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

**Ans:** 60\*60\*60

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

**Ans:** seconds\_per\_hour=60\*60\*60

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

**Ans:**

24\* seconds\_per\_hour

24\* minutes\_per\_hour\*60

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

**Ans:** seconds\_per\_day=24\* seconds\_per\_hour

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

**Ans:** seconds\_per\_day/seconds\_per\_hour

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?

**Ans:** seconds\_per\_day//seconds\_per\_hour, yes value is agree with floating point from previous question aide from 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, ...

**Ans:**

def genPrimes(upper\_limit):

for n in range(2, upper\_limit+1):

count =0

for j in range(2, n//2+1):

if n%j==0:

count+=1

if count==0:

yield n

gen=genPrimes(100)

print(next(gen))

print(next(gen))

print(next(gen))

print(next(gen))

print(next(gen))

print(next(gen))

print(next(gen))

print(next(gen