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).

sol. 60

>>> 60 \* 60

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

seconds\_per\_hour = 60 \* 60

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

minutes\_per\_hour = 60

seconds\_per\_hour = 60 \* 60

hours\_per\_day = 24

seconds\_per\_day = seconds\_per\_hour \* hours\_per\_day

print(seconds\_per\_day)

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

minutes\_per\_hour = 60

seconds\_per\_hour = 60 \* 60 # or 3600

hours\_per\_day = 24

seconds\_per\_day = seconds\_per\_hour \* hours\_per\_day

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

Seconds\_per\_day/seconds\_per\_hour

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?

Seconds\_per\_day//seconds\_per\_hour

yes

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 = [] # List to store prime numbers

candidate = 2 # Starting number to check for primality

while True:

is\_prime = True

for prime in primes:

if candidate % prime == 0:

is\_prime = False

break

if is\_prime:

primes.append(candidate)

yield candidate

candidate += 1

# Example usage:

prime\_gen = genPrimes()

for \_ in range(10): # Print the first 10 prime numbers

print(next(prime\_gen))