Math 451 ASSIGNMENT 9 **Due at various days and times - see below**

The purpose of this exercise is to create a function that will take a matrix and a vector as arguments (inputs) and return the product of the matrix and vector (the product itself another vector) as its output. You will be asked to do this in a series of function files, each a bit more sophisticated than the one before. They will have staggered turn-in times, and **I will send out working codes after each one is due**, that can be used for the next step if you don't manage to get one step done.

I will be in my office 8:30-9:45, 12:00-1:00 and 2:00-4:00 Tuesday, and can come to any of the Boivin computer labs during those times if you need help. If I'm not in my office, look in our usual computer lab classroom.

1. Multiply a 3×3 matrix times a vector with three components **by hand, paying careful attention to what you do along the way**. Note that there are three steps, one to create each component of the resulting vector. But then note that each of those steps consists of three multiplications. Thus you will have a loop that you go through three times to create each component, but inside that will be another loop for doing the three multiplications that are added up to get a single component. **You will not turn anything in for this.**

2. Create a function file that takes a 3×3 matrix and a vector with three components and finds their product one component at a time. Initialize the output vector (*as a matrix*) as all zeros using the `zeros` command. Then write a separate line of code to find each component of the output vector **using no matrix or vector computations**, so there will be just four lines of code within the function. Call it *yourfirstname_9a* and e-mail it to me. **Due at 4 PM Tuesday, April 29th**.

3. Condense your three lines into one line inside a for loop. Look at which value(s) in your three lines changed *from line to line*, and replace them with a counter variable. Name the new version *yourfirstname_9b* and e-mail it to me. **Due at 4 PM Tuesday, April 29th**.

4. Now look at the line of code inside your for loop. The goal now is to put another for loop around that line so that the right hand side can be condensed in the event that the matrix and vector are very large.

   (a) Put a `for` and `end` on either side of that statement, without putting a counter on the for loop yet.

   (b) Due to your initialization of the output vector, each of its components begins as a zero. The first time through loop, increase that value by the result of the first multiplication. Each time after that you will add the latest product to the sum until you have created the entire sum. This will be similar to the dot product computation discussed in lecture on Monday.

   (c) Figure out what your counter needs to go to for the inner for loop and try running your function. When it works, save it as *yourfirstname_9c*. **Due at 10 AM Thursday, April 30th - e-mail me what you have at that time, even if it doesn't work**.

5. Now alter your function to accept a matrix and vector of any sizes *but such that the multiplication is defined* and create the product. Save this as *yourfirstname_9d* and e-mail it to me. **Due at 4 PM Thursday, April 30th**.

6. Finally, make your function so that it will accept a matrix and vector of any sizes and first determine whether they can be multiplied. The output should be a message that they can't if that is the case, or the product if they *CAN* be multiplied. Save this as *yourfirstname_9e* and e-mail it to me. **Due at 11 AM Friday, May 2nd**.

Challenge: Modify your final code so that it takes two matrices as its argument and gives their product by finding it without multiplying any matrices or vectors, just numbers. If you succeed, name it *yourfirstname_mm* and e-mail it to me.