Introduction to MATLAB

Introductory Problems

1. Create a vector that contains the numbers 1, 2, 3, 4, 5:
   vector = [ 1 2 3 4 5 ];
   vector = [ 1:5 ];

2. Clear the Workspace and Command windows:
   clear
   clc

3. Create a matrix with the same contents as the following table:

   \[
   \begin{array}{ccc}
   4 & 5 & 6 \\
   10 & 50 & 3 \\
   15 & 56 & 20 \\
   \end{array}
   \]
   Matrix = [ 4 5 6; 10 50 3; 15 56 20 ];

   Alternatively:
   
   V1 = [4 5 6];
   V2 = [10 50 3];
   V3 = [15 56 20];
   Matrix1 = [V1; V2; V3];

   **Note:** The semicolon between V1, V2 and V3 specify that we want to insert V2 and V3 into new rows of the matrix. Moreover, Matrix = [V1, V2, V3] would insert all variables into the same row.

4. Transpose the matrix from (3) using an inverted comma ('):

   \[
   \begin{array}{ccc}
   4 & 10 & 15 \\
   5 & 50 & 56 \\
   6 & 3 & 20 \\
   \end{array}
   \]
   Matrix2 = [ 4 5 6; 10 50 3; 15 56 20 ]' ;
   Or
   TransposedMatrix = Matrix';

   **Note:** You cannot use spaces in variable names, instead remove the spaces or use an underscore.
6. Vertically concatenate matrices 1 and 2: 
\[
\text{Matrix3} = \text{cat}(1, \text{a}, \text{b});
\]

7. Horizontally concatenate matrices 1 and 2: 
\[
\text{Matrix4} = \text{cat}(2, \text{a}, \text{b});
\]

8. Create a cell array that contains matrix 1 in slice 1 and matrix 2 in slice 2: 
\[
\text{CellArray} = \text{zeros}(3, 3, 2);
\]
\[
\text{CellArray}(\cdot,\cdot,1) = \text{Matrix1};
\]
\[
\text{CellArray}(\cdot,\cdot,2) = \text{Matrix2};
\]

Notes: zeros creates an empty matrix, as given: 
zeros(Number Columns, Number Rows, Number Slices)  
We use a colon in the second and third lines to select all elements of each column and row.

8. Determine the third power of each element within vector (a): 
\[
\text{a} = [4 \ 6 \ 7] .^3
\]

9. Multiply each element of vector (a) by the respective element in vector (b): 
\[
a = [4 \ 6 \ 7];
\]
\[
b = [3 \ 4 \ 5];
\]
\[
c = a.*b;
\]

10. Multiply each element of matrix 1 by the respective element in matrix 2 
\[
\text{Matrix5} = \text{Matrix1} * \text{Matrix2};
\]

Note: Removing the dot (.) will perform matrix multiplication, which calculates a very different result. Check for yourself. In these problems, we essentially use matrices as arrays to store data.

11. Create an empty 3 x 3 x 2 array and fill the centre of slices 1 and 2 with 6 and 7 respectively.
\[
\text{Array} = \text{zeros}(3,3,2);
\]
\[
\text{Array}(2,2,1) = 6;
\]
\[
\text{Array}(2,2,2) = 7;
\]

12. Insert numbers 5:13 into a 3 by 3 matrix using a for loop: 
\[
\text{TempMatrix} = \text{zeros}(3,3);
\]
\[
\text{for} \ i = 1:9;
\]
\[
a = 4 + i;
\]
\[
\text{TempMatrix}(i) = a;
\]
end
13. Insert numbers 5:13 into a 3 by 3 matrix using a while loop:

```matlab
TempMatrix = zeros(3,3);
i = 0;
a = 5;

while i < 10
    TempMatrix = a + i;
i = i + 1;
end
```

14. Multiply each row of matrix 1 by the following factors using a nested loop.

```
| 4  5  6 |
| 10 50 3 |
| 15 56 20 |
```

Row 1 Multiplication Factor = 7
Row 2 Multiplication Factor = 3
Row 3 Multiplication Factor = 4

```matlab
Matrix1 = [ 4 5 6; 10 50 3; 15 56 20];
NewMatrix = zeros(3,3);
RowNumber = 1;
ColumnNumber = 1;
RowFactor = 7;

for i = 1: 3
    for j = 1:3
        NewMatrix(RowNumber, ColumnNumber) = ...
        Matrix1(RowNumber, ColumnNumber) * RowFactor;
    end
    if RowNumber == 3
        RowFactor = 3;
    elseif RowNumber == 3 && ColumnNumber == 2;
        RowFactor = 4;
    end
end
```

**Alternative Method:**

```matlab
Matrix1 = [ 4 5 6; 10 50 3; 15 56 20];
NewMatrix = zeros(3,3);
RowNumber = 1;
ColumnNumber = 1;
```
while Column_Number < 4 && Row_Number < 4
    NewMatrix(Row_Number, Column_Number) = ...
    Matrix1(Row_Number, Column_Number) * Row_Factor;
    Row_Number = Row_Number + 1;

    if Row_Number == 4;
        Column_Number = Column_Number + 1;
        Row_Number = 1;
        Row_Factor = 3;
    elseif Row_Number == 1 && Column_Number == 3;
        Row_Factor = 4;
    end
end

**Problems Without Answers**

**Problem 1**
Determine the first 100 numbers in the fibonacci sequence and plot. The first two numbers in the sequence are:

\[ F_1 = 0 \]
\[ F_2 = 1 \]

Where \( F_n = F_{n-1} + F_{n-2} \)

**Problem 2**
Create a matrix, which contains the following in each column using for/while loop (where \( x = 1:100 \)). Plot the functions on the same graph and try the subplot function.

*Note:* You must use the command: hold on to plot several functions on the same graph. Use help subplot in the command window

**Column 1:** \( y = x \)
**Column 2:** \( y = x^2 \)
**Column 3:** \( y = x^3 \)
**Column 4:** \( y = x^4 \)

**Problem 3**
Create a matrix, which contains the following in each column using a nested loop (where \( x = 1:100 \))

**Column 1:** \( y = x \)
**Column 2:** \( y = x^2 \)
**Column 3:** \( y = x^3 \)
**Column 4:** \( y = x^4 \)
Problem 4
We have collected absorbance data for Cytochrome C. Import sample and control absorbance files into MATLAB. Write a function to perform a baseline correction, plot the baseline corrected absorbance, and calculate the concentration using the Beer Lambert Law

\[ A = e \times l \times c \]

Pathlength = 1cm
Molar extinction Coefficient 280nm = 11585 M\(^{-1}\) cm\(^{-1}\)
Wavelength Range: 400 to 250 nm, 0.2 nm steps.

Note: Baseline Absorbance data has not been stored the same as Cytochrome C Absorbance data, you must either rearrange both datasets into columns or rows.