Exercises marked with a ⋆ can be handed in for bonus points. Due date is April 27.

Exercise 1
Recall the Lemma of the lecture which states that
\[ \text{ord}_p(n!) = \sum_{i=1}^{\infty} \left\lfloor \frac{n}{p^i} \right\rfloor = \frac{n - S_p(n)}{p - 1} \]
where \( S_p(n) \) is the sum of the digits of \( n \) written in base \( p \). Prove the second equality.

Exercise 2 (⋆)
Determine the number of lines (in \( \Theta \)-notation) that the following algorithm prints.

```
SPAM(n)
1 for i ← 1…n
2 do Print a line “i/n”
3 if n > 1
4 then SPAM(n/2)
5 SPAM(n/2)
```

Exercise 3
Let \( R \) be a ring, and \( \omega \in R \) be a primitive \( n \)-th root of unity. Show:

1. \( \omega^{-1} \) is a primitive \( n \)-th root of unity.
2. If \( n \) is even, then \( \omega^2 \) is a primitive \((n/2)\)-th root of unity. If \( n \) is odd, then \( \omega^2 \) is a primitive \( n \)-th root of unity.
3. Let \( k \in \mathbb{Z} \) and \( d = n/\gcd(n,k) \). Then \( \omega^k \) is a \( d \)-th root of unity.
4. Determine the number of primitive \( n \)-th roots of unity in \( \mathbb{C} \).

Exercise 4
Let \( n \in \mathbb{N} \). Show that 2 is a primitive \( 2n \)-th root of unity modulo \( 2^n + 1 \) if and only if \( n \) is a power of 2.
Exercise 5 (⋆)
Update to the latest version of the Subversion repository and find the new functionalities, in particular the placeholder `polynomial::multiply_fft` and `test_polynomial`. Implement multiplication of polynomials in $\mathbb{Z}[x]$ using FFT with modular arithmetic and make sure that the tests in `test_polynomial` run successfully.

Note: Use the new functions of `integer` for modular arithmetic.