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).

Minutes\_in\_hour=60

Seconds\_in\_Minutes=60

sec\_in\_an\_Hour= Minutes\_in\_hour \* Seconds\_in\_Hour

sec\_in\_an\_Hour

Output: 3600

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

Seconds\_per\_hour=sec\_in\_an\_hour

Output:

3600

1. 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\_day= seconds\_per\_hour \* 24

OR

Seconds\_per\_day = minutes\_in\_hour \* 24 \*seconds\_in\_minutes

Output:

86400

1. 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

seconds\_per\_day=Seconds\_per\_day

output:

86400

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

seconds\_per\_day / Seconds\_per\_hour

Output:

24.0

1. 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

output:

24

Yes this value agrees with the floating point value

1. 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():

prime = [2]

yield prime [0]

guess = 3

while True:

if (guess % x != 0 for x in primes):

prime.append(guess)

if guess == prime [-1]:

yield prime [-1]

guess +=2