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

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

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

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

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?

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

**Solution: 1**

print(60\*60)

**Solution: 2**

seconds\_per\_hour = 60\*60

print(seconds\_per\_hour)

**Solution: 3**

minutes\_per\_hour = 60

print(seconds\_per\_hour\*24)

**Solution: 4**

seconds\_per\_day = 24\*60\*60

print(seconds\_per\_day)

**Solution: 5**

print(seconds\_per\_day/seconds\_per\_hour)

**Solution: 6**

print(seconds\_per\_day//seconds\_per\_hour, end='')

print(' -> yes this values agree with the floating point value from the previous question')

**Solution: 7**

def genPrimes():

n = 0

while True:

if n == 2 or n == 3 :

yield n

elif ((n-1)%6 == 0 or (n+1)%6 == 0) and n !=1:

yield n

n = n+1

output = genPrimes()

for ele in range(5):

print(next(output))