Computer Science, Claremont McKenna College
CS51 - Introduction to Computer Science, Spring 2015

Problem Set 8 Due: 11:55 PM, Apr. 5, 2015

General Instructions

Please carefully read and follow the directions exactly for each problem. Files and classes should be named exactly as directed in the problem (including capitalization!) as this will help with grading.

You should create your programs using your preferred text-editor, the Eclipse text editor, or jGrasp. Do not use a word processor such as Microsoft Word, WordPad, Google Docs, Apple's Pages, etc...

Your programs should be formatted in a way that's readable. In other words, indent appropriately, use informative names for variables, etc… Points will be deducted if your indenting style makes it difficult to follow your code. If you are uncertain about what is a readable style, look at the examples from class as a starting point for a reasonable coding style. .

Your programs should compile and run without errors. Please do not submit programs that do not compile! It's better to submit partial implementation that compiles as oppose to implementations that do not compile.

At the top of each file you submit, you should put your name and your email as comments.

This homework set should be submitted via Sakai using the Assignment Menu option (on the left pane). You should only submit the requested ".java" files. Do not submit ".class" files or Eclipse-specific project files. Finally, please do not submit files using Sakai's Dropbox Menu.

Turn in the following file(s):

```
BabyNamesStats.java
names.txt (provided to you)
TicTacToe.java
```

Problem 1

In this problem, you will practice reading from a file and writing to a file utilizing Scanner and PrintStream objects.

Every 10 years, the Social Security Administration gives data about the ranking of the most popular boy and girl names for children born in the US since 1990. This file is provided to you in a file called: `names.txt`. The format of this file is that each line contains a name and then a sequence of numbers that indicate the rank of that name for each decade starting with 1900. Each number represents a ranking for a decade and a value of 0 indicates that the name was unranked. The highest rank is 1 and the lowest rank is 999. For example:

```
Abe 248 328 532 764 733 0 0 0 0 0 0
```

This line shows that Abe was ranked 248 in the decade starting in 1900, ranked 328 in the decade starting in 1910, ranked 532 in the decade starting in 1920, etc… Note that Abe was unranked starting in 1960 and the highest rank Abe achieved was 248 in 1900.

Your task in the problem is to write a program in a class called `BabyNamesStats`, that reads the file `names.txt` to find various statistics about each name. Your program should then write this information out to a file called "rank_results.txt". The format for each name written should be: the name, followed by the number of decades it was ranked, followed by the average rank (for the years it was ranked), followed by the highest rank. For example, the results for Abe should be:

```
Abe, Number of Decades Ranked: 5, Average Rank: 521, Highest Rank: 248
```

Note that some names are listed but never ranked. You should handle this case by just printing the name and then the comment: "never ranked". For example:

```
Ziola, never ranked
```

You are provided with the `names.txt` file and if you are using Eclipse, this file should be placed in your chosen *project* folder (not in the src folder!). If you are using jGrasp or the command line, this file should be placed in the same folder as your source code.

Hints: consider processing a line at a time instead of a token at a time and you may want to create a smaller version of names.txt with perhaps just 5 lines to check that your code is working correctly.

Problem 2

In this problem you will implement a Tic Tac Toe game using a 2D array. You are given an existing implementation in `Ttt.java` but this version does not use 2D arrays. Copy `Ttt.java` to `TicTacToe.java` and rename the class to `TicTacToe`. Inside the class, you will see nine static variables declared as follows:

```
private static char s1;
private static char s2;
private static char s3;
private static char s4;
private static char s5;
private static char s6;
private static char s7;
private static char s8;
private static char s9;
```

You are to replace these nine lines with the following single line:

```
private static char[][] board;
```

In other words, you will be using a two dimensional array of an appropriate size to represent the game board. Remember that this one line declares a 2D array, but it does not actually allocate any memory for it so you will need to decide where in the program you will do the allocation.

With this new data structure (the 2D array) to represent the game board, the rest of the program will have to be updated in such a way that the program now works again. Your task is to update the remainder of `TicTacToe.java` and make it work without any errors. Do not change the signature of the existing methods unless you absolutely have to, in which case you must justify the change(s) in the comments.

Hint: before you try to solve the problem, play with the given `Ttt.java` and try to understand the given program.