**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 seconds/minute \* 60 minutes/hour = 3600 seconds/hour.

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

Ans: 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.**

Ans: seconds\_in\_a\_day = seconds\_per\_hour \* 24

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

Ans: seconds\_per\_day = seconds\_in\_a\_day

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

Ans: floating\_point\_result = seconds\_per\_day / seconds\_per\_hour

print(floating\_point\_result)

**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?**

Ans: integer\_division\_result = seconds\_per\_day // seconds\_per\_hour

print(integer\_division\_result)

**7. 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 is\_prime(n):

if n <= 1:

return False

if n <= 3:

return True

if n % 2 == 0 or n % 3 == 0:

return False

i = 5

while i \* i <= n:

if n % i == 0 or n % (i + 2) == 0:

return False

i += 6

return True

def genPrimes():

num = 2

while True:

if is\_prime(num):

yield num

num += 1

# Example usage of the genPrimes generator

prime\_generator = genPrimes()

for \_ in range(10):

print(next(prime\_generator))