CISC 1115 Homework 10

This homework will give you practice reading in data, writing to files, writing and calling methods, and creating, populating, printing, sorting, searching, and modifying parallel arrays.

Write a Java program to create a database of films. Our database will be taken from ***Empire Magazine's*** Top 50 Best Sci-Fi Movies Of All Time.

First, use the empireList.txt file already made for you here. (Download and import it, copy it into a new text file with the title empireList, etc. Do it however you see fit)

- In the main method, create four parallel arrays to hold the film's rank, title, year, and director respectively.

- Create a method called **readData()** to populate the arrays from the empireList.txt file. The data is in the order of ***rank, title, year, director*** (each piece of data is separated by a comma (',') how can we use this to read each piece of data?)

- Create a method called **printData()** that will print a header and the data from the arrays to an output file in a four-column table format: **Rank Title Year Director**

- Use bubble sort to sort the data in three different ways:

    - Create a method called **sortByRank()** to sort the data in ascending order by rank.
    - Create a method called **sortByYear()** to sort the data in ascending order by year.
    - Create a method called **sortByTitle()** to sort the data in alphabetical order (A-Z) by title

- After each sort call the **printData()** method to print to file, each with a specific header of how the data is sorted

- Create a method called **mostPopular()** that will answer the following question: ***who is the most popular director?*** (who has the most movies on the list) and print to the output file. (If more than one director shows up the most, you only need to print the first one encountered -- hint: remember the code for determining the most frequent integer)

- Create methods for searching the data by movie, year, or director. (hint: use a separate method for each)
    - search for movie by title
    - search for all movies in specific year
    - search for movie(s) by director name
        - if not found output error:
            "Unfortunately there is no record of that film in this list."
        - if found:
            Ranking. Title (Year) dir. Director

- Create a method called **addFilm()** that prompts the user to add a new movie to the data
    - prompt the user for the film that they would like to add
    - **the film cannot be already on the list BUT can be a remake** (so it CANNOT have the same title AND year)
    - If the film is not already on the list:
        - add ranking as N/A (hint: how can we include rankings such as 1, 2, 3, etc. AND N/A. What datatype should we use?)
        - add title, year, and director

- Sort by ranking (again) and print to file.

The output file should be in the following order:

- The entire data set, ordered by rank
- The entire data set, ordered by year
- The entire data set, ordered by title
- Who the most popular director is
- The film, year, and director searched for, and the results of each search
- The entire data set, ordered by rank again (your added film should be in the list now)

Print everything except for prompts to an output file.

Submit the Java file and the output file.