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).

🡪 number\_of\_seconds\_in\_a\_minute = 60

number\_of\_minutes\_in\_an\_hour = 60

number\_of\_seconds\_in\_a\_minute \* number\_of\_minutes\_in\_an\_hour

output is 3600

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

🡪 seconds\_per\_hour = number\_of\_seconds\_in\_a\_minute \* number\_of\_minutes\_in\_an\_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.

🡪 seconds\_per\_hour \*24

Output is 86400

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

🡪 seconds\_per\_day = seconds\_per\_hour \*24

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

🡪 seconds\_per\_day/seconds\_per\_hour

Output is 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?

🡪seconds\_per\_day/seconds\_per\_hour

Ouput is 24

Yes, both the floating point (/) division and integer (//) division gave the same result aside from the final .0

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():

while True:

try:

start = int(input("Please enter lower range: "))

end = int(input("Please enter upper range: "))

except Exception as e:

print(e)

print("Sorry! This is invalid input, please enter a integer")

else:

print("Entered correct value")

for num in range(start,end+1):

if num>1:

for i in range(2,num):

if (num%i == 0):

break

else:

yield num

x = genPrimes()

next(x) 