Day 2++ Python Exercises — Intermediate Challenge

1. From two lists `a` and `b`, create a list of elements that are present in only one of the lists (i.e. not shared) *but preserve the original order*. Do *not* use sets.

2. Make a list of the squares of even numbers from 1 to 100, but only include those whose square ends with an even digit. Use a single list comprehension.

3. Write a list comprehension that returns tuples `(x, x**2, x**3)` for all numbers `x` from 1 to 30, but only include them if `x` is divisible by either 3 or 4 but not both.

4. Write a function that takes a list of numbers and returns a list of primes whose digits sum to an odd number.

5. You have a list of sentences. Use list comprehension and string methods to return a list of the number of words *only in those sentences* that contain the word "data".

6. Create a new dictionary from an existing one with only the items where the *value is a float* and round them to 1 decimal place.

7. Using `map()` and a lambda, return a list of booleans where each boolean indicates whether the corresponding string in a list is a palindrome.

8. You have a list of full names in format `'Firstname Lastname'`. Use `map()` to convert them into email-like format: `'firstname.lastname@email.com'`.

9. Use `filter()` and a lambda to remove all sublists from a list of lists that contain negative numbers.

10. Given a list of student dictionaries (with 'name', 'math', 'history', 'science' scores), return a list of names of students whose *average score* is above 75.

11. From a dictionary of items and their weights in grams, filter out items whose weight is less than the *median* weight. Use dictionary comprehension.

12. Given a temperature log (dict of date: temp), create a new dict containing only entries where the temperature increased compared to the previous day.

13. Write a function that finds the most frequent pair of consecutive letters in a string. Ignore spaces and punctuation.

14. **You are given a dictionary of users and their last login timestamps. Remove any user who hasn't logged in the last 30 days. Assume current time is `now = datetime.now()` and timestamps are `datetime` objects.**

15. **Given a nested dictionary of product categories and product prices, flatten it into a single dictionary with keys as `category_product` and values as the price.**

```
products = {
    "fruits": {"apple": 1.5, "banana": 1.2},
    "drinks": {"water": 0.5, "soda": 1.1}
}
# → {'fruits_apple': 1.5, 'fruits_banana': 1.2, ...}
```

16. **Create a dictionary where the key is a lowercase letter and the value is how many times it appears in a given string (ignore case, skip non-alpha).**

17. **Use dictionary comprehension to invert a dictionary, *but group together keys that share the same value* into a list.**

```
# {'a': 1, 'b': 2, 'c': 1} → {1: ['a', 'c'], 2: ['b']}
```

18. **Given two sets, return a list of elements that are in exactly *one* of the sets but also divisible by 3 or 7.**

19. **Write a function that takes a list of numbers and returns the maximum *product* of any 3 elements. Do not sort the list.**

20. **Create a generator function that yields squares of numbers from 1 to n, but skips numbers whose square ends with 9.**