# Project Gourd #1

Suppose the following list represents the number of rocks inside jars in a laboratory (each integer represents the number of rocks in a single jar):

| | | | | |
|---|---|---|---|---|
| 4 | 87 | 118 | 150 | 193 |
| 21 | 93 | 119 | 160 | 207 |
| 30 | 111 | 125 | 176 | 214 |
| 80 | 112 | 130 | 184 | 241 |
| 83 | 113 | 143 | 191 | 243 |

If we only take a look at the leading digit of each of these numbers (highlighted in the list above), one at a time, we can generate the following table that represents the frequency of each leading digit and its percentage (of the total number of integers/jars):

```
-------------------------------
Leading Digit  Count           %
-------------------------------
0              0           0.00%
1              14         56.00%
2              5          20.00%
3              1           4.00%
4              1           4.00%
5              0           0.00%
6              0           0.00%
7              0           0.00%
8              3          12.00%
9              1           4.00%

-------------------------------
TOTAL          25        100.00%
===============================
```

**Your task:**
Write a Java program that takes as input some arbitrary number of integers and generates a table like the one above that is representative of that set of integers (although there's no need to bold the top and bottom rows). I will provide several sets of integers, each set in a single file. You will submit your **single .java source** file. You will also be prepared to discuss, in class (after the assignment is due), any observations that you make with respect to the results (i.e., what you observe in the table data).

**Requirements:**
    (1) You will submit only a **single .java source** file.
    (2) You must read the input from standard input (`stdin`).
    (3) You must display the table exactly as mine is displayed in terms of content and formatting.
        (a) Clearly, the table header will be exactly the same as in the table above.

(b) Clearly, the actual values in columns two and three will be different (for different input data).
(c) Clearly, the values in the first column will be the same.
(d) Clearly, the values in the bottom row will be similar (the only difference should be the total value for the `Count` column for different input data).

**Notes:**
(1) You may not understand everything that has been written in this document so far.
(2) You will most likely not know how to entirely solve this problem.
(3) You probably do not have all of the knowledge needed to finish the assignment.
   (a) That means that you will most likely not be able to submit a working assignment.
(4) You most certainly won't find this problem on the Internet (and therefore won't be able to copy a video or some snippet of code from someone who has already solved the problem).
(5) But you can Google the things that you do not know how to do:
   (a) How to read from standard input in Java.
   (b) How to read the input as integers.
   (c) How to identify the leading digit of an integer.
   (d) How to calculate the percentage.
   (e) How format the output so that it is aligned.
   (f) How to format a floating point number in the output (table) so that it has only two digits to the right of the decimal point.
   (g) How to display a set of floating point numbers so that they are aligned on the decimal point.

**The point:**
(1) You need to learn how to code without relying on videos or snippets of code showing you how to solve the entire problem (i.e., from someone who has already solved the problem).
(2) You need to know how to Google for tidbits that can help with issues related to small tasks (e.g., how do I read from standard input in Java?).
   (a) Which is related to learning how to properly structure a question so that it returns the best set of answers.
(3) You need to think through the problem before you get your hands on the keyboard in order to figure out how to solve it at a high level.

Some of you (maybe most of you) will not be able to submit a working program. That's OK! I would rather have something that you have thought through and worked on individually that's not working than something that's working that you copied.

Don't fret! Once submitted, I'll go over some things (tactics) in class and give you the opportunity to try again and fix issues in your code. We'll keep doing this a few times so that by the end, you have a working program – and you've learned something in the process!