Instruction sheet for Worksheet 1, read carefully please.

The zipped file contains 4 files, `WS1_a.py` ... `WS1_c.py` and `WS1_notebook.ipynb` file, along with the plots and the required tables.

- The `.py` files are the soluions for individual part of the worksheet problem.
- The Jupyter notebook file i.e `.ipynb` file contains the step-by-step explantion for each step and hence is suggested for evaluation.

Results and conslusion for the Worksheets

Part 1a:

*We found the **linear relationship** between the *Relative Error* and the step parameter **h**.*
The plot is in `error_vs_h_a.jpg` .

Part 1b:

We proved that the central difference method is exact for quadratic funcions.

The plot is `error_vs_h_b.jpg` .
In the plot, we can see that the error is almost zero for the central difference method. I also plotted the *log(Relative error)* to exxagerate the error. Also at some points the error is ~0, as we get runtime error for "divide by zero encountered in log10".
So we can say cen central difference method is exact for quadratic functions.

Part 1c:

- First we show the table of *Computed Numerical Second Derivative* for different values of x and step size *h*
- The table is also stored in the `.csv` file `df_table.csv` .
- We also saved the table for absolute and relative errors as `df_er_table.csv` and `df_rel_er_table.csv`
- To find the optimal step size we plloted the error: `error_vs_h.jpg` . **Note:** I took the log of the error to exxagerate the differences.
    - We observer that the error is the lowest for the 3 point from left which corresponds to ***h=1e-4***
- As reading the line plot may be tricky, we also plotted an image from the error array, where the pixel values is the error values: `error_vs_h_image.jpg` . We see that the error is the lowest for the 3rd column of the image which again corresponds to ***h=1e-4***.
- Thus we conclude that the optimal value of *h* is ***h=1e-4***