Introduction to Scientific Computing, PSCB57, Fall 2018
Assignment 2
Numpy Arrays and Matplotlib

Instructions

- You must submit the assignment electronically via Quercus. The deadline for this assignment is Monday, October 1st, 9am. Late assignments will not be accepted unless accompanied by supportive documentation.
- This assignment comes in multiple parts. Submit all your answers in one Jupyter Notebook file with the file type `ipynb`.
- Use mark-down cells to add your name, student number. Also use mark-down cells and python comments to describe your code! Well documented code might help you with the quiz.
- Do not use any packages or libraries other than numpy and matplotlib in this assignment.
- You must be present at the tutorial on Tuesday where you will be quizzed about your assignment. If you do not show up or fail to pass the quiz, your assignment might be marked as 0% even if it was correct.
- Plagiarism is taken very seriously. However, you are not expected to work in solitude and are encouraged to talk to your classmates. But keep in mind that if you submit an assignment, you have to fully understand it in order to pass the quiz.

Part 1

In this part you will create two numpy arrays of length $N = 100$.

- Create a numpy array with the name `counter` and length N that contains all integer numbers from 0 to $N - 1$, i.e. 0,1,2,3,. . .

- Create a numpy array with the name `fib` and length N. Then fill the array with Fibonacci numbers starting with 0,1,1,2,. . .

Write your code so that you can easily change the length of the arrays by just changing one variable at a later time.

Part 2

In this part you will use the matplotlib library and the arrays from Part 1 to create two plots.

- Make a scatter plot where the `counter` array corresponds to the horizontal axis, the `fib` array corresponds to the vertical axis.

- Instead of plotting `fib` versus `counter`, now plot the natural logarithm of `fib` versus `counter`. Use a plotting style where the datapoints are connected by lines.

Make sure to add meaningful axis labels to all your plots. Also make sure that python does not output any error messages or warnings. If it does, then think of ways to avoid them, implement one of them, and describe in the notebook what you've done.