

For this exercise, I'm providing slightly buggy code. Your task is to debug them (ideally, using the debugger!) and see if you can figure out what's wrong, and then fix it.

Task 1:

We provide a file, `lucas.py`, which contains a function that is supposed to return a list of the first n digits from the Lucas sequence. A [Lucas sequence](#) is very similar to a Fibonacci sequence, and except it starts with the values 2 and 1. Subsequent values are found by summing the two previous values. Thus the first 7 values are 2, 1, 3, 4, 7, 11, 18. Tragically, there's a bug in the code provided! Use some of the debugging techniques from the reading to see if you can figure out where things are wrong.

Task 2:

We've done some work so far looking at prime numbers, using the simplest, and least efficient, algorithm to do so: trial division. No, we're going to look at a different technique: using the Sieve of Eratosthenes, a cleverer technique, and one that's also faster. I've provided a file, `sieve.py`, which contains an implementation of this algorithm, and a link to [the Wikipedia article](#) that explains how the algorithm works.

Like the above, however, my code unfortunately doesn't work. See if you can figure out why and how to fix it!