**Step 1: Download the starter code.**

In this assignment, we are providing starter code.

Here is a file containing headers and docstrings for the three functions that you will test:

[a1.py](https://d3c33hcgiwev3.cloudfront.net/_c0fce9830ee85040cf81e23d19ee682d_a1.py?Expires=1491523200&Signature=LnMEtKBk5oA1rN3UYRSHvt%7EreC9NEtuBhoXdUkaq1hvRD4%7EAw4dcLYKka-SGnBLbBSaknMYK9tlj4JldFuQYgmMPA0NI2MJ7j2ju21eQjJokbnGv6SJUrLUQfFfWI8V22CJDneTvcinA2iGOG5XMDNMPqGcCjnKHLTvhimp3Rew_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

Here are three files, each containing one unittest class header. You will write your unittest methods in these files.

[test\_num\_buses.py](https://d3c33hcgiwev3.cloudfront.net/_c0fce9830ee85040cf81e23d19ee682d_test_num_buses.py?Expires=1491523200&Signature=Q2IwTpS4smOyD7Gw5JkSB0WOG9CvydJ3%7ESUFsDzHA%7ECK-Ye49JooPmy8dW5gRx8BbFkHMazKGIyqhyXblwc57ALWHzxFkFBk1VeWz83qItmYaKc7F4pJhCysBmOQdilLRPGew0OlLDig4c11DqE9zXim2WTmNbVWTQHSaapUE0o_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

[test\_stock\_price\_summary.py](https://d3c33hcgiwev3.cloudfront.net/_c0fce9830ee85040cf81e23d19ee682d_test_stock_price_summary.py?Expires=1491523200&Signature=XKB2A7s1NZMApg8QWJm3QCHqsM6XmoRflEVyFuomKNLbtO2GZN1tVzdmGgAAHNZBrP5R75Tuz7MawyZGtnN2K4o5Eb3c1OxFZWWKg-Gwrum4liB%7EvmpX76ry-qzYnjlf1Sk3kKMP2zknbZzG9UVsEPyrIwKTdFhhj36irymOm%7Es_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

[test\_swap\_k.py](https://d3c33hcgiwev3.cloudfront.net/_c0fce9830ee85040cf81e23d19ee682d_test_swap_k.py?Expires=1491523200&Signature=jVYJkRL3lvs7iApPaR0wmULYY01CzGPQsX3EpU-DmpnuGKjSbtKFvCMYm2-ysmrZLKnhsnxq8F4Wl6K9MYmFmBBQ7UaZArzf9VVnk8f-AKdOUnOLXtmEmnVUAhEdibeGjjVbd9tXkUoMCZoJEP3vG-kwZ%7EnJYH5wxDU%7EPZc-Xmk_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

**Step 2: Read and implement the three functions in a1.py.**

The starter code contains a header and a docstring for each function, so the descriptions are not repeated here. Each docstring contains one example call. To gain a better understanding of each function, you should add additional examples to the docstring.

You will not submit a1.py, but we recommend that you implement these functions because it will help you come up with good test cases.

We encourage you to post BUGGY implementations of the functions to the discussion forum so that other people can try out their testing code. Do not post your test cases, of course!

**Step 3: Choose test cases.**

Next, you should choose your tests for each of the three functions. As we did in the Choosing Test Cases video, write down (on paper) a table containing:

* a description of each test,
* the values of the arguments, and
* the expected result.

As you build the table, consider the various factors (size, dichotomies, boundaries, and order) to make sure that you have sufficient test coverage. Avoid duplicate tests.

You will not submit this part and it will not be marked. It's important to do this, though, in order to make sure you have a thorough set of tests.

One of the three functions mutates a list. In your tests for this function, be sure to test that the list was mutated properly.

**Step 4: Write test method headers, including docstrings.**

Once you have finished planning your tests on paper, write one method header for each test in the appropriate test file.

You must follow these guidelines because you will be marked on them:

* Give the test method a meaningful name that includes the name of the function and briefly describes the test case.
* Include only one of your tests in each method you write.
* Include a docstring description for the method that clearly describes the purpose of this test case. You do not need to include the values of the arguments in this description, but you should include an appropriate description of those values such as "An odd length word that begins with a vowel."

You will not be marked on grammar and spelling, but you will be marked on clarity. Please make an effort to be clear.