Name: ____________________

Date: ____________________

Collaborators: ______________________________

(Collaborators submit their individually written assignments together)

Question:	1	2	3	4	Total
Points:	15	45	20	10	90
Score:					

Instructor/grader comments:

Solving linear systems

1. Alice buys three apples, a dozen bananas, and one cantaloupe for $2.36. Bob buys a dozen apples and two cantaloupes for $5.26. Carol buys two bananas and three cantaloupes for $2.77. How much do single pieces of each fruit cost?

   (a) (5 points) In the space below write down the system of three linear equations with three unknowns that describe the problem:

   (b) (5 points) In matrix notation, $Ax = b$, what are A and b for your system of equations:

   (c) (5 points) Use matlab's backslash operator to solve the system of equations above. Store your code (generation of A and b, calling your function, printing the solution, etc.) in the file **hw02p1.m**.

   Hint: You might want to use the command `format bank` to show numbers with only two digits after the decimal point.

2. The purpose of the this exercise is to write a script that (a) measures the performance of matlab code and (b) try to deduce the scaling properties of the algorithm that built-in function `det()` uses.

   (a) (5 points) modify the function `mycramer()` that we wrote in class to use matlab function `det()` instead of our own `mydetermin()`.

   Test your function by comparing the results produced by your function with the results produced by matlab's backslash operator. Use small random matrices (size

$3 < n < 10$) and corresponding random column vectors for testing. Do not include the code for your tests into the homework submission.

(b) (15 points) Let's `nmin = 200` and `nmax = 400`. Preallocate a one dimensional array of the size `(nmax - nmin + 1)` for storing the timing. Initialize the random number generator, e.g. as following: `seed = 23; rng(seed);` The value of the `seed` variable can be an arbitrary integer.

For `n = nmin:nmax` repeat the following steps:

1. generate a random square matrix `A` and a random column vector `b` of size `n`
2. warm up by solving the system of linear equations `A*x = b` using your function `mycramer`
3. repeat the solution again measuring the time for the function call and store the time into the `(n - nmin + 1)`th element of the array you preallocated earlier

(c) (5 points) Plot the graph of time vs. matrix size. Chose the type of the axis (linear or log). Provide labels, title, grid. Use 'disconnected dots' line style.

(d) (20 points) The running time of the function call `mycramer()` is dominated by the run time of `det()`. The later time is proportional to n^α, where α is a small number. To estimate α, plot on the same figure three graphs of the functions $C_1 n$, $C_2 n^2$, and $C_3 n^3$, where C_i are irrelevant constants that you chose so that the graphs are possibly more informative. Decide which α is the best to describe your numerical experiment. Describe your reasonings and findings in the project README.md file

3. Students of linear algebra learn that the solution to the system of linear equations

$$Ax = b$$

can be written

$$x = A^{-1}b,$$

where A^{-1} is the inverse of matrix A;

$$AA^{-1} = I,$$

where I is the identity matrix.

Here is one of many methods how to compute A^{-1}: the inverse of a matrix A can be defined as the matrix X whose columns x_j solve the equations

$$Ax_j = e_j,$$

where e_j is the jth column of the identity matrix.

Note: in the vast majority of practical computational problems, it is unnecessary and inadvisable to actually compute A^{-1}.

(a) (15 points) Using the gaussian elimination code that we develop in class, write a matlab function `hw02p3inverse(A)` that uses the algorithm above to compute the inverse of A. Your function is not supposed use the matlab backslash operator or `inv` function. Your function is supposed to have a help text for use with matlab help system.

(b) (5 points) Write a matlab script (call it e.g. **hw02p3.m**) that tests your function by comparing the inverses it computes with the inverses obtained from the matlab `inv(A)` function using two random matrices of size $n = 8$. Include the command `help hw02p3inverse` in your script

Hint: one way to compare two matrices, say A and B, is to use matlab command `norm(A - B)`.

Gitlab

4. (10 points) Create a gitlab project called **hw02** (name it exactly as shown). Upload **all** matlab files that are required to run your code. Share the project with the instructor and the TA and grant them **Reporter** privileges.