AP Computer Science Name_____
Movie Actors Index

A movie database consists of a movies file and an actor file. The task is to produce a cross-index, listing for each actor or actress of all the movies that he or she starred in.

The movies file, a pre-written text file, lists movies in chronological order by date of release. Each line has the year of release, the movie title, and the list of actors and actresses starring in the movie. The actors file, also a pre-written text file, contains the names of actors and actresses, one name per line.

The actor cross-index program should display the actor's name and the list of movies he or she has starred in, starting from the most recent.

Your program should include the following two classes (Files):

1. Write a `LinkedListFromFile` class that is derived from `LinkedList`. In addition to the default (no-args) constructor, supply a constructor that takes the file name as an argument, reads lines of text from the file, and adds them to the list (in the same order). Use the Scanner class.

2. Write a `MovieActorsIndex` class that includes your main and creates two linked lists, one from the actors.txt file, and the other from the movies.txt file. Use your `LinkedListFromFile` class to construct the lists.
   Provide a simple four-option menu for the user:

   - `displayList` : Prints out the list of actors only, one per line.
   - `displayActorsMovies` : Takes on actors name, and displays all the movies they starred in, in chronological order.
   - `displayAllActorsMovies` : Displays all actors and the movies they have starred in, in chronological order.
   - `quitWithFileOutput` : Exits the program, printing all actors and movies starred in, in reverse-chronological order (Most recent first).

   When writing any of the display methods, use a simple iterator to obtain the names of the actors in the list. Use a list Iterator (ListIterator object) to obtain records from the list of movies.

   Assume that the names of actors start in a fixed position in the movie record. If the record contains additional information (e.g., the movie director's name), assume that it starts at a fixed position after the names of the actors. Search for a matching actor name only in a substring that cuts to the relevant portion of the record.