CSE101 Assignment
HW4: It's a small world after all

- All code must be written in C/C++.
- Please be careful about using built-in libraries or data structures. The assignment instructions will tell you what is acceptable, and what is not. If you have any doubts, please ask the instructors or TAs.

1 Problem description

This is a fun assignment. Once you solve it, you might waste countless hours improving your pop culture knowledge. You have been forewarned.

Read this document carefully. Half the questions on Piazza can be answered by just reading the instructions.

Main objective: You are going to design a program that solves the "Six Degrees of Kevin Bacon" game. In terms of CS language, you will parse a dataset into a graph and perform shortest path computations. But I think it's more interesting to talk about Six Degrees of separation.

Given any two actors, the aim is to find a "path" between them, consisting to actors with whom they have coacted. For example, given Brad Pitt and Rachel McAdams, one can find a "path" between them:

Brad Pitt -(Fury: Blood Brothers)- David Ayer -(The Making of 'End of Watch')- Jake Gyllenhaal -(Southpaw: Inside the Ring)- Rachel McAdams

Meaning, Brad Pitt appeared with David Ayer in "Fury: Blood Brothers", who appeared with Jake Gyllenhaal in "The Making of 'End of Watch'", who appeared with Rachel McAdams in "Southpaw: Inside the Ring".

I will give you a snapshot of movie credit data (scraped from IMBD), and your program needs to find such paths. Efficiency in important, since your code will have to parse more than 100,000 movies (with a similar number of actors) to find these paths.

You can use any built in data structures for storing and processing the data. Do whatever you want; just do it efficiently.

Setup: You can access a Codio unit for this assignment. There is a directory "Sixdegrees" that contains a number of test input/output files, which shall be explained later. You must write all your code in that directory, and not in

any subdirectory. There are also some testing scripts. Please check out the README for more details on that.

Format and Output: You should provide a Makefile. On running `make`, it should create an executable "sixdegrees". You should run the executable with *two* command line arguments: the first is an input file, the second is the output file. You must provide a README with an explanation of the usage and a description of the files involved. *Please cite any sources you used, such as online code, code from a previous course (that you may have written), or extensive discussions with someone.*

All your files must be of the form *.c, *.cpp, *.h, *.hpp. When we grade, all other code files will be deleted. (So do not try to script some part in another language.)

Please read the following carefully, since it explains how the IMDB data is provided.

There is a file `cleaned_movielist.txt` that contains a snapshot of IMDB data.

Each line of this file is of the form <MOVIE> <ACTOR1> <ACTOR2> ..... To make parsing easy, the movie and actor names have underscores in them to represent space. Thus, to parse a line, you simply tokenize by space. The first token is the movie, all remaining tokens are actors. For example, here is a line you may recognize.

Terminator_2:_Judgment_Day Arnold_Schwarzenegger Linda_Hamilton Edward_Furlong Robert_Patrick

This file is the "data file" and is not provided as an input to your program. It is a fixed file. **There may be different movies with the same name.** Each line is a different movie, even if the movies have the same name. Be wary of this issues, since it can lead to errors.

Each of line of the input file is of the following form:
<ACTOR1> <ACTOR2>

For each such line, the output file must contain a shortest path between the actors, formatted exactly as follows.

- If either <ACTOR1> or <ACTOR2> are not present in the data (meaning there is no movie with them), output "Not present".
- If there is no path between <ACTOR1> or <ACTOR2>, output "Not present".
- If <ACTOR1> is the same as <ACTOR2>, just print <ACTOR1>.
- (The real case) Print a shortest path between the actors, with the movie connecting adjacent actors. For example:

  Frank_Sinatra_Jr. -(Do_It_in_the_Dirt)- Suzan_Averitt -(Rebel_Dabble_Babble)- James_Franco -(Love_Conquers_All:_The_Making_of_Tristan_+_Isolde)- Jim_Lemley -(Through_the_Eyes_of_Director_Timur_Bekmambetov)- Thomas_Kretschmann -(Prince_Valiant)- Katherine_Heigl

Pay attention to the space and parentheses format, and follow it exactly. The line has: <ACTOR1> -(<MOVIE>)- <NEXT ACTOR> -(.... So Frank Sinatra acted with Suzan Averitt in "Do It in the Dirt", and Suzan Averitt acted with James Franco in "Love Conquers All: The Making of Tristan+Isolde", etc.

Your output may be different, because there could be multiple paths with different movies. You only need to provide one movie for the link between two successive actors. Note that your output path must have the same length as my output path.

For more clarity, look at the test input/output files.

Data structure suggestions: In one word, graphs. In one word and one acronym, graphs and BFS.

How to represent your graph? Do whatever you want. Think about adjacency lists, and feel free to store neighborhoods in unordered map data structures.

The test cases:

- simple-input.txt, simple-output.txt: A simple test case with an output of "Not present" and a path with a single vertex. Your output must exactly be the same.
- more-input.txt, more-output.txt: Ummm...more input and output.

2 Grading

You code should terminate within two minutes for any input file with at most ten inputs (the BFS does take a bit of time).

1. (10 points) I'll throw some more test inputs.
2. (5 points) If the output is correct on simple-input.txt and more-input.txt, you get five points.

3 How did I get the data

I got the raw data from `https://datasets.imdbws.com/` and `https://www.imdb.com/interfaces/`. To get it manageable size, I (and a student I had hired over the summer) only kept non-adult titles. Annoyingly, the existing Codio box does not have enough memory to process this file, even though my laptop did it with ease. So I trimmed it to US releases, and even deleted half of those at random. A lot of well-known movies are not present (sorry!).