Assignment 1 Report

Group Members (Name, netID, github username)

Allison Martin	jam1250	totally85
Kyra Johnson	kvj18	kvj18
Will Nobles	wtn20	willtnobles
Brandon Long	bhl47	bhl47

Lessons Learned

We had an interesting time with test driven development (TDD) on this project. Initially we encountered many setbacks due to having chosen to develop in C++ and use gtest as our testing software. It turned out that just installing and using gtest was much more difficult than we anticipated (we never actually got it to work), so we decided to convert our code to python and use pytest. Pytest was much more simple to use and allowed us to see the many benefits of TDD such as improved efficiency in debugging code. Using a behavior driven approach combined with TDD allows you to make sure each function of your code is working properly as you are developing, this is much better than developing code and ending up with a coding or logical error that you must spend time finding and fixing.

I would choose to apply for a TDD project. Test driven development can be tedious since you must pre-write/run tests when you know that they are going to fail, but future time it saves in debugging/testing code after completion should more than make up for this. Also as a programmer it helps you see exactly what is going on with the program as you code it and prevents any simple or dumb mistake from ruining the whole program.

Select Naming and Organizational Conventions

We are working with Python, and we are using the pytest framework to test our functions. We have organized our work by naming the files similarly to their corresponding tests. The files names accurately describe the functions and tests inside them. The names of our testing functions will begin with "test_", with the rest of the function name describing what is being tested.

Setup and Use Instruction

Python-version 3.6.0 download link: https://www.python.org/downloads/release/python-360/
PyCharm Community Edition 2016.3.2 download link:
https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=windows&code=PCC

Operating System-Windows

Test Framework instructions:

Using **pytest** to develop our unit tests was quite simple to set up. It required no tedious steps beyond downloading python itself and running the tests in command line. The following steps were taken to set up the testing framework:

1. We downloaded Python version 3.6.0. Some of us used pycharm as our development environment.
2. In order for pycharm to use version 3.6.0 of python, we had to take the following steps in the pycharm environment: file->settings->project->project interpreter->Python 3.6. Then, we clicked "+" to add pytest.
3. We wrote our unit tests and our production code.
4. Using the command line, we executed the necessary command to run our unit test and recorded the results.

Testing Output Report