Exercise 1: Euclid's Algorithm

The following is the pseudocode for Euclid's GCD algorithm:

Algorithm 1 *Euclid*($m : \mathbb{N}$, $n : \mathbb{N}$)

```
if m = 0 then
    return n
else if n = 0 then
    return m
else if m ≤ n then
    return EUCLID(m, n mod m)
else
    return EUCLID(m mod n, n)
end if
```

(a) Write out a table of the values of m and n, as well as the ultimate action (return statement) taken by the algorithm, for the inputs $m = 270$ and $n = 192$:

m	n	Action
270	192	**return** EUCLID(78, 192)
...	...	...

(b) Write this algorithm in R.

Exercise 2: Palindrome

A "palindrome" is a word or sentence that reads the same forwards and backwards, for example "radar".

Write the pseudocode for an algorithm that checks if an array of letters $A[1 \dots n]$ (e.g. $A = [\text{'r'}, \text{'a'}, \text{'d'}, \text{'a'}, \text{'r'}]$) is a palindrome...

(a) ... using any of the pseudocode constructs from the lecture (but without recursion).

(b) ... using none of the loop constructs (no **for** or **while**), by calling your algorithm recursively. You will likely need to use the additional construct **del** $A[i]$, which deletes the i-th element from the array A and turns it from a length n array into a length $n-1$ array.

You can check for (in)equality of letters, e.g. $A[3] = A[7]$ or $A[3] \neq A[7]$, and can use standard arithmetic operations. You may find floor division: "$\lfloor x/y \rfloor$", yielding the greatest integer less than or equal to x/y, to be useful.

Exercise 3: Numeric Differentiation

The derivative of a function f at a point x is defined as:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Write an R function `derive(f, x0, h)` that computes the derivative of a given R-function `f` at a point `x0` for a given step size `h` by using the definition above.

Check your implementation by computing the derivative of the following functions and compare against the analytical solutions. What value of `h` is needed to get a good approximation of the derivative? What do you observe about the relative / abolute difference between the numeric and the analytical derivative?

$f(x)$	x_0	Analytical $f'(x)$
x^3	1	$3x^2$
$\cos(x^2)$	10^{-3}	$-2x\sin(x^2)$
$\sqrt{x}$	10^{-6}	$\frac{1}{2\sqrt{x}}$
$\sqrt{x-1000}$	$1000+10^{-6}$	$\frac{1}{2\sqrt{x-1000}}$

Exercise 4: Matrix Product

(a) Write a function that computes the product of two numeric matrices, `a` and `b`. Do not use R's built-in matrix multiplication functions and compute the product element-wise, instead.

(b) Compare the runtime performance of your function with R's built-in matrix multiplication function `a %*% b`: Perform a systematic benchmark for squared matrices of sizes $n \in \{2^1, 2^2, \ldots, 2^{12}\}$ (only up to 2^8 for your own function) and plot the results in a log-log plot. What do you observe? The `microbenchmark` package may help you here.