Print your name: SOLUTION

- If you need more space to answer a question than we give you, you may use the additional blank sheet of paper attached to your exam. Make sure that we know where to look for your answer.

- Read each question carefully and make sure that you answer everything asked of you. Write legibly so that we can read your solutions. Please do not write anything in red.

- We suggest that for solutions that require you to write Python code, you include comments. They will help your grader understand what you intend, which can help you get partial credit.

- You have until noon to complete the exams
Question 1  
15 points

Consider the following pair of linear equations.

\[-3y + 2x = -4\]
\[-2y + 3x = -1\]

a.) Solve for \(x\) and \(y\) using any technique that you would like (show all your work).

b.) Formulate these equations as a vector scaling/summation problem.

c.) Sketch (on the next page) the two vectors, their scaled versions, and their vector sum.

\[a.) \quad -3y + 2x = -4\]
\[2x = 3y - 4\]
\[x = \frac{3}{2}y - 2\]
\[-2y + 3x = -1\]
\[-2y + 3\left(\frac{3}{2}y - 2\right) = -1\]
\[-2y + \frac{9}{2}y - 6 = -1\]
\[-\frac{4}{2}y + 9y - 12 = -2\]
\[5y = 10\]
\[y = 2\]

\[\Rightarrow \quad x = \frac{3}{2}(2) - 2\]
\[\begin{align*}
\text{if } & x = 1 \\
\text{then } & y = 2
\end{align*}\]

\[b.) \quad \begin{pmatrix} -3 \\ -2 \end{pmatrix} y + \begin{pmatrix} 2 \\ 3 \end{pmatrix} x = \begin{pmatrix} -4 \\ -1 \end{pmatrix}\]
The image shows a graph with the following points:

- Point $A$ at $(-1, -1)$
- Point $B$ at $(-3, -2)$
- Point $C$ at $(2, 3)$
Question 2  

15 points

Consider the following two linear equations. Formulate these equations in matrix/vector form and then solve for $y$ and $z$ by directly solving this matrix/vector formulation (show all you work).

Clearly denote which quantities in your solution are matrices (capital letters), vectors (lower-case letters with a line above it), and scalars (lower-case letters).

\[
\begin{align*}
4y &= 1 \\
2y &= 1 - 2z
\end{align*}
\]

\[
\begin{pmatrix}
4 \\
2
\end{pmatrix} y + \begin{pmatrix}
0 \\
2
\end{pmatrix} z = \begin{pmatrix} 1 \end{pmatrix}
\]

\[
\begin{pmatrix}
4 & 0 \\
2 & 2
\end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\]

\[
\mathbf{\hat{u}} = \mathbf{b} \\
\mathbf{\hat{u}} = \mathbf{\eta}^{-1} \mathbf{b}
\]

\[
\mathbf{\hat{u}} = \frac{1}{4 \cdot 2 - 0 \cdot 2} \begin{pmatrix} 2 & 0 \\
-2 & 4
\end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\]

\[
= \frac{1}{8} \begin{pmatrix} 2 \\ 2 \end{pmatrix}
\]

\[
= \begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \end{pmatrix}
\]
Question 3  

10 points

You are given the 2-D vectors $\vec{x} = (x_1 \ y_2)^T$ and $\vec{y} = (y_1 \ y_2)^T$ and asked to determine a vector $\vec{z} = (z_1 \ z_2)^T$ whose dot-product with $\vec{x}$ and $\vec{y}$ are each 1. Specify the fully constrained linear system (in matrix/vector form) for determining $\vec{z}$. You do not need to solve the system.

Clearly denote which quantities in your solution are matrices (capital letters), vectors (lower-case letters with a line above it), and scalars (lower-case letters).

$$
\begin{pmatrix}
  x_1 & x_2 \\
  y_1 & y_2
\end{pmatrix}
\begin{pmatrix}
  z_1 \\
  z_2
\end{pmatrix} =
\begin{pmatrix}
  1 \\
  1
\end{pmatrix}

M \vec{z} = \vec{b}

\vec{z} = M^{-1} \vec{b}$$
Question 4  
15 points

Consider the following:

\[
\begin{pmatrix}
5 \\
2 \\
1
\end{pmatrix} + b \begin{pmatrix}
15 \\
6 \\
3
\end{pmatrix} = \begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
\]

a.) Is this an under- or over-constrained system of constraints?

b.) If you believe they exist, give an example of values of \(x, y,\) and \(z\) for which you can solve this equation, otherwise explain why not.

c.) If you believe they exist, give an example of values of \(x, y,\) and \(z\) for which you cannot solve this equation, otherwise explain why not.

\(a.)\) OVER-CONSTRAINED

\(b.)\) \(a = 1, \quad b = 1 \rightarrow \begin{pmatrix}
x \\
y \\
z
\end{pmatrix} = \begin{pmatrix}
20 \\
8 \\
4
\end{pmatrix}
\)

\(c.)\) BECAUSE \(\begin{pmatrix}
15 \\
6 \\
3
\end{pmatrix} = 3 \begin{pmatrix}
5 \\
2 \\
1
\end{pmatrix}, \) ANY VECTOR THAT IS NOT A MULTIPLE OF \(\begin{pmatrix}
5 \\
2 \\
1
\end{pmatrix}\) FAILS TO SATISFY SYSTEM.

e.g., \(\begin{pmatrix}
1 \\
1
\end{pmatrix}\).
Question 5  
20 points

You are given \( n > 1 \) points \((x_i, y_i, z_i), i = 1, 2, ..., n\), and asked to estimate, using least-squares, the parameter \( a \) that best fit the model: \( z = ax^2 + y \).

a.) Specify the over-constrained system of linear equations in matrix form.

b.) Specify the quadratic-error function for this least-squares estimation.

c.) Solve for the least-squares estimator by differentiating your error function, setting it equal to zero, and then solving for \( a \).

Clearly denote which quantities in your solution are matrices (capital letters), vectors (lower-case letters with a line above it), and scalars (lower-case letters).

\[
\begin{align*}
\mathbf{x}^T \mathbf{a} &= \mathbf{b} \\
E(a) &= \| \mathbf{x} a - \mathbf{b} \|^2 \\
\frac{dE}{da} &= 2 \mathbf{x}^T (\mathbf{x} a - \mathbf{b}) = 0 \\
\mathbf{x}^T \mathbf{x} a &= \mathbf{x}^T \mathbf{b} \\
\mathbf{a} &= \frac{\mathbf{x}^T \mathbf{b}}{\mathbf{x}^T \mathbf{x}}
\end{align*}
\]
Question 6 15 points

You previously wrote a recursive version of binary search. Here you will write an iterative (non-recursive) version of binary search. Your Python function will take two parameters, a list of integers A and an integer x. If x is found in the list, then your function should return the index of this element (e.g., if the list is [1, 3, 5, 7, 9] and x is 3, then your function will return 1). If x is not found in the list, then your function should return -1. Your function should only use a single while loop and a single if-elif-else ladder.

HINT: the iterative version of binary search works very similar to the recursive version in which you will keep track of a left/right index value, compute the mid-point, determine which half of the list the element (if present) is in, update the values of the left/right indices accordingly, and repeat.

```python
def binary_search(A, x):
    left = 0
    right = len(A) - 1
    while (left <= right):
        mid = (left + right) / 2
        if (A[mid] > x):
            right = mid - 1
        elif (A[mid] < x):
            left = mid + 1
        else:
            return (mid)  # FOUND IT
    return (-1)  # DIDN'T FIND IT.
```
Question 7  10 points

Below is the code for insertion sort. Using big-O notation, specify the run-time complexity of insertion sort when the list \( L \) is sorted, briefly explain your answer?

```python
def insertion_sort(L):
    for i in range(1, len(L)):
        key = L[i]
        j = i - 1
        while j >= 0 and L[j] > key:
            L[j+1] = L[j]
            j -= 1
        L[j+1] = key
```

**Because** \( L[i-1] > L[i] \) **is always false for a sorted list, the inner loop never evaluates, so run-time is dictated by outer for-loop \( \Rightarrow O(n) \).**