7 - Sieve of Eratosthenes

The sieve of Eratosthenes is a way of computing all the prime numbers below a certain number. (A prime number is a number that is only divisible by itself and 1). This algorithm is explained excellently in this video

(Links to an external site.)

, or you can read about this ancient algorithm on Wikipedia

(Links to an external site.)

.

Implement this algorithm:

- Implement a function cross_out_multiples that takes as arguments a list of boolean values (true/false) called is_prime and a number n. The function sets the boolean values at all multiples of n (2*n, 3*n, 4*n ...) that are in the list to false.
- Implement a function sieve(n) which gives back a list of all primes below n.

Put your code in sieve.py This program is tested via unit tests.

Assuming that you have implemented cross_out_multiples correctly, the for loop in sieve should work as follows:
It goes from 2 until the end of the list. and for every i, it first checks whether it is a prime. and if it is, then it calls cross_out_multiples(list, i).

So given the input [True, True, True, True, True, True, True]
i starts at 2. list[2] is indeed true. So it calls cross_out_multiples(list, i=2).
Which will give us the following list:
[True, True, True, True, False, True, False]

and so on.