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

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

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

5. Divide seconds_per_day by seconds_per_hour. Use floating-point (/) division.

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?

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:-
# Task 1 & 2: Calculate and assign seconds in an hour
seconds_per_hour = 60 * 60
print(f"Seconds per hour: {seconds_per_hour}")

# Task 3 & 4: Calculate and assign seconds in a day
seconds_per_day = seconds_per_hour * 24
print(f"Seconds per day: {seconds_per_day}")

# Task 5: Floating-point division of seconds_per_day by seconds_per_hour
hours_in_day = seconds_per_day / seconds_per_hour
print(f"Hours in a day (floating-point division): {hours_in_day}")

# Task 6: Integer division of seconds_per_day by seconds_per_hour
hours_in_day_int = seconds_per_day // seconds_per_hour
print(f"Hours in a day (integer division): {hours_in_day_int}")

# Task 7: Generator for prime numbers
def genPrimes():
    primes = []   # primes generated so far
    last = 1      # last number tried
    while True:
        last += 1
        for p in primes:
            if last % p == 0:
                break
        else:
            primes.append(last)
            yield last

# Using the generator
prime_gen = genPrimes()
for _ in range(10):
    print(next(prime_gen), end=' ')
```