Write a matlab script that uses Euler’s method to integrate the following initial value problem (IVP) from $t = 0$ to $t = 1$: 

$$y' = 2y - 1, \quad y(0) = 1.$$ 

Your script should include the following features:

- You should be able to specify the number of steps $N$ from $t = 0$ to $t = 1$, from which your script calculates the step size $h = 1/N$.
- Your script should store the values of $t_n$ and the numerically obtained values of $y_n$ for $n = 1, \ldots, N$ (note that $t_0 = 0$ and $t_N = 1$).
- Your script should allow you to input the exact solution of the initial value problem (IVP) for $y(t_n)$.
- Your script should calculate and store the cumulative error at each step, according to the formula $e_n = |y_n - y(t_n)|$.

Using your script, do the following:

1. Find the exact solution of the IVP and enter it into the appropriate part of your script.
2. Run your script for $N = 16$, $N = 32$, $N = 64$, etc., doubling the number of steps for each run up to $N = 8192$, in each case storing the final cumulative error value, $e_N$. Graph these error values versus the corresponding step sizes $h$. Deduce an approximate formula for how the cumulative error depends on $h$. Explain why we refer to Euler’s method as being *first-order accurate*.

3. Run the script with your smallest value of $h$ and, on the same axes, graph the numerically obtained solution using circles (‘o’) and the exact solution using a solid line.

**Format.** Your project must be typed on a computer, with the exception of equations, which may be written into blank spaces left in the typed text. All graphs must have a title and axis labels (type `doc plot` at the matlab command line for instructions). Remember to explain your results and to provide sufficient graphical evidence to support them. Include a printout of the script in your project.