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 seconds/minute \* 60 minutes/hour

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

Sol 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.

Sol # Given data

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

hours\_per\_day = 24

# Calculate seconds in a day

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

Sol # Given data

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

hours\_per\_day = 24

# Calculate seconds in a day and save the result in a variable

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

# Print the result

print(seconds\_per\_day)

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

Sol # Given data

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

hours\_per\_day = 24

# Calculate seconds in a day and save the result in a variable

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

# Divide seconds\_per\_day by seconds\_per\_hour using floating-point division

result = seconds\_per\_day / seconds\_per\_hour

# Print the result

print(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?

Sol # Given data

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

hours\_per\_day = 24

# Calculate seconds in a day and save the result in a variable

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

# Divide seconds\_per\_day by seconds\_per\_hour using integer division

result\_integer\_division = seconds\_per\_day // seconds\_per\_hour

# Print the result

print(result\_integer\_division)

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

Sol def is\_prime(num):

if num < 2:

return False

for i in range(2, int(num \*\* 0.5) + 1):

if num % i == 0:

return False

return True

def genPrimes():

num = 2

while True:

if is\_prime(num):

yield num

num += 1

# Create a generator instance

prime\_generator = genPrimes()

# Print the first 10 prime numbers

for \_ in range(10):

print(next(prime\_generator))