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).

# Step 1: Calculate the number of seconds in an hour

seconds\_in\_minute = 60

minutes\_in\_hour = 60

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

print("Seconds in an hour:", seconds\_in\_hour)

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

# Step 2: Assign the result to a variable called seconds\_per\_hour

seconds\_per\_hour = seconds\_in\_hour

print("Seconds per hour:", seconds\_per\_hour)

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

# Step 3: Calculate the number of seconds in a day

hours\_per\_day = 24

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

print("Seconds in a day:", seconds\_in\_day)

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

# Step 4: Save the result in a variable called seconds\_per\_day

seconds\_per\_day = seconds\_in\_day

print("Seconds per day:", seconds\_per\_day)

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

# Step 5: Floating-point division to find hours in a day

hours\_in\_day\_float = seconds\_per\_day / seconds\_per\_hour

print("Hours in a day (float division):", hours\_in\_day\_float)

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?

# Step 5: Floating-point division to find hours in a day

hours\_in\_day\_float = seconds\_per\_day / seconds\_per\_hour

print("Hours in a day (float division):", hours\_in\_day\_float)

# Step 6: Integer division to find hours in a day

hours\_in\_day\_int = seconds\_per\_day // seconds\_per\_hour

print("Hours in a day (integer division):", hours\_in\_day\_int)

# Interpretation

print("Does the integer division match the floating-point division? ", hours\_in\_day\_float == hours\_in\_day\_int)

# Step 1: Calculate the number of seconds in an hour

seconds\_in\_minute = 60

minutes\_in\_hour = 60

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

print("Seconds in an hour:", seconds\_in\_hour)

# Step 2: Assign the result to a variable called seconds\_per\_hour

seconds\_per\_hour = seconds\_in\_hour

print("Seconds per hour:", seconds\_per\_hour)

# Step 3: Calculate the number of seconds in a day

hours\_per\_day = 24

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

print("Seconds in a day:", seconds\_in\_day)

# Step 4: Save the result in a variable called seconds\_per\_day

seconds\_per\_day = seconds\_in\_day

print("Seconds per day:", seconds\_per\_day)

# Step 5: Floating-point division to find hours in a day

hours\_in\_day\_float = seconds\_per\_day / seconds\_per\_hour

print("Hours in a day (float division):", hours\_in\_day\_float)

# Step 6: Integer division to find hours in a day

hours\_in\_day\_int = seconds\_per\_day // seconds\_per\_hour

print("Hours in a day (integer division):", hours\_in\_day\_int)

# Interpretation

print("Does the integer division match the floating-point division? ", hours\_in\_day\_float == hours\_in\_day\_int)

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 is\_prime(n):

if n < 2:

return False

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

if n % i == 0:

return False

return True

def genPrimes():

num = 2

while True:

if is\_prime(num):

yield num

num += 1

# Example usage:

prime\_gen = genPrimes()

for \_ in range(5):

print(next(prime\_gen))