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).  
Ans: 3600 seconds are there in an hour.   
minutest\_in\_hour = 60  
seconds\_in\_minute = 60  
print(f”seconds in an hour is {minutest\_in\_hour \* seconds\_in\_minute}”)

2. Assign the result from the previous task (seconds in an hour) to a variable called seconds\_per\_hour.  
Ans:   
minutest\_in\_hour = 60  
seconds\_in\_minute = 60  
seconds\_in\_hour = minutest\_in\_hour \* seconds\_in\_minute  
print(f”seconds in an hour is {seconds\_in\_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.  
Ans:   
minutest\_in\_hour = 60  
seconds\_in\_minute = 60  
seconds\_in\_hour = minutest\_in\_hour \* seconds\_in\_minute  
print(f”seconds in a day is {seconds\_in\_hour \* 24}”)

4. Calculate seconds per day again, but this time save the result in a variable called seconds\_per\_day  
Ans:  
minutest\_in\_hour = 60  
seconds\_in\_minute = 60  
seconds\_in\_hour = minutest\_in\_hour \* seconds\_in\_minute  
second\_in\_day = seconds\_in\_hour \* 24  
print(f”seconds in a day is {second\_in\_day}”)

5. Divide seconds\_per\_day by seconds\_per\_hour. Use floating-point (/) division.  
Ans:   
seconds\_per\_day = 24\*60\*60  
seconds\_in\_hour = 60\*60  
print(seconds\_per\_day/seconds\_in\_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: Yes, this number agree with the floating point value from pervious question.  
seconds\_per\_day = 24\*60\*60  
seconds\_in\_hour = 60\*60  
print(seconds\_per\_day//seconds\_in\_hour)

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(n):  
 prime\_lst = []  
 num = 2  
 while num <= n:  
 for i in range(2, num):  
 if num % i == 0:  
 break  
 else:  
 yield num  
 num += 1

prime\_numbers = genPrimes(19)

for number in prime\_numbers  
 print(number)