Math 451 ASSIGNMENT 8 **Due at 10 AM Sunday, April 27th**

1. Write a function file named by your last initial followed by your first initial, both lower case, then followed by AB (upper case). The exceptions are Jeremy ($rijAB$), Joe ($rojAB$), Miles ($tamAB$) and Michael ($timAB$).

   - the input to your function should be two matrices, and the output should be either
     - the product of the two matrices if they can be multiplied
     - the message *The matrices cannot be multiplied in the order given* if that is the case.
   - You are to multiply the matrices by simply using *

2. Write a function file named as above but with lower case ud instead of AB. The purpose of this function is to "turn a matrix A upside down" to get a matrix B. By this I mean the first row of B should be the last row of A, the second row of B should be the next to last row of A, and so on. **For this you will write three separate codes, building up to the final one. When one of them runs, you will copy it below itself and comment the lower one. You will then modify the copied code to get the next version.**

   (a) Your first code should be a function that takes in a 3×3 matrix A and creates B without using any loops or counters. This should be pretty simple and primitive. When it runs, copy, and comment *the lower one.* This will put each new version *above* previous versions. **Leave some space between the versions.**

   (b) Your first version should have three lines to create the three rows of B. We now want to condense those into a single line inside a for loop. To do this you will need to fill in the blank: The k row of B comes from the _____ row of A. (**your answer should be an expression containing the index** k.) When it runs, copy, and comment *the lower one.*

   (c) Now modify so that your function can read in any matrix A, of any size, and "turn it upside down." **Hint:** When you write `for k=a:b` either, or both, of a and b can be variables as long as they have been given values before the `for` loop.

   (d) **Add comments to the last version** and save.

3. Write a function file named with your initials followed by *trans* that creates the transpose of a matrix a column or row at a time. This will be similar to your last function.

4. Write a function named with your initials again, followed by *addAB* that adds two matrices *an entry at a time* when possible and returns the message *The matrices cannot be added* if that is the case. This will require what are called **nested for loops**. If you know how to do this sort of thing, carry on. Otherwise, I would suggest using the following steps, **testing each iteration of your function before moving to the next:**

   (a) Start by assuming that both matrices will be 3×3, and write a function that creates the sum of the two matrices one row at a time, with a separate line of code for each row.

   (b) Reduce your three lines of code to one that is included in a for loop to make it execute three times, once for each row.

   (c) Now, *inside that for loop*, write another for loop that gets the sum of the two rows and entry at a time.

   (d) Now assume that the two matrices will be the same size as each other, but you don't know the sizes of them.

   (e) Finally, allow the two matrices can be any sizes, and you have to determine whether they can be added, then add them if they can be and return the appropriate message if they can't.