1. How many seconds are in an hour? Use the interactive interpreter as a calculator and multiply the number of seconds in a minute (60) by the number of minutes in an hour (also 60).

**Ans**: 60 \* 60

1. Assign the result from the previous task (seconds in an hour) to a variable called seconds\_per\_hour.

**Ans**: seconds\_per\_hour = 60 \* 60

1. How many seconds do you think there are in a day? Make use of the variables seconds per hour and minutes per hour.

**Ans**: minutes\_per\_hour = 60

seconds\_per\_day = seconds\_per\_hour \* minutes\_per\_hour \* 24

1. Calculate seconds per day again, but this time save the result in a variable called seconds\_per\_day.

**Ans**: seconds\_per\_day = seconds\_per\_hour \* minutes\_per\_hour \* 24

1. Divide seconds\_per\_day by seconds\_per\_hour. Use floating-point (/) division.

**Ans**: seconds\_per\_day / seconds\_per\_hour

1. Divide seconds\_per\_day by seconds\_per\_hour, using integer (//) division. Did this number agree with the floating-point value from the previous question, aside from the final .0?

**Ans**: seconds\_per\_day // seconds\_per\_hour

The result from integer division should be the same as the floating-point division but without the decimal part.

1. Write a generator, genPrimes, that returns the sequence of prime numbers on successive calls to its next() method: 2, 3, 5, 7, 11, ...

**Ans**: def genPrimes():

primes = []

num = 2

while True:

for p in primes:

if num % p == 0:

break

else:

primes.append(num)

yield num

num += 1

primes\_generator = genPrimes()

print(next(primes\_generator))

print(next(primes\_generator))

print(next(primes\_generator))