**CP122 - Lab #8 - File I/O: Benford's Law**

**Assignment Overview**



**Benford's Law and World City Populations (Benford.java)**

A scientist named Frank Benford noticed something interesting in 1938: the earlier pages of a book of logarithms were more worn than the later pages. This meant that scientists were looking up logarithm values for numbers with smaller leading digits more frequently than for larger leading digits.

When working with any normal dataset you might assume that each leading digit would have the same probability of occurring, namely 1/9. For many real-world datasets, this turns out to be incorrect. Dr. Benford went on to test a number of different datasets and found that smaller leading digits were more common for things like electricity bill totals, house numbers, length of rivers, etc. [Read a bit more about it](http://en.wikipedia.org/wiki/Benford%27s_law), if you're interested.

For this project we're going to work with a real-world dataset: [populations.txt](http://cs.coloradocollege.edu/~mwhitehead/courses/backup/CS1/Labs/8/populations.txt). The dataset contains a couple thousand lines of data concerning populations of large cities throughout the world. Each line of the data file has the name of a city followed by its population. These values are separated by a space character. Your job is to figure out if this list of city populations generally follows Benford's Law.

In order to solve this problem, you're going to need to read in the dataset file and process it city by city. This will be a bit tricky because each line has two types of data: a text city name and an integer city population. Remember that the Scanner class has several different methods for reading in data from a file: .nextLine(), .next(), .nextInt(), .nextDouble(), etc.

You may be tempted to use .nextInt() to read in each city population...this may not be the easiest way to solve the problem, though. To see if the city populations follow Benford's Law, we'll need to check the value of the first digit of each population. If you scan in the city population as an int, it will be more difficult to get at that first digit.

Once you have the basic scanning working, you're going to need to count the number of occurrences of each different leading digit. What would be the best way to store these counts? How many variables will you need? Is there any other kind of data structure that might help out? You should also keep the total number of cities in another variable, so that you can report the percentage of each leading digit at the end.

Again, make sure your program handles possible exceptions cleanly.