# Exercise 3

## Implementation

Adapted from: https://faculty-web.msoe.edu/hasker/se3800/ex/3/

For this exercise you will work in structured pairs. The pairing model is called Collaborative-Adversarial Pairs. The two of you should collaborate and agree on a design. Then you should split into adversaries, with one implementing the design and the other implementing the tests. You should try to split your design in such a way that each of you can be a tester and an implementer once. The tasks are:

- Develop it as a Java console app that reads standard input and writes all results to standard output.
- Each pair of students needs to collaborate on the design. There are no constraints on your design other than it must be a design you're willing to defend.
- Create a git repository for the project in Bitbucket and invite instructor (https://bitbucket.org/bcdennis72/ ). In addition to source files, check in files needed to build the system. Do not check in .class files or IDE project files. Create a .gitignore appropriate for your IDE.
- Ensure each commit has meaningful, traceable commit messages. "Writing tests" or "wrote class X" is rarely helpful; describe what progress was made. Traceable means that you trace each commit back to a user story - include the story identifier in the commit message. Note that git does allow updating previous commit messages, so take advantage of that when necessary.
- Ensure your code is readable. There should be appropriate whitespace, but no tabs.
- Include JUnit tests for your implementation. Use EMMA or an equivalent tool to ensure your tests cover at least 90% of your code.
- Each group member is to write their share of the code (50% each for 2, 30% each for 3).
- Each group member is to write the tests to exercise the other member's portion of the system. Be aggressive: attempt to write tests that are highly likely to find errors in your partner's code.
- Clearly document who wrote which portions of the system.
- Be sure your the final version of your implementation is checked in by the due date.

You may use the user stories from either partner’s submission from Exercise 2.

**Due:** Wed 1/6 @11:59 PM
**Submission URL:** http://goo.gl/forms/59RwxeXWbc

**Grading Criteria:**

1. Were the commit messages traceable and meaningful? (10pts)
2. Was Bitbucket set up and the instructor invited? (5pts)
3. Do the tests meet the minimum requirements? (10pts)
4. Was the work fairly distributed and the CAP process followed? (5pts)
5. Were all the features implemented? (20pts)