Exercise 2
Number representation
(handed out: 16.04.2025 – hand in: 23.04.2025)

1. **Task** *Integer representation* (4 P)
   Understanding the representation in computers is not only essential but can also help to avoid common mistakes.
   a) When counting with ten fingers (digits!) and using the base of 2, what is the largest (decimal) number you can present? (1 P)
   b) Most computers don't use a sign bit for integer numbers (disadvantages of using a sign bit? 1 P), but use the *two's complement* (look it up!). What is then the bit pattern of the largest positive `int` $n_{\max}$ (32 bit integer)? What is the bit pattern and what is the value of $n_{\max} + 1$? (2 P)

2. **Task** *Floating point number representation* (3 P)
   The representation of floating point numbers is more complex and has many issues ...
   a) Complete the following code fragment:
```
float y, x = 1.043E-13 ;
x = x / 10. ;
y = x*x - (1.043E-14)*(1.043E-14) ;
```
   What is the expected/computed result for `y` and why is subtraction of floats potentially problematic? (1 P)
   b) What is the bit pattern representation of the following `float` numbers: `7`, `1.E-7`, and `7+1.E-7` and what is therefore the result of this summation? (2 P)

3. **Task** *GRS bits* (3 P + 2 BP)
   We want to use the example $1.000 \times 2^5 - 1.001 \times 2^1$ from the lecture to understand the advantage of GRS bits.
   a) Just as an exercise, convert the given binary operands to decimal numbers. (1P)
   b) Instead of infinite precision, use GRS bits for the calculation, so 3-bit mantissa + GRS bits. (2P)
   c) *Bonus:* Show that just having R+S bits wouldn't be enough, consider therefore also the calculation of $1.000 \times 2^5 - 1.111 \times 2^1$ for comparison. (2 Bonus P)

4. **Task** *Radius calculation* (3 P)
Write a C++ program for calculating the radius of a star from its given luminosity L_* and its given effective temperature T_{eff} as in

$$L_* = 4\,\pi\,\sigma_{\text{SB}}\,R_*^2\,T_{\text{eff}}^4\,. \tag{1}$$

The user should be asked for $\log(L_*/L_\odot)$ (luminosity in units of solar luminosity) and T_{eff}/K with help of `cout`. The user enters the values with `cin`:

```
cin >> logLsun ;
 ...
cin >> Teff ;
```

and obtains in return the stellar radius in $R_\odot$ (solar radii).
Hints: Which libraries do you need (`#include <...>`)?
You don't need the value of σ_{SB}, but only the effective temperature of the sun: $T_{\text{eff}} = 5778\,\text{K}$. (2 P) What is the numerical advantage of doing so? (1 P)
E.g., which radius has a white dwarf of $T_{\text{eff}} = 144\,\text{kK}$ and $\log(L_*/L_\odot) = 3.8$? (Answer: $R = 0.13\,R_\odot$).