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).

**Solution : 3600**

**min\_per\_hour= 60**

**sec\_per\_min = 60**

**hour\_per\_day =24**

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

**Solution : seconds\_per\_hour = min\_per\_hour \* sec\_per\_min**

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

**Solution : 86400**

**seconds\_per\_day= ( min\_per\_hour \* sec\_per\_min) \* hour\_per\_day**

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

**Solution :**

**seconds\_per\_day= ( min\_per\_hour \* sec\_per\_min) \* hour\_per\_day**

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

**Solution : 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?

**Solution : Yes. 24**

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 :**

**def gen\_primes():**

**n = 2**

**primes = set()**

**while True:**

**for p in primes:**

**if n % p == 0:**

**break**

**else:**

**primes.add(n)**

**yield n**

**n += 1**

**g=gen\_primes()**

**for i in g :**

**print(str(i) +", ")**