Assignment 02 (Due: Monday, February 1, 2016)

CSCE 155*N*

1 Lab Objectives

- Improve your understanding of arrays and array operations
- Differentiate array operators and matrix operators
- Create, access, modify, and resize arrays
- Use arrays and logical arrays as indexes into another array
- Solve computational problems that involve array manipulation

2 Prior to Laboratory

- **Read** the laboratory handout and **verify** (on a computer) the examples in the **Practice** sections
- Read chapters 1 & 2

3 Topics Covered in Lab

- Multidimensional Arrays and its internal storage mechanism
- Array operators and matrix operators
- Array modifications
- Array slicing and subarrays
- Simple problem solving

4 Activities/Exercises

- Array operators and matrix operators
- Array modification drills
- Simple problem solving

4.1 Practice: Array Creation

Type each of the following assignment statements in the command window

4.1.1 Arrays

- `array = [ 1 2 3 4 ]`
- `a = [ 0 1 + 7 ]`
- `b = a( 2 )`
- `a( 2 ) = 12; disp( a );`
- `b = [ a( 2 ) 7 a ]`
- `x = [ 1 : 2 : 10 ]`
- `y = 1 : 2 : 10`
- `ylen = length( y ); disp( ylen );`
- `g = 1 : 4; k = g'; disp( k );`

4.1.2 Matrices

- `x = [ 1 2 3; 4 5 6 ]`
- `y = [ 1 2 3; 4 5 ]`
- `x( 2 )`
- `x( 2 , 1 )`
- `x( 2 , 3 ) = 20;`
- `a = 2 + x`

  `a = 2 - x`

  `a = 2 * x`

  `a = x / 2`
- `b = zeros( 2 )`
- `c = size( b )`
- `d = zeros( 1 , 2 )`
- `e = ones( 3 )`
- `f = eye( 4 )`

4.2 Before You Begin

- **Download files from** http://cse.unl.edu/~cse155n/labs/02 **to your** `Z:\csce155n` **directory**

4.3 Array Operations

- Modify `AtimesB.m`, so that the result of $A \times B$ is correctly computed and returned
- Modify `CtimesD.m`, so that the result of $C \times D$ is correctly computed and returned
- Modify `CplusD.m`, so that the result of $C + D$ is correctly computed and returned
- **Not all of these operations are straight forward. In the comments, explain what steps you used to accomplish the required task**
- An example of the values A, B, C, and D is provided below
- **You can test each function with AtimesBGUI, CtimesDGUI, and CplusDGUI**

$$A = \begin{bmatrix} 10 & 2 & 4 \\ 5 & 6 & 2 \\ 3 & -3 & 1 \end{bmatrix} \; B = \begin{bmatrix} 5 & 9 & -12 \end{bmatrix} \; C = \begin{bmatrix} 5 & 10 & 4 \\ -3 & 5 & -3 \end{bmatrix} \; D = \begin{bmatrix} 1 & 2 \\ 5 & 10 \\ 8 & 6 \end{bmatrix}$$

4.4 3D Matrices

- Modify `makeMatrix.m`, so that the function returns a matrix with three (3) dimensions
- **makeMatrixGUI can be used to test your code**

4.5 Vector Manipulation

- Modify `timeToClock.m`, so that the function returns the correct *whole* number of hours, minutes, and seconds
- **The built-in MATLAB function `fix` can be used to get the integer part of a fractional number**
- **timeToClockGUI can be used to test your code**

4.6 Mathematical Operations over Vectors and Basic Plotting

- Modify `plotCosine.m`, to create a vector of 50 evenly spaced points between $-\pi$ and π, calculate the cos of those points, then plot those points, versus their cos
- Label the plot's x axis and y axis
- Include a legend for the curve
- Save your plot, as an EPS file, with the filename `plotCosine.eps`
- **Hint: MATLAB has built-in functions** `linspace`, `cos`, `plot`, `xlabel`, `ylabel`, and `legend`

4.7 Plotting Several Curves

The sales (in billions of dollars) for two separate divisions of XYZ Corporation for each of the four quarters of 2014 are stored in a file called `salesFigures.mat`.

- Modify `plotSales.m`, to load the file `salesFigures.mat`
- Separate this 2×4 matrix into vectors for the two divisions
- Create a single plot that contains two curves (one for each division)
- **Each division's curve should be a different color**
- Label the plot's x axis and y axis
- Include a legend for the curve
- Save your plot, as an EPS file, with the filename `plotSales.eps`
- **Hint: MATLAB has built-in functions** `load` and `hold`

4.8 `members02lab.txt`

- Open file `members02lab.txt`
- Replace `bwayne` and `dgrayso` with the `cse.unl.edu` usernames of you and your partner
- Save the File

5 webgrader and `diffs`

Because the webgrader will test your programs and supply the input (and handle the output), the `diff` program is being used to check for the correctness of your programs. If nothing appears in the `diff` section, that means that your program produced the correct output for the given input.

You must run the webgrader at least once before $11:59:00$**pm on Tuesday night.**

5.1 `contributions02lab.txt`

- Open file `contributions02lab.txt`
- Write **your** explanation of what you and your partner each contributed to completing the lab
- Save the File

6 Code Documentation

Remember to document your files in the way that we did for the previous labs. It will come in handy when you look back at code after a long time, or when someone else is trying to understand what your code does.

7 What to Submit

You will be submitting eleven (11) files (`AtimesB.m`, `CtimesD.m`, `CplusD.m`, `makeMatrix.m`, `timeToClock.m`, `plotCosine.m`, `plotCosine.eps`, `plotSales.m`, `plotSales.eps`, `members02lab.txt`, and `contributions02lab.t`
The webgrader will also tell you what was received.

8 Additional Resources

Online MATLAB Documentation
CSE Webhandin
CSE webgrader

9 Think About...

- There are many useful functions already provided in MATLAB. Research ways that `min` and `prod` could be used
- MATLAB supports treating arrays as a single entity (and hiding how individual elements are used). What are the advantages and disadvantages of hiding these details from the user?

10 Point Allocation

Component	Points
`AtimesB.m`	8
`CtimesD.m`	8
`CplusD.m`	8
`makeMatrix.m`	8
`timeToClock.m`	8
`plotCosine.m`	8
`plotCosine.eps`	12
`plotSales.m`	8
`plotSales.eps`	12
`members02lab.txt`	5
`contributions02lab.txt`	5
webgrader PDF	10
Total	100