Name : ____________________ **Class Number:** ______

ENGR 11: Lesson 2 Suggested Problems

Theoretic Problems: Discussed in notes

---

1. **MATLAB: MATrix Laboratory**

   A. When we say "array programming language," what do we mean?

   B. What are the two types of arrays we focus on in this lesson?

   C. What is the difference between one- and two-dimensional arrays?

   D. What recommendations does Jeff make for naming vectors and arrays in MATLAB?

---

2. **Arrays: one versus two dimensions**

   A. What is a *one-dimensional array*?

      i. What is a column vector?

      ii. What is a row vector?

      iii. What are the similarities and differences between column and row vectors?

      iv. What does the transpose operator do to vectors? What happens when we take the transpose of a column vector? What happens when we take the transpose of a row vector?

   B. What is a *two-dimensional array*?

      i. What is the entry-by-entry definition of a matrix?

      ii. What are the dimensions of a matrix?

      iii. What is a diagonal matrix? What is the main diagonal of a matrix?

      iv. What does I_n stand for? Write the entry-by-entry definition of this matrix for $n = 2, 3, 4, 5$.

      v. What is the column partition of a matrix? How is this related to colon notation?

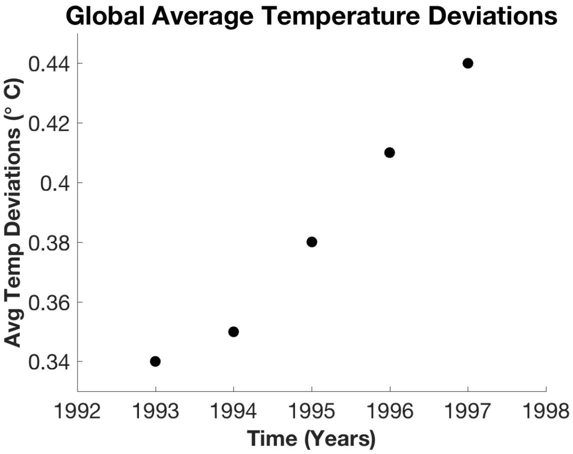
      vi. What is the row partition of a matrix? How is this related to colon notation?

Problems Solved in Jeff's Notes

3. What is the general syntax we use to create column vectors in MATLAB? What are the two (equivalent) methods we can use to delimit the individual row entries of a column vector?

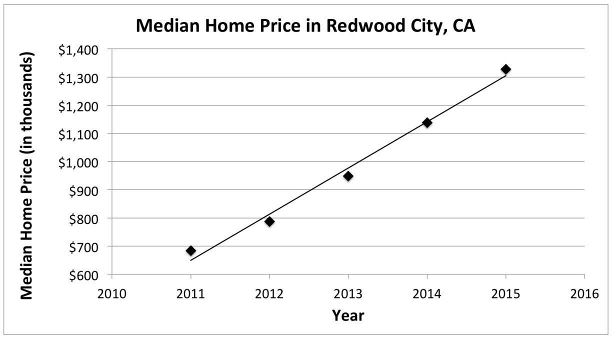
4. Eg. 2.2: Use the following data set of global mean temperature deviations during the 1990's to create two column vectors, named `year` and `Avg_temp_dev`, in MATLAB.

Year	Global Average Temperature Deviations (in °C)
1993	0.3400
1994	0.3500
1995	0.3800
1996	0.4100
1997	0.4400



5. Eg. 2.3: Use the `clear` command to delete the variable `year`. Then, use the following data set of median sales prices for single family homes in Redwood City, CA during the 2010's to create two column vectors, named `year` and `RWC_home_price`.

Year	Median Home Price (in thousands of $)
2011	$684.0
2012	$787.5
2013	$949.0
2014	$1138.0
2015	$1328.0



6. What is the general syntax we use to create row vectors in MATLAB? What are the two (equivalent) methods we can use to delimit the individual column entries of a row vector?

7. Eg. 2.4: Delete the variable `year` from the Workspace. Then, use the following data set of population growth in the USA from 1930 - 2000 to create two row vectors, named `year` and `USA_Population`.

Year	1930	1940	1950	1960	1970	1980	1990	2000
Population	123.2	132.2	151.3	179.3	203.3	226.5	248.7	271.4

8. What is MATLAB's colon syntax, with a general format given as

```
variable_name = [ m : q : n]
```

used to do? Explain every symbol in this syntax. Be sure to interpret the variables `m`, `q`, and `n`. After you've explained the general purpose and format of this syntax, use this to do the following:

Eg. 2.5: Create the row-vector-valued variable `t` given by $t = \begin{bmatrix} 2 & 4 & 6 & 8 & 10 \end{bmatrix}$.

Eg. 2.6: Create the row-vector-valued variable `x` given by $x = \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \end{bmatrix}$.

Eg. 2.7: Create the row-vector-valued variable `y` given by $y = \begin{bmatrix} 100 & 90 & 80 & 70 & 60 & 50 & 40 \end{bmatrix}$.

Eg. 2.8: Predict what output results from the command `z = [ 0.0 : 0.3 : 1.0]`

---

9. What is MATLAB's `linspace` function, with general syntax is given as

```
variable_name = linspace(x1, x2, n)
```

used to do? Explain every symbol in this syntax. Be sure to interpret the variables `x1`, `x2`, and `n`. After you've explained the general purpose and format of this syntax, use `linspace` to do the following:

Eg. 2.9: Create the row-vector-valued variable `x` given by $x = \begin{bmatrix} 0 & 2 & 4 & 6 & 8 & 10 \end{bmatrix}$.

Eg. 2.10: Suppose we want to use `linspace` to produce the row vector $\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}$. With this goal in mind, what is the difference in the output produced by the command `linspace(0, 5, 5)` and output that results from the command `linspace(0, 5, 6)`? Which command produces the desired result? Why?

Eg. 2.11: Create the row-vector-valued variable `y` given by $y = \begin{bmatrix} 60 & 54 & 48 & \cdots & 12 & 6 & 0 \end{bmatrix}$.

Eg. 2.12: Predict what output results from the command `linspace(0, 10)`. What is the default dimension of a row vector output by `linspace`.

---

10. Eg. 2.13: Create a matrix-valued variable $A \in \mathbb{R}^{3\times 3}$, whose individual entries are given by

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

---

11. Eg. 2.14: What occurs as a result of executing the following command?

```
A = [
1:2:11;
0:5:25;
linspace(10,60,6);
67, 2, 43, 68, 4, 13]
```

What value is stored for variable `A` in the Workspace?

12. Eg. 2.15: What occurs as a result of executing the following command? Analyze this code line-by-line and try to predict the effect of each line of code before executing this in the Command Window. Then, execute this command and compare the results with your predictions.

```
a = 12; b = -4; c = 15; d = 1;
Mat = [
a, b*d, sin(pi/2), c^2;
d, sqrt(c^2/5) 14^2/7, cos(pi/4)]
```

13. Eg. 2.16: Consider the code below:

```
C = [
1, 0, 0, -1;
-1, 1, 0, 0;
0, 1, -1, 0;
0, 0, -1, 1;
0, -1, 0, 1]
```

Write the expected output by hand before running this command in MATLAB. Then, compare your written output with the actual output produced when you execute this command within the command window.

14. Eg. 2.17: What happens when you execute the command below?

```
D = [
1, 2, 3;
1, 2];
```

Why does MATLAB produce the results you see? What can you do to modify this command to conform to MATLAB's rules for matrices?

15. Properly use the `zeros`, `ones`, and `eye` commands:

Eg. 2.18: Create a matrix $Z \in \mathbb{R}^{4\times 5}$ that has all zero entries using the `zeros` function.

Eg. 2.19: Use the `ones` function to create a matrix $N \in \mathbb{R}^{6\times 3}$ in which all entries equal 1.

Eg. 2.20: Create the square identify matrix $I_4 \in \mathbb{R}^{4\times 4}$ using the `eye` function.

16. Properly use the transpose operator ' to transform to take the transpose of vectors and matrices:

Eg. 2.21: Consider the following code:

```
x = [0 : 0.5 : 2 ]
y = x'
```

Predict the output when we execute these commands in the Command Window. Then, test your hypothesis by actually running these commands.

Eg. 2.21: What output does the command `y = linspace(0, 2, 5)'` produce?

Eg. 2.22: Consider the following code:

```
A = [   1,  0,  -3;
        0,  1,   2;
        0,  0,   1];
B = A';
```

Write out the entry-by-entry definition of B. What is $B(3,2)$? How about $B(1,3)$?

17. Refer to a contiguous sets of elements in array by using the proper addressing techniques:

Eg. 2.23: Create the 1×10 vector $\mathbf{x}$ given by

$$\mathbf{x} = \begin{bmatrix} 0 & 1 & 4 & 9 & 16 & 25 & 36 & 49 & 64 & 81 \end{bmatrix}.$$

Then use one-dimensional array addressing to execute each of the following operations:

$$x_{17} + x_{19}, \qquad\qquad (\ x_{13}\ )^{x_{11}} + \sqrt[2]{x_{18}}$$

Eg. 2.24: Create the 3×4 matrix

$$A = \begin{bmatrix} 5 & 1 & 8 & 0 \\ 10 & -1 & 9 & 14 \\ 2 & 3 & -3 & 7 \end{bmatrix}.$$

Then, use two-dimensional array addressing to output the following entries in the Command Window:

$$a_{31} = 2, \qquad a_{13} = 8, \qquad a_{24} = 14.$$

Eg. 2.25: Create the 7×1 column vector

$$\mathbf{x} = \begin{bmatrix} 30 & 25 & 20 & 15 & 10 & 5 & 0 \end{bmatrix}^T$$

Then use colon notation to output the rows $1 - 7$ of this column vector. Now use colon notation to output only rows $3 - 6$ of this vector.

Eg. 2.26: Create the 1×8 row vector

$$\mathbf{y} = \begin{bmatrix} -800 & -600 & -400 & -200 & 0 & 200 & 400 & 600 \end{bmatrix}$$

Then use colon notation to output columns $1-8$ of this row vector. Now use colon notation to output only columns $4 - 7$ of this vector.

Eg. 2.27: Consider the of the following code:

```
A = [   1:2:11;
        linspace(2,12,6);
        3:3:18;
        linspace(4,24,6);
        linspace(5,25,6)]
```

Before executing this command, try to predict the output (actually write it down on paper). Then execute this command and compare the actual output with your prediction. Now use colon notation to do each of the following:

A. Output all rows in column 4
B. Output rows $2 - 3$ of column 4
C. Output all columns in rows $2 - 3$
D. Output all columns in row 4
E. Output columns $2 - 5$ in row 4
F. Output all rows in columns $2 - 5$

18. Refer to a general sets of elements in array by using the proper addressing techniques:

Eg. 2.28: Predict the output that results from the commands below. Actually write down your results on paper before executing these commands. Then, run this code and compare the results you achieve to your initial guesses.

```
x = [66:-6:11]
v = x(1, [2, 5, 8:10])
A = [   10:-1:4;
        ones(1,7);
        2:2:14;
        zeros(1,7)]
B = A([1, 3:4], [1,3,5:7])
```

19. Add, change, or delete elements in existing vector-valued variables using the proper addressing techniques

Eg. 2.29: Predict the output that results from the commands below. Actually write down your results on paper before executing these commands. Then, run this code and compare the results you achieve to your initial guesses.

```
x = [0, 1, 8, 27, 64]
x(1, 6:10) = [125, 216, 343, 512, 729]
w = x(1, 1:5)
y = x(1, 6:10)
z = [w, y]
```

Eg. 2.30: Predict the output that results from the commands below. Actually write down your results on paper before executing these commands. Then, run this code and compare the results you achieve to your initial guesses.

```
twos = linspace(2,2,5)
twos(1,9) = 2
threes(8,1) = 3
```

Eg. 2.31: Create block-partitioned vectors via concatenation using the following code

```
v1 = [10, -10, 8, -8]; v2 = [4:-2:-5];
v3 = [v1, v2]
v4 = [v2'; v1']
```

Eg. 2.32: Add rows and columns to an existing matrix using the code given below. Discuss the impact of each line of code.

```
B = [   0,  0,  1;
        0,  1,  0;
        0,  0,  1];
A = [eye(3), ones(3,3), B]
C = [eye(3), B]
D = [3:3:9; 12:-4:4]
D(3:4,:) = [zeros(1,3); ones(1,3)]
```

Suggested Problems

20. Use either the `linspace` command or colon notation in the form `[m : q : n]` to accomplish each of the following tasks:

    A. Create a 7×1 column vector `n` that has seven equally spaced elements in which the first element is 24 and the last element is 3. Remember that a column vector can be created by taking the transpose of a row vector.

    B. Using the `linspace` command, create a row vector with variable name `eights` that has 11 elements that are all the number 8.

    C. Use a single command to create a 1×9 row vector with variable name `b` with the following elements

    $$\begin{bmatrix} -10 & -4 & 2 & 8 & 17 & 23 & 28 & 33 & 38 \end{bmatrix}$$

    D. Create a variable `h` that is a column vector in which the first element is 8 and the last element is 80 and the increment between elements is 4.

21. Create a 1×16 row vector `v1 = -5:3:40`. Then, using the elements from `v1`, create a new 10×1 column vector `v2` such that the first five rows of `v2` are identical to the first five columns of `v1`, and the last five rows of `v2` are equal to columns $11 - 15$ of vector `v1`. When creating the vector `v2`, do not type the elements of the vector `v1` explicitly. Instead, use the colon symbol to address the desired range of elements.

22. Create each of the following matrices by typing one command. Do not type individual elements.

$$\text{A.}\ \begin{bmatrix} 3 & 7 \\ 3 & 7 \\ 3 & 7 \\ 3 & 7 \\ 3 & 7 \\ 3 & 7 \end{bmatrix} \qquad \text{B.}\ \begin{bmatrix} 1 & 0.5 & 0 & -0.5 & -1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 16 & 12 & 8 & 4 & 0 \end{bmatrix} \qquad \text{C.}\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad \text{D.}\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & -2 & -4 \end{bmatrix}$$

23. Create each of the following matrices by typing one command. Do not type individual elements.

$$M = \begin{bmatrix} -0.2 & 0.0 & 0.2 & 0.4 & 0.6 & 0.8 & 1.0 & 1.2 & 1.4 \\ 20 & 15 & 10 & 5 & 0 & -5 & -10 & -15 & -20 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ -0.04 & -0.03 & -0.02 & -0.01 & 0.00 & 0.01 & 0.02 & 0.03 & 0.04 \\ -1000 & -600 & -200 & 200 & 600 & 1000 & 1400 & 1800 & 2200 \end{bmatrix}$$

    A. Create a 3×6 matrix M_1 from the first, third, and fifth rows and the first three and last three columns of the matrix M.

    B. Create a 3×3 matrix M_2 from rows $2 - 4$ and the second, fifth and eighth columns of M

    C. Change all elements in the the third row of M to have value -3.

    D. Delete the fourth row of M.