Coding Challenges Rookies

Exercise 1: Palindrome 20 min

Write a method to check if a given String is a palindrome. The method should be case insensitive, and spaces should be ignored.

Bonus: Enhance the solution to be able to check only subparts of the string to be a palindrome with a left and right index border.

Exercise 2: Anagram 30 min

Write a method to check if two given Strings (case insensitive) are an anagram (containing the same chars with the same frequencies), for example:

- "lamp" and "palm" => a x 1, l x 1, m x 1, p x 1
- "Otto" and "Toto" => o x 2, t x 2
- "pairs" and "Paris" => a x 1, i x 1, p x 1, r x 1, l x 1
- "silent" and "listen"
- "creative" and "reactive"

Bonus: Enhance the solution for a list of Strings like `List.of("Ampel", "Palme", "Maple")`. Think of reusing your original solution and comparing it in pairs. Find another possibility of converting strings into suitable character histograms using `HashMap`.

Exercise 3: Divisors 15 min

Write a method to calculate all proper divisors of a given number (`int`) and return them as `List<Integer>` (all real divisors without the number itself). As a special case for one, the list should contain 1:

- 4 => [1, 2]
- 6 => [1, 2, 3]
- 20 => [1, 2, 4, 5, 10]
- 70 => [1, 2, 5, 7, 10, 14, 35]

Exercise 4: Prime numbers >30 min

Write a method to calculate all prime numbers up to a maximum (`int`). Use the algorithm Sieve of Eratosthenes as described below.

The algorithm for determining prime numbers up to a given maximum value, named Sieve of Eratosthenes, dates back to the Greek mathematician with the same name. It uses the following steps:

1. Initially, all numbers from two to the maximum value are written down, for example:

   $$2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$$

2. Gradually eliminate those numbers that cannot be prime numbers.
   1. The smallest unmarked number, here 2, corresponds to the first prime number. Now cross out all multiples of it, i.e., in the example 4, 6, 8, 10, 12, 14:

      $$2, 3, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}, 9, \cancel{10}, 11, \cancel{12}, 13, \cancel{14}, 15$$

   2. We continue with the number 3, which is the second prime number. Now the multiples are crossed out again, in the following 6, 9, 12, 15:

      $$2, 3, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}, \cancel{9}, \cancel{10}, 11, \cancel{12}, 13, \cancel{14}, \cancel{15}$$

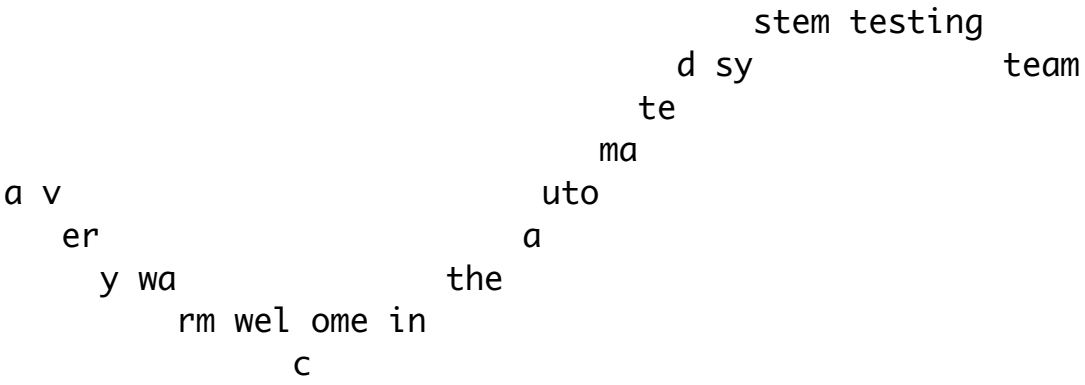
   3. The next unmarked number and thus a prime number is 5. The procedure is repeated as long as there are still unmarked numbers after the current prime number:

      $$2, 3, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}, \cancel{9}, \cancel{10}, 11, \cancel{12}, 13, \cancel{14}, \cancel{15}$$

Exercise 5: Zick Zack and Sine-/Cosine-Strings 30 min

Write a method to output a string as zick zack or with sine-/cosine orientation as follows:

```
a         o         a         i
      c m       m t       t n
  v     l   e     o   e     s   g
   e     e         t   d     e
    r     w     i     u       t     t
     y         n   a     s         e
         m                   y   m     a
       w r         t e         s e       m
        a           h           t
```



Exercise 6: Magic Triangle 30 min

Write a method `boolean isMagicTriangle(List<Integer>)` that checks whether a sequence of numbers forms a magic triangle. Such a triangle is defined as one where the respective sums of the three sides' values must all be equal.

The following shows a negative example and positive cases for one triangle each of side length three and side length four:

This results in the following sides and sums:

input	values 1	values 2
side 1	1 + 5 + 3 = 9	2 + 5 + 9 + 1 = 17
side 2	3 + 4 + 2 = 9	1 + 6 + 7 + 3 = 17
side 3	2 + 6 + 1 = 9	3 + 4 + 8 + 2 = 17

Hint: Represent the individual sides of the triangle as sublists and use the `List<E> subList(int startInclusive, int endExclusive)` method to extract the respective sides.

General Advice

Consider testing and, more importantly, start thinking about the problem and get a feeling *before* coding.