Assignments are due at the beginning of the due date. You should submit a printed hard copy of the report. Please upload the report and all the source code and executable on the class canvas website. In the report, you should explain your methods, some illustration figures of the results, as well as any remaining issues and potential solutions.

**Assignment 1: Histogram Equalization**

Part 1 (**Due October 03, 2017**): Write a program that conduct histogram equalization on a given image. Run your algorithm on the book example images of chapter 3: Figure 3.8, Figure 3.9, and Figure 3.16. You can try any other images as well.

---

**Matlab Tutorial**

- A short Matlab introduction
- Matlab tutorial
- The demos accessible by typing "demo" within Matlab.

After reading the tutorial, you should be comfortable doing the following tasks using an image of your choice:

1. Read an image into a variable
   ***Hint #1:*** "help imread"
   ***Hint #2:*** Use single quotes around the filename.
   ***Hint #3:*** Ending a command with a semicolon supresses printing the result.

2. Display the image
   ***Hint:*** "help image"

3. Adjust the axes so that the aspect ratio is 1:1 (i.e., "square pixels")
   ***Hint:*** "help axis"

4. Convert the image to grayscale
   ***Hint:*** "help rgb2gray"

5. Convert the grayscale image to floating point
   ***Hint:*** "help double"

6. Display the floating-point image, using a grayscale colormap
   ***Hint #1:*** "colormap(gray)"
   ***Hint #2:*** "help imagesc"

7. Plot the intensities along one scanline of the image
   ***Hint #1:*** Extracting a part of a matrix is done by

   ```
   matrix2 = matrix1(row_min:row_max,col_min:col_max);
   ```

   Indices in matlab are 1-based (not 0-based as in C). `row_max` or `col_max` may be "end" to indicate the last element.
   Just a ":" is equivalent to "1:end".
   ***Hint #2:*** "help plot"

8. Store the width and height of the image in variables "width" and "height"
   ***Hint #1:*** "help size"
   ***Hint #2:*** Functions in matlab may return multiple values. You can get at the values using the notation

   ```
   [var1, var2] = func(x)
   ```

   ***Hint #3:*** In matlab, the number of rows is the first dimension and the number of columns is the second.

9. Write a pair of nested "for" loops to set a grid of every 10th pixel horizontally and every 20th pixel vertically to 0
   ***Hint #1:*** "help for"
   ***Hint #2:*** The "start:increment:stop" notation

10. Create a function "maxrow" that takes a matrix and a row index and returns the brightest pixel in the given row. Store the function in a file called "maxrow.m" so that matlab loads it automatically when you call the function.
    ***Hint #1:*** "help function"
    ***Hint #2:*** Matlab has many built-in functions that operate on entire vectors or matrices. There might be one to compute the maximum...

11. Write the modified image back to a new file
    ***Hint #1:*** "help uint8" to convert the image back to integer
    ***Hint #2:*** "help imwrite"