HW4 CS103 Fall 2019 Johnstone due: 10.23.19

HW4 is about gradually generalizing a function, extending its power to a more general case.
I also added a challenge question (pages) to entertain you (which has nothing to do with generalization).

HW4 syntax tested

Some of the earlier syntax. You should only use syntax covered in lecture for this homework.

HW4 style constraints

- each line should contain no more than 80 characters
- only syntax taught so far in CS103
- no use of modules

Deliverables

hw4_19fa103.py
hw4_tests_19fa103.py

Elaboration on tests

- write at least 2 test calls per fn in hw4_tests_19fa103.py;
both should be different from the test data I have provided
- you should write test calls even for functions you do not successfully implement

Grading

grading a problem is correct if all autograde tests pass
grade will be determined by #correct, with adjustments for testing and style
1: D; 2: C; 3: B; 4: A; 5: A+
testing script adjustment testing script incomplete: decrease of one letter grade
style grade adjustment decrease of one letter grade if code does not reflect good style or does not implement the code in the requested way (e.g., for loop in Q1, no for loop in Q2)
examples of good style clarity, simple syntax, good variable names
examples of poor style confusing, strange syntax, strange variable names

HW4 problems

The problems are fully defined by the docstrings in hw4_19fa103.py I am not giving you a tester this time: gradually passing the responsibility to you to test your code.

In this homework, you should assume that the input is correct. To simplify your code, there is no need to test the type or value of the input parameters (although that is a great thought). Imagine that the test data has already been vetted by a separate preprocessing step.