Isfandyar Rabbani, 932960451
CS 362
Assignment 4

Random Testing

I created two random tests for cards. By joining the functions that are considered separate from each other, there was an increase in code coverage for the overall program because more than one component is being utilized for a given test.

Code Coverage

There was not enough code coverage in my previous tests, so during this iteration I made sure to increase the code coverage. That way, a higher percentage of the function or module was used during the unit test. This is good practice as it can test the robustness of the code to ensure it really is modular and all outcomes are predictable.

Unit vs Random

Unit tests are good for checking that one part of the program is working correctly, and that is an important component of building a robust system. On the other hand, random tests are just as necessary because they provide the opportunity to test the intended work flow of the system, or even to ensure that the system maintains correct functionality under an unusual work flow. Perhaps the parts work individually, but there might be cases where the parts are not interacting with each other correctly. Both are valuable types of tests to consider. Regarding the tests that had more coverage, the random tests because the tests by that time were more honed to cover everything. Also, the unit testing did not have as many results. There is also so much more repetition with the random testing that it ends up testing every single permutation possible. Regarding fault detection capability, the random testing was better because it tested every single variable value, and because I programed the random test using asserts to break at the spot where it was designed to. This is opposed to tallying the faults, used in the unit tests, which would not as easily identify where in the code the program is breaking.