**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**

def seconds\_in\_an\_hour():

'return numbers of seconds in given hours'

hours = float(input('please type numbers of hours: '))

seconds\_in\_hours = hours\*seconds\_per\_minute\*minutes\_per\_hour

return seconds\_in\_hours

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

seconds\_per\_hour = seconds\_in\_an\_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.**

24\*minutes\_per\_hour\*seconds\_per\_minute

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

seconds\_per\_day = 24\*minutes\_per\_hour\*seconds\_per\_minute

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

seconds\_per\_day/seconds\_per\_hour = 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?**

seconds\_per\_day//seconds\_per\_hour = 24.0

It is sama as floating point division because denominator is perfectly divisible by numerator.

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

def prime\_nos(n):

' prime number genetaor '

for i in range(2,n+1):

for j in range(2,i):

if i % j ==0:

break

else:

yield i