USA                  |          |   350000 |        1172
 1737 | 12 Monkeys                 |      | TV-14          |       42 | English    | USA                  |          |          |
 2282 | 12 Rounds                  | 2009 | PG-13          |      108 | English    | USA                  | 12232937 | 22000000 |        1479
(10 rows)

a2=> select * from rating limit 10;
 movie_id | num_critic_for_reviews | num_user_for_reviews | num_voted_users | movie_facebook_likes | cast_total_facebook_likes | imdb_score
----------+------------------------+----------------------+-----------------+----------------------+---------------------------+------------
      407 |                     84 |                   77 |           26413 |                  372 |                      4182 |        4.8
     3699 |                    411 |                  440 |          126893 |                33000 |                     14504 |        7.3
     3016 |                      1 |                   10 |             314 |                26000 |                      2059 |        7.5
     2846 |                    133 |                  549 |          222099 |                10000 |                     37907 |        7.2
     3421 |                     26 |                   34 |            5557 |                  294 |                      2512 |        6.4
     3645 |                     68 |                  133 |           38273 |                    0 |                      2200 |        7.2
     2430 |                    450 |                  440 |          279179 |                63000 |                     11984 |        7.6
     4823 |                    177 |                  888 |          447785 |                40000 |                      1433 |        8.9
     1737 |                     13 |                   56 |           20839 |                12000 |                      3309 |        7.6
     2282 |                    113 |                  113 |           22823 |                    0 |                      2799 |        5.6
(10 rows)

a2=> select * from director limit 10;
  id  |       name       | facebook_likes
------+------------------+----------------
 1139 | Brad Bird        |            663
  741 | Don Hall         |             38
 1295 | Rich Moore       |             66
 2132 | Dean DeBlois     |            255
 1228 | Jonathan Mostow  |             84
  516 | James Gunn       |            571
  203 | Hideaki Anno     |             28
 2312 | David Fincher    |          21000
 1753 | Matthew Vaughn   |            905
 1317 | Francis Lawrence |            508
(10 rows)

a2=> select * from actor limit 10;
  id  |         name         | facebook_likes
------+----------------------+----------------
  408 | Christian Bale       |          23000
 1430 | Donna Murphy         |            553
```

```
   66 | Robert Downey Jr.    |          21000
  502 | Daniel Radcliffe     |          11000
 2011 | Lauren Cohan         |           4000
  945 | Marlon Brando        |          10000
 1245 | Ruth Wilson          |           2000
 1590 | Christopher Meloni   |           3000
 3674 | Pierfrancesco Favino |            216
 2270 | Adam Brown           |            972
(10 rows)

a2=> select * from acting limit 10;
 movie_id | actor_id
----------+----------
      407 |     2024
     3699 |     1841
     3016 |       11
     2846 |      195
     3421 |      738
     3645 |     1186
     2430 |      211
     4823 |     1299
     1737 |      786
     2282 |      866
(10 rows)

a2=> select * from genre limit 10;
 movie_id |   genre
----------+-----------
      407 | Adventure
      407 | Comedy
      407 | Family
     3699 | Drama
     3699 | Horror
     3699 | Mystery
     3699 | Sci-Fi
     3699 | Thriller
     3016 | Drama
     2846 | Comedy
(10 rows)

a2=> select * from keyword limit 10;
 movie_id |    keyword
----------+----------------
      407 | dog
      407 | parole
      407 | parole officer
      407 | prison
      407 | puppy
     3699 | alien
     3699 | bunker
     3699 | car crash
     3699 | kidnapping
     3699 | minimal cast
(10 rows)
```

The sample PHP program

By following the steps above, you can prepare the setup of your assignment 2 and experience the provided sample program as follows:

```
-bash-4.1$ dropdb a2
-bash-4.1$ createdb a2
-bash-4.1$ psql -f a2.db a2
SET
SET
...
ALTER TABLE
ALTER TABLE

-bash-4.1$ psql -f updates.sql a2
-bash-4.1$ ./pg
Usage: ./pg Year

-bash-4.1$ ./pg 1988
1. Beetlejuice (PG, English, 92)
2. Big (PG, English, 130)
3. Crocodile Dundee II (PG, English, 108)
4. My Stepmother Is an Alien (PG-13, English, 105)
5. Poltergeist III (PG-13, English, 98)
6. Scrooged (PG-13, English, 101)
7. Twins (PG, English, 107)

-bash-4.1$
-bash-4.1$ ./pg 1989
1. Back to the Future Part II (PG, English, 108)
2. Batman (PG-13, English, 126)
3. Bill & Ted's Excellent Adventure (PG, English, 90)
4. Dead Poets Society (PG, English, 128)
5. Driving Miss Daisy (PG, English, 99)
6. Henry V (PG-13, English, 137)
7. Indiana Jones and the Last Crusade (PG-13, English, 127)
8. Licence to Kill (PG-13, English, 133)
9. New York Stories (PG, English, 124)
10. Star Trek V: The Final Frontier (PG, English, 107)
11. The Abyss (PG-13, English, 171)
12. Troop Beverly Hills (PG, English, 105)
13. UHF (PG-13, English, 150)
14. We're No Angels (PG-13, English, 106)
```

Submission and Testing

We will test your submission on `Grieg` as follows:

- create a testing subdirectory
- untar the `a2.tgz` file into that directory
- load your submitted `a2.php` and `updates.sql` files over the top of the standard ones
- create a new database *a2* and initialise it with `a2.db` (For testing purposes, the dataset may be modified and slightly different from the dataset in your a2.db)
- run the command: `psql a2 -f updates.sql` (using your `updates.sql`)

- run a series of tests using your submitted `acting`, `title`, `toprank`, `similar`, `shortest` and `degrees` scripts
- manually inspect your submitted PHP code and SQL code

Your submitted code must be *complete* so that when we do the above, your PHP will work just as it did in your assignment directory and with a database with the identical schema (with either the same or slightly modified dataset) to yours (`a2.db`). If your code does not work when installed for testing on `Grieg`, as described above (for example, your `updates.sql` did not contain all of the required definitions, or you did not submit all of the required PHP files, etc), you will be penalised by a 50% administrative penalty.

Some things that you can do wrong:

- make changes to files *other than* `a2.php`, `updates.sql` and the corresponding PHP commandline files (such as `toprank`)
- forget to include all of your PHP functions in `a2.php` or other PHP files
- forget to include all of your SQL and PLpgSQL definitions in `updates.sql`

Before you submit, you should test out whether the files you submit will work by following a similar sequence of steps to those noted above on `Grieg`.

Tasks

For each task, you are required to implement an executable PHP program command that takes in commandline arguments and displays the result in a specified format (which we will use for auto-marking). We assume the sample output format for each task below to to test your programs. In particular, as shown in the code of `pg`, if any field of your output is empty, you should hide that field and its related formatting text (for example, in `pg`, a comma and a space will not be printed for `duration` if it is empty). Of course, you are free to print debugging information temporarily (e.g. to add debugging `print_r()` calls), but please remember to eventually disable or remove them when your solution is submitted for marking. For all the tasks below, unless it is specified explicitly, we assume movies including any titles stored in the movie table (i.e., including movies and TV shows etc).

In implementing your functions, you are free to partition the functionality however you like between the database and the PHP scripts. In the past, some students have solved similar assignments to this by writing just about everything in SQL views and PLpgSQL functions and using PHP simply as a vehicle for collecting the results. Others have done most of the computational work in PHP. Do whatever you feel most comfortable with, but make sure that you limit your changes just to the files that you are required to submit, and do not create additional tables.

For each task below, output nothing if there is no result returned from the database. If the task does not specify particular output order, your program can output in any row order. If multiple columns are involved in your output, your output should follow the same column order as the sample output presented in each task below (for example, the order of the fields inside the brackets of the `pg` output matters).

Task A: The list of movies acted by a given actor (1 marks)

The `acting` script lists the movie title, its director, its year, its content rating and IMDB score, one per line, of all the movies acted by the given actor. The output is sorted by year and then by movie title, both in ascending order. If any movies have empty year, output them at the end and sort them by movie title in ascending order. It accepts one command-line argument: an actor name (has to exactly match a name stored in the database and the matching is case insensitive). It has the following output format:

```
grieg$ ./acting "james franco"
1. Whatever It Takes -- David Raynr (2000, PG-13, 5.5)
2. City by the Sea -- Michael Caton-Jones (2002, R, 6.2)
3. Deuces Wild -- Scott Kalvert (2002, R, 5.6)
4. Spider-Man -- Sam Raimi (2002, PG-13, 7.3)
5. Spider-Man 2 -- Sam Raimi (2004, PG-13, 7.3)
6. The Great Raid -- John Dahl (2005, R, 6.7)
7. Flyboys -- Tony Bill (2006, PG-13, 6.6)
8. The Dead Girl -- Karen Moncrieff (2006, R, 6.7)
9. In the Valley of Elah -- Paul Haggis (2007, R, 7.2)
10. Spider-Man 3 -- Sam Raimi (2007, PG-13, 6.2)
11. Milk -- Gus Van Sant (2008, R, 7.6)
12. Pineapple Express -- David Gordon Green (2008, R, 7.0)
13. 127 Hours -- Danny Boyle (2010, R, 7.6)
14. Date Night -- Shawn Levy (2010, PG-13, 6.3)
15. Eat Pray Love -- Ryan Murphy (2010, PG-13, 5.7)
16. Rise of the Planet of the Apes -- Rupert Wyatt (2011, PG-13, 7.6)
17. Your Highness -- David Gordon Green (2011, R, 5.6)
18. Spring Breakers -- Harmony Korine (2012, R, 5.3)
19. The Iceman -- Ariel Vromen (2012, R, 6.9)
20. Homefront -- Gary Fleder (2013, R, 6.5)
21. Oz the Great and Powerful -- Sam Raimi (2013, PG, 6.4)
22. Palo Alto -- Gia Coppola (2013, R, 6.3)
23. This Is the End -- Evan Goldberg (2013, R, 6.7)
24. The Interview -- Evan Goldberg (2014, R, 6.6)
25. The Little Prince -- Mark Osborne (2015, PG, 7.8)
26. Sausage Party -- Greg Tiernan (2016, R, 7.5)
```

For all the tasks in this assignment, as mentioned previously, your program shall output nothing if there are no results. If there are N results, it shall output exactly N lines of output without any extra lines or messages. We can then use the line count to measure the size of an output. For example:

```
grieg$ ./acting "james franco"|wc -l
26
grieg$ ./acting "john smith" |wc -l
0
```

Task B: List movie information by its title substring (1 marks)

The `title` script lists the movie title, year, content rating, IMDB score and genres of those movies with the title matching the given substring (case insensitive), one per line. The output of multiple genres of a movie should be concatenated in one line delimited by a comma and sorted alphabetically in ascending order, as shown by an example below. The rows are ordered by year (ascending), then by IMDB rating (descending) and finally by title (ascending). It has the following output format:

```
grieg$ ./title "star war"
1. Star Wars: Episode IV - A New Hope (1977, PG, 8.7) [Action,Adventure,Fantasy,Sci-Fi]
2. Star Wars: Episode V - The Empire Strikes Back (1980, PG, 8.8) [Action,Adventure,Fantasy,Sci-Fi]
3. Star Wars: Episode VI - Return of the Jedi (1983, PG, 8.4) [Action,Adventure,Fantasy,Sci-Fi]
4. Star Wars: Episode I - The Phantom Menace (1999, PG, 6.5) [Action,Adventure,Fantasy,Sci-Fi]
5. Star Wars: Episode II - Attack of the Clones (2002, PG, 6.7) [Action,Adventure,Fantasy,Sci-Fi]
6. Star Wars: Episode III - Revenge of the Sith (2005, PG-13, 7.6) [Action,Adventure,Fantasy,Sci-Fi]
7. Star Wars: The Clone Wars (TV-PG, 7.9) [Action,Adventure,Animation,Drama,Fantasy,Sci-Fi]

grieg$ ./title "happy"
1. Happy Gilmore (1996, PG-13, 7.0) [Comedy,Sport]
2. Happy, Texas (1999, PG-13, 6.3) [Comedy,Crime,Romance]
3. The Pursuit of Happyness (2006, PG-13, 8.0) [Biography,Drama]
4. Happy Feet (2006, PG, 6.5) [Animation,Comedy,Family,Music,Romance]
5. Another Happy Day (2011, R, 6.0) [Comedy,Drama]
6. Happy Feet 2 (2011, PG, 5.9) [Animation,Comedy,Family,Musical]
7. Happy Christmas (2014, R, 5.6) [Comedy,Drama]
8. Happy Valley (TV-MA, 8.5) [Crime,Drama]

grieg$ ./title "mars"
1. Invaders from Mars (1986, PG, 5.5) [Horror,Sci-Fi]
2. Mars Attacks! (1996, PG-13, 6.3) [Action,Comedy,Sci-Fi]
3. Mission to Mars (2000, PG, 5.6) [Adventure,Sci-Fi,Thriller]
4. Ghosts of Mars (2001, R, 4.9) [Action,Horror,Sci-Fi]
5. We Are Marshall (2006, PG, 7.1) [Drama,Sport]
6. Forgetting Sarah Marshall (2008, R, 7.2) [Comedy,Drama,Romance]
7. The 41-Year-Old Virgin Who Knocked Up Sarah Marshall and Felt Superbad About It (2010, R, 2.7) [Comedy]
8. Mars Needs Moms (2011, PG, 5.4) [Action,Adventure,Animation,Comedy,Family,Sci-Fi]
9. The Last Days on Mars (2013, R, 5.5) [Horror,Sci-Fi,Thriller]
```

```
10. Veronica Mars (TV-14, 8.4) [Crime,Drama,Mystery]
```

Task C: Top ranked movies (2 marks)

The `toprank` script takes in 3 or 4 commandline arguments:

```
./toprank K StartYear EndYear
```

or:

```
./toprank Genres K StartYear EndYear
```

where Genres is a list of genres separated by '&', K is the top K movies ranked by IMDB score and then by the number of votes (both in descending order) between (and including) StartYear and EndYear, with 1 <= K <= 1000, 1900 < StartYear <= EndYear < 2020 and your program will not be tested with a list of more than 8 genres. We interpret '&' as conjunction, i.e., the selected movies shall contain all the specified genres. When Genres is not provided (when your program takes in 3 arguments), perform the same ranking but on movies with any genres. Do not include any movie titles with empty year. For example:

```
./toprank "Action&Sci-Fi&Adventure" 10 2005 2005
1. Serenity (2005, PG-13, English) [8.0, 242599]
2. Star Wars: Episode III - Revenge of the Sith (2005, PG-13, English) [7.6, 520104]
3. The Island (2005, PG-13, English) [6.9, 263329]
4. Zathura: A Space Adventure (2005, PG, English) [6.1, 67707]
5. Doom (2005, R, English) [5.2, 88146]
6. Stealth (2005, PG-13, English) [5.0, 45455]
7. A Sound of Thunder (2005, PG-13, English) [4.2, 16474]
8. The Helix... Loaded (2005, R, English) [1.9, 534]

./toprank "Sci-Fi&Adventure&Action" 20 1920 2019
1. Inception (2010, PG-13, English) [8.8, 1468200]
2. Star Wars: Episode V - The Empire Strikes Back (1980, PG, English) [8.8, 837759]
3. Star Wars: Episode IV - A New Hope (1977, PG, English) [8.7, 911097]
4. Star Wars: Episode VI - Return of the Jedi (1983, PG, English) [8.4, 681857]
5. Aliens (1986, R, English) [8.4, 488537]
6. Captain America: Civil War (2016, PG-13, English) [8.2, 272670]
7. Godzilla Resurgence (2016, Japanese) [8.2, 374]
8. The Avengers (2012, PG-13, English) [8.1, 995415]
9. Guardians of the Galaxy (2014, PG-13, English) [8.1, 682155]
10. Mad Max: Fury Road (2015, R, English) [8.1, 552503]
11. Deadpool (2016, R, English) [8.1, 479047]
12. Destiny (2014, English) [8.1, 3089]
13. X-Men: Days of Future Past (2014, PG-13, English) [8.0, 514125]
14. Star Trek (2009, PG-13, English) [8.0, 504419]
15. Serenity (2005, PG-13, English) [8.0, 242599]
16. The Iron Giant (1999, PG, English) [8.0, 128455]
17. Avatar (2009, PG-13, English) [7.9, 886204]
18. Iron Man (2008, PG-13, English) [7.9, 696338]
19. Edge of Tomorrow (2014, PG-13, English) [7.9, 431620]
20. Big Hero 6 (2014, PG, English) [7.9, 279093]

./toprank 20 1920 2019
1. The Shawshank Redemption (1994, R, English) [9.3, 1689764]
2. The Godfather (1972, R, English) [9.2, 1155770]
3. The Dark Knight (2008, PG-13, English) [9.0, 1676169]
4. The Godfather: Part II (1974, R, English) [9.0, 790926]
5. Pulp Fiction (1994, R, English) [8.9, 1324680]
6. The Lord of the Rings: The Return of the King (2003, PG-13, English) [8.9, 1215718]
7. Schindler's List (1993, R, English) [8.9, 865020]
8. The Good, the Bad and the Ugly (1966, Approved, Italian) [8.9, 503509]
9. 12 Angry Men (1957, Not Rated, English) [8.9, 447785]
10. Inception (2010, PG-13, English) [8.8, 1468200]
11. Fight Club (1999, R, English) [8.8, 1347461]
12. Forrest Gump (1994, PG-13, English) [8.8, 1251222]
13. The Lord of the Rings: The Fellowship of the Ring (2001, PG-13, English) [8.8, 1238746]
14. Star Wars: Episode V - The Empire Strikes Back (1980, PG, English) [8.8, 837759]
15. The Matrix (1999, R, English) [8.7, 1217752]
16. The Lord of the Rings: The Two Towers (2002, PG-13, English) [8.7, 1100446]
17. Star Wars: Episode IV - A New Hope (1977, PG, English) [8.7, 911097]
18. Goodfellas (1990, R, English) [8.7, 728685]
19. One Flew Over the Cuckoo's Nest (1975, R, English) [8.7, 680041]
20. City of God (2002, R, Portuguese) [8.7, 533200]
```

Task D: Similar movies (3 marks)

Suppose you are asked to implement a movie recommendation feature for IMDB. Based on the movie that a user is currently browsing, a list of *similar* movies will be presented. In Task D, the `similar` script takes in 2 arguments: a case-insensitive movie full title (this shall produce a single match. If not, just use the one with the most recent year as if it is the only match); and a number N (1 <= N <= 1000). It will then find its similar N movies (excluding itself) that share the maximum number of common genres (output nothing if no other movies have any common genres with the given movie). For those with the same number of common genres, they will then be ranked by the maximum number (including zero number) of common plot keywords. If both are the same, then they will be ranked by their IMDB scores and then by the number of votes (the higher the better).

```
./similar "Happy Feet" 30
1. Hairspray (2007) [4, 2, 6.7, 98693]
2. Confessions of a Teenage Drama Queen (2004) [4, 1, 4.6, 23408]
3. Aladdin (1992) [4, 0, 8.0, 260939]
4. Tangled (2010) [4, 0, 7.8, 294810]
5. The Book of Life (2014) [4, 0, 7.3, 45580]
6. Shrek 2 (2004) [4, 0, 7.2, 314630]
7. Enchanted (2007) [4, 0, 7.1, 142496]
8. The Road to El Dorado (2000) [4, 0, 6.9, 58300]
9. A Monster in Paris (2011) [4, 0, 6.8, 15790]
10. Sinbad: Legend of the Seven Seas (2003) [4, 0, 6.7, 36144]
11. Bandslam (2009) [4, 0, 6.4, 11958]
12. Freaky Friday (2003) [4, 0, 6.1, 96693]
13. Rugrats in Paris: The Movie (2000) [4, 0, 6.1, 8146]
14. Home on the Range (2004) [4, 0, 5.4, 13581]
15. The Lizzie McGuire Movie (2003) [4, 0, 5.3, 27580]
16. Alpha and Omega (2010) [4, 0, 5.3, 10986]
17. Roadside Romeo (2008) [4, 0, 5.3, 922]
18. High School Musical (2006) [4, 0, 5.2, 59254]
19. Alvin and the Chipmunks (2007) [4, 0, 5.2, 57276]
20. Alvin and the Chipmunks: The Road Chip (2015) [4, 0, 5.0, 9418]
21. High School Musical 2 (2007) [4, 0, 4.8, 39786]
22. High School Musical 3: Senior Year (2008) [4, 0, 4.5, 43795]
23. Alvin and the Chipmunks: The Squeakquel (2009) [4, 0, 4.5, 31649]
24. Alvin and the Chipmunks: Chipwrecked (2011) [4, 0, 4.4, 22838]
25. Hannah Montana: The Movie (2009) [4, 0, 4.2, 31760]
26. Doug's 1st Movie (1999) [3, 2, 5.0, 2448]
27. Monsters, Inc. (2001) [3, 1, 8.1, 585659]
28. The Iron Giant (1999) [3, 1, 8.0, 128455]
29. Toy Story 2 (1999) [3, 1, 7.9, 385871]
30. Arthur [3, 1, 7.4, 8495]
```

```
./similar "The Shawshank Redemption" 30
1. American History X (1998) [2, 1, 8.6, 782437]
2. The Grand Budapest Hotel (2014) [2, 1, 8.1, 475518]
3. The Town (2010) [2, 1, 7.6, 280228]
4. Escape from Alcatraz (1979) [2, 1, 7.6, 87090]
5. Bronson (2008) [2, 1, 7.1, 84817]
6. Bandits (2001) [2, 1, 6.6, 57038]
7. Contraband (2012) [2, 1, 6.5, 101977]
8. Faster (2010) [2, 1, 6.5, 80574]
9. Hurricane Streets (1997) [2, 1, 6.5, 1038]
10. Undisputed (2002) [2, 1, 6.1, 21542]
11. Prison (1987) [2, 1, 5.9, 2705]
12. Fled (1996) [2, 1, 5.3, 7073]
13. Civil Brand (2002) [2, 1, 5.3, 553]
14. American Heist (2014) [2, 1, 5.2, 12372]
15. Get Rich or Die Tryin' (2005) [2, 1, 5.0, 35834]
16. The Godfather (1972) [2, 0, 9.2, 1155770]
17. The Dark Knight (2008) [2, 0, 9.0, 1676169]
18. The Godfather: Part II (1974) [2, 0, 9.0, 790926]
19. Fargo [2, 0, 9.0, 170055]
20. Pulp Fiction (1994) [2, 0, 8.9, 1324680]
21. 12 Angry Men (1957) [2, 0, 8.9, 447785]
22. Daredevil [2, 0, 8.8, 213483]
23. Goodfellas (1990) [2, 0, 8.7, 728685]
24. City of God (2002) [2, 0, 8.7, 533200]
25. Gomorrah [2, 0, 8.7, 9638]
26. Se7en (1995) [2, 0, 8.6, 1023511]
27. The Silence of the Lambs (1991) [2, 0, 8.6, 887467]
28. The Usual Suspects (1995) [2, 0, 8.6, 740918]
29. Hannibal [2, 0, 8.6, 159910]
30. Luther [2, 0, 8.6, 70568]
```

Task E: Six degrees of Kevin Bacon (4 marks)

Tasks E & F are inspired by Six degrees of Kevin Bacon. The `shortest` script takes in two actor names (with case insensitive matching), and lists the shortest path (up to Six Degrees of Separation) between two given actors. In other words, if two actors are not connected within six degrees, you can assume that they are not connected at all. The output will be a list of actors, the movies and the years. If there are more than one the same shortest paths, output all of them (all output lines are sorted alphabetically in ascending order). Reference(Wikipedia): Six Degrees of Kevin Bacon

```
./shortest "tom cruise" "Jeremy Renner"
1. Tom Cruise was in Mission: Impossible - Ghost Protocol (2011) with Jeremy Renner
2. Tom Cruise was in Mission: Impossible - Rogue Nation (2015) with Jeremy Renner

./shortest "chris evans" "Scarlett Johansson"
1. Chris Evans was in Captain America: Civil War (2016) with Scarlett Johansson
2. Chris Evans was in Captain America: The Winter Soldier (2014) with Scarlett Johansson

./shortest "tom cruise" "Robert Downey Jr."
1. Tom Cruise was in Days of Thunder (1990) with Robert Duvall; Robert Duvall was in Lucky You (2007) with Robert Downey Jr.
2. Tom Cruise was in Days of Thunder (1990) with Robert Duvall; Robert Duvall was in The Judge (2014) with Robert Downey Jr.
3. Tom Cruise was in Jack Reacher (2012) with Robert Duvall; Robert Duvall was in Lucky You (2007) with Robert Downey Jr.
4. Tom Cruise was in Jack Reacher (2012) with Robert Duvall; Robert Duvall was in The Judge (2014) with Robert Downey Jr.
5. Tom Cruise was in Mission: Impossible (1996) with Kristin Scott Thomas; Kristin Scott Thomas was in Richard III (1995) with Robert Downey Jr.
6. Tom Cruise was in Mission: Impossible III (2006) with Eddie Marsan; Eddie Marsan was in Sherlock Holmes (2009) with Robert Downey Jr.
7. Tom Cruise was in Mission: Impossible III (2006) with Eddie Marsan; Eddie Marsan was in Sherlock Holmes: A Game of Shadows (2011) with Robert Downey Jr.
8. Tom Cruise was in The Firm (1993) with Holly Hunter; Holly Hunter was in Home for the Holidays (1995) with Robert Downey Jr.

./shortest "brad pitt" "will smith"
1. Brad Pitt was in Burn After Reading (2008) with Kevin Sussman; Kevin Sussman was in Hitch (2005) with Will Smith
2. Brad Pitt was in Ocean's Thirteen (2007) with Matt Damon; Matt Damon was in The Legend of Bagger Vance (2000) with Will Smith
3. Brad Pitt was in True Romance (1993) with Michael Rapaport; Michael Rapaport was in Hitch (2005) with Will Smith
```

Task F: Actors with N degrees of separation (4 marks)

The `degrees` takes in an actor name (case insensitive) and M & N degrees of separation (where 1 <= M <= N <= 6). It outputs all actors (no directors) that are exactly M to N degrees of separation (which represents the shortest path) from the given actor. Output the list of actors with the degree of separation indicated in brackets, sorted by the degree of separation and then by name, both in ascending order. For example:

```
./degrees "chris evans" 1 2
1. Andre Braugher (1)
2. Ari Graynor (1)
3. Benedict Wong (1)
4. Dane Cook (1)
5. Dominic Cooper (1)
6. Ed Begley Jr. (1)
7. Ewen Bremner (1)
8. Hayley Atwell (1)
9. Ioan Gruffudd (1)
10. James Franco (1)
11. Jason Patric (1)
12. Jason Statham (1)
13. JoAnna Garcia Swisher (1)
14. Kang-ho Song (1)
15. Keanu Reeves (1)
16. Mako (1)
17. Mia Kirshner (1)
18. Noel Gugliemi (1)
19. Robert Davi (1)
20. Robert Downey Jr. (1)
21. Sarah Michelle Gellar (1)
22. scar Jaenada (1)
23. Scarlett Johansson (1)
24. Troy Garity (1)
25. Aaron Yoo (2)
26. Adam Alexi-Malle (2)
27. Adam Arkin (2)
28. Adam Baldwin (2)
...
368. Wendell Pierce (2)
369. William Atherton (2)
370. William Smith (2)
371. Woo-sung Jung (2)
372. Zach Gilford (2)
373. Zach McGowan (2)
374. Zach Woods (2)
375. Zooey Deschanel (2)

./degrees "chris evans" 2 2
```

```
1. Aaron Yoo (2)
2. Adam Alexi-Malle (2)
3. Adam Arkin (2)
4. Adam Baldwin (2)
5. Ah-sung Ko (2)
6. Aidan Quinn (2)
...
345. William Atherton (2)
346. William Smith (2)
347. Woo-sung Jung (2)
348. Zach Gilford (2)
349. Zach McGowan (2)
350. Zach Woods (2)
351. Zooey Deschanel (2)

./degrees "tom cruise" 1 1
1. Adrian Pasdar (1)
2. Andrew Garfield (1)
3. Beth Grant (1)
4. Bill Cobbs (1)
5. Bonnie Hunt (1)
6. Brad Pitt (1)
7. Chad Lindberg (1)
8. David Oyelowo (1)
9. Debi Mazar (1)
10. Demi Moore (1)
...
45. Tom Waits (1)
46. Tom Wilkinson (1)
47. Tony Goldwyn (1)
48. Valeria Golino (1)
49. Vanessa Redgrave (1)
50. Vinessa Shaw (1)
51. Wilford Brimley (1)
52. William Smith (1)
53. Zo Bell (1)

./degrees "tom cruise" 1 2 | wc -l
674
```