If you are using a jupyter notebook (recommended), then keep all your programs in a single notebook. A good programming style is to define a function for one task with clearly defined input (arguments) and output. For plots you may use matplotlib (if you are using python) or gnuplot (if you are using c or fortran) or LsqFit module if you are using Julia.

---

If you are planning to submit separate programs, then please follow the guideline below:

- Keep all files of a worksheet in a single folder.
- Follow a systematic naming convention. You may name the program files as `Q1.py` or `Q1a.py`, `Q1b.py` for question 1 (if you have created multiple files for a single question). The data file should be named as `Q1-data-a.dat` and so on.
- Finally compress the entire folder as a single `.zip` or `.tgz` (using `tar cvfz archive.tgz folder-name/`, and submit the file in WeLearn.

---

---

1. (20 points) The Van der Pol oscillator is a classic example of a system with non-linear damping. It is defined by the equation

$$\frac{d^2y}{dx} = \mu(1-y^2)\frac{dy}{dx} - \lambda y.$$

where, $\mu$ is the damping coefficient and has a strong influence on the nature of the dynamics. $\lambda$ is the square of the oscillator freqency. For $\mu = 0$, it is a regular simple harmonic oscillator. For $\mu > 0$, it starts displaying a limit cycle behavior. For $\mu = 5$, it shows periodic slow and sharp changes in $y$. Naturally, such systems benefit from the use of adaptive time stepping. In this example, you need to solve the above equation for $\mu = 5$, $\lambda = 40$, and from $t = 0.0$ to $t = 20$. The initial condition is $y(0) = 0.5$ and $y'(0) = 0.0$.

   (a) (8 points) Use regular RK4 to solve the equation with small time steps $h = 10^{-4}$.

   (b) (9+1 points) Use the Dormand Prince adaptive method. How many time steps does it require?

   (c) (2 points) Plot the phase space trajectory of the oscillator ($y'$ versus $y$).