Work with your neighbor. (This will be graded for participation only.)

1. Consider the following problem specification: *Write a program that reads a file and computes (and prints out) the length of the longest line in that file.*

   Write the following black box tests for this program:

   ANS:

   a. two error cases
      the file does not exist
      the file is readable but not organized into lines (it's a JPEG,…)

   b. two edge cases
      the file has one line
      the file is empty

   c. one regular (normal) case
      the file has many lines, each line containing readable values

2. Consider the following problem specification: *Write a program that reads a (possibly empty) file containing only numbers (and whitespace) and prints out the difference between the smallest and largest numbers. An empty input file should generate no output.*

   Write the following black box tests for this program:
   ANS:

   a. two error cases
      a file that does not exist
      a file that does not have numbers in it

   b. two edge cases
      the file has one line with one number
      an empty file

   c. one regular (normal) case
      a file with many lines, one number per line

3. The function `my_sqrt(n)` returns the square root of `n`. Use an assert statement to enforce that `n` must be non-negative.

```
import math

def my_sqrt(n):
```

ANS:

```
    assert n >= 0, "n must be non-negative"

    return math.sqrt(n)
```

4. Write `assert` statements to enforce the following:

   a. `x > y`

   ANS:

   ```
   assert x  >  y
   ```

   b. `word` is a key in the dictionary `word_count`

   ANS:

   ```
   assert word in word_count.keys()
   ```

   c. `i` is an even number

   ANS:

   ```
   assert i % 2 == 0
   ```

   d. the string `s` has at least 2 characters

   ANS:

   ```
   assert len(s) >= 2
   ```

5. Suppose that you have a list of numbers, `num_list`. You need to ensure that `num_list` has at least one even number in it. Write a function `has_evens(num_list)` that can be used in the assert statement below:

```
assert has_evens(num_list), "no evens in num_list”
```

ANS:

```
def has_evens(num_list):
    for elem in num_list:
        if elem % 2 == 0:
            return True
    return False
```