**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).**

Answer 1:

#Calculate the number of seconds in an hour

seconds\_in\_minute = 60

minutes\_in\_hour = 60

#Multiply the number of seconds in a minute by the number of minutes in an hour

seconds\_in\_hour = seconds\_in\_minute \* minutes\_in\_hour

#Print the result

print(seconds\_in\_hour)

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

Answer 2: seconds\_per\_hour = seconds\_in\_hour

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

Answer 3:

minutes\_in\_day = 24 \* minutes\_in\_hour

seconds\_per\_day\_guess = minutes\_in\_day \* seconds\_in\_minute

print(seconds\_per\_day\_guess) # Output: 86400

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

Answer 4: seconds\_per\_day = seconds\_per\_hour \* 24

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

Answer 5:

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

print(floating\_point\_result) # Output: 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?**

Answer 6:

integer\_result = seconds\_per\_day // seconds\_per\_hour

print(integer\_result) #Output: 24

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

Answer 7:

def genPrimes():

primes = [2]

yield 2

current\_number = 3

while True:

if all(current\_number % prime != 0 for prime in primes):

primes.append(current\_number)

yield current\_number

current\_number += 2

#Example of using genPrimes

prime\_generator = genPrimes()

for \_ in range(5): # Print the first 5 prime numbers

print(next(prime\_generator))