This homework will ONLY be graded for completion (10pts). You can use the pseudocodes found in the textbook, but note you will have to translate them to MATLAB. You may also use the example we did in class.

**P1.** Write a MATLAB function with inputs

- $m$ and $n$, two positive integers,
- a matrix $A$ of size $m \times n$,
- a vector $x$ of size $n \times 1$,

and which outputs the product vector $A \times x$ (note that the size of this vector should be $m \times 1$). Your code should check that the sizes of the inputs are right and then do the multiplication using two nested “for” loops.

a) How many floating point operations (additions and multiplications) does the code use? Find a formula in terms of $m$ and $n$.

b) Run the code with $n = 100$, $m = 100$, and inputs $A = \text{rand}(m,n)$ and $x = \text{rand}(n,1)$. Modify the function by introducing a new output variable, `op_count`, which initializes to 0 outside of the nested for loops (just like we did with `product` in class) and is incremented inside the innermost of the for loops by the number of operations done to update `product`. (We will do an example of this on Wednesday.) Output `op_count` as well as `product`.

Repeat the experiment for $m = 200$, $n = 100$; $m = 100$, $n = 200$; and $m = n = 200$. Do the ratios of the new `op_count` outputs to the old one you got for $m = n = 100$ confirm the floating point operation formula you found in part a)?

**P2.** Write a MATLAB function with inputs

- $m$, $n$, and $p$, three positive integers,
- a matrix $A$ of size $m \times n$,
- a matrix $B$ of size $n \times p$,

and which outputs the product matrix $A \times B$ (note that the size of this matrix should be $m \times p$). Your code should check that the sizes are right and then do the multiplication using three nested “for” loops.

a) How many floating point operations (additions and multiplications) does the code use? Find a formula in terms of $m$, $n$, and $p$. 
b) Run the code with $m = 100$, $n = 100$, $p = 100$, and inputs $A = \text{rand}(m, n)$ and $B = \text{rand}(n, p)$. Modify the function by introducing a new output variable, $\text{op\_count}$, which initializes to 0 outside of the nested for loops (just like we did with $\text{product}$ in class) and is incremented inside the innermost of the for loops by the number of operations done to update $\text{product}$. (We will do an example of this on Wednesday.) Output $\text{op\_count}$ as well as $\text{product}$.

Repeat the experiment for $m = 200$, $n = 100$, $p = 100$; $m = 100$, $n = 200$, $p = 100$; $m = 100$, $n = 100$; $p = 200$; and $n = m = p = 200$. Do the ratios of the new $\text{op\_count}$ outputs to the old one you got for $m = n = p = 100$ confirm the floating point operation formula you found in part a)?