PYTHON ASSIGNMENT - 15

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

>>> 60 \* 60

3600

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

seconds\_per\_hour = 3600

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

and minutes per hour.

>>>seconds\_per\_hour \* 24

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

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

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

hours\_per\_day\_float = 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?

hours\_per\_day\_int = seconds\_per\_day // seconds\_per\_hour

print(hours\_per\_day\_float) # Output: 24.0

print(hours\_per\_day\_int) # Output: 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, ...

def genPrimes():

primes = []

candidate = 2

while True:

is\_prime = all(candidate % p != 0 for p in primes)

if is\_prime:

primes.append(candidate)

yield candidate

candidate += 1

# Example usage:

prime\_generator = genPrimes()

print(next(prime\_generator)) # Output: 2

print(next(prime\_generator)) # Output: 3

print(next(prime\_generator)) # Output: 5

print(next(prime\_generator)) # Output: 7

print(next(prime\_generator)) # Output: 11