STAT309

HW 1: Matlab exercises
Due: Thursday October 7 **by email** to rina@uchicago.edu
(You can email me all your '.m' function files as attachments, or just paste the code into the text of the email; if something didn't work or if you have any observations about the code or the problem, be sure to write it in your email).

1. Write a function that inputs a vector v, and outputs the value and position of the maximum element in v, without using the 'max' command (or the commands 'min', 'sort', or any command that does anything similar - you may only use basic arithmetic symbols like +,-,<,>,etc, and function commands like 'if','for',etc).

   Recall that since you need a two-part output, your function should start with a first line that looks like:
   function [value,position]=FindMaximum(v)

2. Write a recursive function (a function which calls on itself) to compute the Fibonacci sequence. Your function should input the desired length n, and output the first n Fibonacci numbers.

3. In this exercise you will write functions to produce matrices for basic column operations. For example, if the column operation is 'Multiply the first column by 5.5', your function would produce a matrix A, such that for any other matrix M (of the correct size), multiplying $M \times A$ would have the effect of multiplying the first column by 5.5.

   - Start by completing exercise 1.1 on page 9 of the book. You don't need to hand it in, but you need to be able to do it before trying the coding problems below.
   - Next, write functions to do each of the following tasks. Make sure that your functions work properly by testing them on examples (such as the example in exercise 1.1).
     (a) Write a function to produce a matrix which multiplies column j by a scalar c. Start your function file with the line:

     function A=ColumnTimesScalar(ncols,j,c)

     Here 'ncols' is the number of columns in a matrix M that you might be operating on.
     (b) Write a function to produce a matrix which deletes column j. Start your function file with the line:

     function A=DeleteColumn(ncols,j)
     (c) Write a function to add a multiple of column j to column k. (For example, you might add 5.5 times the 1st column to the 7th column; that answer would then replace the 7th column.) Start your function file with the line:

     function A=AddColumnTimesScalar(ncols,j,k,c)
     (d) Write a function that duplicates column j (the duplicate should appear right after column j). Start your function file with the line:

     function A=DuplicateColumn(ncols,j)

4. (Optional challenge problem) Repeat problem 1, but this time make your function a recursive function.