Using a Python Script in Spyder:

1. **Explore and Read the Data File**
   Load the attached dataset file, **"data_mass_raw.txt"**, into the script. Store its contents in appropriate NumPy variables.
   Hint: This file contains time-series data (in days) of a mass-related quantity, referred to as x.
2. **Data Analysis**
   - Determine the number of time data points.
   - Display the following in the console:
     - Number of data points.
     - Minimum and maximum values of x.
     - The final time value (ensuring it is less than 500 days).
3. **Data Visualization**
   Create a well-designed 1D plot to illustrate the data described above.
4. **Model Implementation**
   Analyze the data using the given model:

$$\frac{dx}{dt} = a \cdot x \cdot \ln\left(\frac{K}{x}\right)$$

- Start with the constants a=0.07 and K=700.
- Numerically solve x(t)x(t)x(t) using the algorithm :

$$x_{k+1} = x_k + \Delta t \cdot a \cdot x_k \cdot \ln\left(\frac{K}{x_k}\right)$$

- • Parameters:
  - Time step: Δt=0.05.
  - Initial value: x0=16.
- Compute x(t) values until the time matches the last point in the dataset.
- Overlay the approximation x(t) on the known data in a graph.

5. **Interpolation**

- With thousands of computed x values, use Scipy to interpolate them onto a less refined grid matching the dataset's time points.
- Ensure the interpolation creates a smooth curve with second derivative continuity.
- Plot the original data, the computed approximation, and the interpolated curve for comparison.

6. **Solver Fonction**

- Implement a function `mon_solver` that automates the numerical solution:
  - Inputs: a, K, x0, Δt.
  - Outputs: Approximated x values sampled at the dataset's time points.
- Test the function with the parameters a=0.07, K=700, x0=16, Δt=0.05.
- Plot the original data and the function's result for verification.