ME565 - Winter 2018 **Due: 11:59pm Friday February 23rd, 2018**
Homework 4

Exercise 4-1: [Modified from Trefethen's Spectral Methods in MATLAB]. Here, we will investigate the computational time required to compute an FFT in matlab. We will be using the commands `tic` and `toc`, which allow us to compute the amount of time a given piece of code takes to run. These are powerful tools to *benchmark* code.

First, compute a random vector of length $N = 2^k$ for each of the following integers $k = 2, 3, \ldots 16$. For each of these random vectors, compute the FFT and time how long it takes using `tic` and `toc`. Note: only put the FFT computation inside of the `tic` and `toc`; don't include the code that computes the random vector. Also, you will likely need to compute the FFT 1,000 times in a loop between `tic` and `toc` to get a good "average computation time", since it happens so rapidly.

Plot your computation time against N for each of the runs above. Please use the `loglog` plot and label your axis!

Next, perform the same computation (but with $k = 2, 3, \ldots 11$) using the discrete Fourier transform (DFT) matrix instead of the FFT. Again, don't put the computation of the DFT matrix inside `tic` and `toc`, just the step where you compute the Fourier transform signal. You can find code to generate the DFT matrix in the supplemental video lecture online.

Please plot your computation time agains N for the DFT matrix. Plot this on the same figure as the FFT computation, and use a different color.

Exercise 4-2: [Modified from Trefethen's Spectral Methods in MATLAB]. Using the same code from the problem above, compute the amount of time for the FFT computation for $N = 100, 101, 102, \ldots, 1000$. Again, plot the computation time against N for each run above, but now use the `plot` command. (plot in black)

Now, for each value of N, compute if this N is prime (using `isprime`). If the value is prime, then plot that data point as a red circle (`'or'`) on top of the original plot. Do you notice any patterns? Please explain.

Exercise 4-3: Load the image `recorder.jpg`. Convert to grayscale and compress the image using the FFT.

(a) Design a compression threshold to keep exactly 10% of the original Fourier coefficients. Compute the L_2 norm of the error between the new compressed image and the original image. Also compute the L_2 norm of the Fourier transformed versions of the compressed and original images.

(b) Repeat for a compression that only keeps 1% of the original Fourier coefficients.
