Exercise 1 (**Lotka–Volterra**) Type: **Matlab**

a) Implement the explicit Euler method in a m-file with the signature `[t,y] = EULER(FUNC,t0,tN,y0,N)`. The function should return the vector of time steps $t = (t_i)_{i=0,\ldots,N}$ and the corresponding approximate solutions $y = (y_i)_{i=0,\ldots,N}$. Test your implementation by writing another function `[f] = TESTFUNC(t,y)` that computes $f(t, y) = y^2 - t$.

b) Modify your file of a) such that

- it accepts an additional parameter `var` that can pass further variables to the ODE (like `[f] = TESTFUNC(t,y,var)`).
- it can solve systems of ODEs where the unknown $y \in \mathbb{R}^d$ is a column vector.

c) Implement the Lotka–Volterra ODE which is given by the following system for the unknown functions $y_1 = y_1(t)$, $y_2 = y_2(t)$ (as in the lecture):

$$\dot{y}_1 = c_1\, y_1\, (1 - d_1\, y_2),$$
$$\dot{y}_2 = c_2\, y_2\, (d_2\, y_1 - 1).$$

The file should have the signature `[f] = LOTKAODE(t,y,var)` where `var` contains the positive constants c_1, c_2, d_1 and d_2.

d) Download the file `lotka_euler.m` from the web page and test your results of b) and c) with it.

e) Implement the method of Heun in a m-file `[t,y] = HEUN(FUNC,t0,tN,y0,N,var)`.

f) Download `lotka_comp.m` from the web page and compare the quality of the results of the explicit Euler method and the method of Heun (e.g. with respect to the number of function evaluations per step).