CSCI 161 Final Exam Online Problem (EXAMPLE)

This portion of the exam is open book and notes (your own only, of course). You may not consult anyone other than the instructor during this portion of the exam. You MAY use any code that you have previously developed in order to complete this programming problem. ***Please read the problem in its entirety before beginning.***

Description: Develop a Java program that reads and displays motion picture (movie) box office information:

a. Reads movie box office sales figures from a file with each line of the file containing year of release, gross sales and movie title for a given movie; storing the data in the arrays named *years*, *grossSales*, and *movies*, respectively.
b. Prints out all data (year of release, gross sales, movie title) in a formatted fashion as well as outputting of the top-grossing movie.

Input Specification: Your program will need the file **GrossSales.txt**. This file contains year of release and gross sales figures (in inflated whole dollars) information for movies, and is of the following format (*not the complete file*):

```
1937 963560000 Snow White and the Seven Dwarfs
1939 1786074500 Gone with the Wind
1940 595845500 Pinocchio
1941 734104300 Fantasia
1942 563218000 Bambi
1945 554666700 The Bells of St. Mary's
1946 451020400 Duel in the Sun
1946 486200000 The Best Years of Our Lives
1950 533064600 Cinderella
1952 530400000 The Greatest Show on Earth
```

The complete file is available for copy on the Linux workstation from the instructor's public folder to the current working directory of your workstation via the following command:

```
cp /home/grader/rogers161/Public/GrossSales.txt .
```

Output Specification: Your program should output two sections of information. The ***first*** section displays the formatted list for all the movies as shown below and the ***second*** section prints out the highest grossing movie, using the same columns and formatting as the list above, but with a different heading.

For the format of both sections, their headers and their data, see the following example:

```
YEAR   GROSS      MOVIE
----   -------    --------------------------------------------------
1937     $964M    Snow White and the Seven Dwarfs
1939   $1.786B    Gone with the Wind
1940     $596M    Pinocchio
1941     $734M    Fantasia
1942     $563M    Bambi
1945     $555M    The Bells of St. Mary's
1946     $451M    Duel in the Sun
1946     $486M    The Best Years of Our Lives
1950     $533M    Cinderella
1952     $530M    The Greatest Show on Earth

TOP GROSSING MOVIE (ADJUSTED FOR INFLATION)
-------------------------------------------
1939   $1.786B    Gone with the Wind
```

Formatting Notes:

a. Your output, including all headings, hyphens/dashes, spaces between columns, etc. should match identically.

b. Note the special formatting of the gross sales column (`GROSS`) as the data in that column will always be in millions or billions of dollars for the sample data provided. Figures in billions should be formatted with three decimal places, e.g. `$1.259B` and figures in millions with no decimal places. Each figure starts with $ and ends with M or B.

Hints and Suggestions: Following are several hints and suggestions to aid you the development of your program:

1. **Use global array variables:** Make your life easier and start with the arrays that you need as global class variables declared at the top of your program instead of in the main method (thus avoiding the need to pass them around.) The arrays that you need are:

```
/** Class Variables **/
public static int[] years;              // read from file
public static int[] grossSales;         // read from file
public static String[] movies;          // read from file
```

2. **Printing a Movie entry:** When printing a movie line with the columns shown as in the output specification, it will be easiest to use the ***printf*** method.

3. **Develop methods wisely** – If all your arrays are in the global space there is no need to pass them to any of your methods. Other than that, helpful methods *might* include:

```
    // Returns number of lines in a supplied file
    int numLines(String fname)

    // Reads file specified and fills arrays
    void readFile(String fname)

    // Given a movie gross sales figure returns a string
    // in the format:
    //       $%.0fM, or $%.3fB   (e.g. $128M, $1.247B)
    // for millions and billions of dollars, respectively.
    String formatSales(int grossSales)

    // Prints information for a single movie given its index
    void printMovie(int idx)

    // Prints header and then all movies
    void printAllMovies()
```

4. **Iteratively Develop your Program:** Build your program in small increments, testing as you go, so that you always have something that compiles. A program that is nearly complete and cleanly compiles will receive more points than a complete program that does not compile! Remember that you may submit early versions of the program as many times as you like!!

Grading: This portion of your exam is worth 25 points, with the following grading rubric used. Your program earns percentages towards that total 25 points as follows:

+70% - Compiles, runs, and **GrossSales.txt** file properly read into *year*s, *grossSales* and *movies* arrays.
+10% - Prints movie stats in properly formatted columnar data.
+10% - Prints sales figures in *friendly* millions and billions format.
+10% - Determines and prints highest grossing movie.

* Credit: Box office and inflation figures based on data from www.boxofficemojo.com