Sheet 1

2023/09/19

1. Write a function `string_reverse` that takes a `&str` as input and returns it, reversed as a `String`;
2. Write a function `bigger` that takes two `i32` and returns the bigger number (`i32`) without using another function call and additional variables;
3. Write a function `multiply` that takes an `i32`, a `f32` and a `f64` and returns the multiplication of the three of them as a `f64` value;
4. Write a function `e_equals_mc_squared` that takes as input a `f32` representing the mass, and that uses a **globally-defined** constant containing the value of the speed of light in a vacuum (expressed in m/s). The function outputs the energy equivalent to the mass input;
5. Given a vector of `i32`, create a function `max_min` that returns the maximum and the minimum value inside that vector;
6. Write a function `lord_farquaad` that takes a `String` and outputs another `String` in which every character 'e' is substituted by the character '💥';
7. In the main function initialize a `HashMap<String, f32>` called `furniture` that stores the pair `String` as key and `f32` as value, where the `String` is the name of the furniture and the `f32` is its price. Then write a function that borrows the `HashMap`, takes a `furniture: String` as input and returns the corresponding `f32`. If there is no such furniture in the `HashMap`, return `-1.0`;
8. Write a function `append` that takes a `String`, appends the word "foobar" to it and returns it.
   Then write a `main` function in which you:
   - Declare a `String` initialized with some text.;
   - Pass the `String` to the function `append`;
   - Print the original `String` and the one returned by `append`;
   (do it in this order!)
9. An Armstrong number is a number that is the sum of its own digits each raised to the power of the number of digits.
   For example:
   - `9` is an Armstrong number, because $9 = 9^1 = 9$
   - `10` is not an Armstrong number, because $10 \neq 1^2 + 0^2 = 1$
   - `153` is an Armstrong number, because: $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$
   - `154` is not an Armstrong number, because: $154 \neq 1^3 + 5^3 + 4^3 = 1 + 125 + 64 = 190$
   Write the function `is_armstrong` that determines whether a number is an Armstrong number;
10. Write a function that takes a 2x2 i32 "matrix" (2x2 tuple) as input, transposes and returns it.