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).

* seconds\_per\_min= 60
* minutes\_per\_hour = 60
* print(seconds\_per\_min \* minutes\_per\_hour)

3600

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

* seconds\_per\_hour = seconds\_per\_min \* minutes\_per\_hour
* print(seconds\_per\_hour)

3600

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

* Hours\_per\_day = 24
* print( Hours\_per\_day \* minutes\_per\_hour \*seconds\_per\_min)

86400

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

* seconds\_per\_day = Hours\_per\_day \* minutes\_per\_hour \*seconds\_per\_min
* print(seconds\_per\_day)

86400

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

* print(seconds\_per\_day/seconds\_per\_hour)

24.0

6. 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?

* print(seconds\_per\_day/seconds\_per\_hour)

24, yes it’s same as earlier

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

def genPrimes():

primes = [] # primes generated so far

last = 1 # last number tried

while True:

last += 1

for p in primes:

if last % p == 0:

break

else:

primes.append(last)

yield last

list(genPrimes())