1. (5 points) Write the MATLAB commands to generate a time vector T that has a start time of 1.0, a sample period of 0.02, and an end time of 2.0.

\[
\begin{array}{c}
\text{T} = \left[ 1.0 : 0.02 : 2.0 \right]
\end{array}
\]

2. (9 points) For the time vector T defined above:
   a) What is T(4)?
      \[ T(4) = 1.06 \]
   b) What is T(end-4)?
      \[ T(\text{end}-4) = 1.92 \]
   c) What is the length of the vector?
      \[ T(5-1) = T(4) \]
   d) (Bonus - 2 points) What is [T(4) + T(end-4) + T(end)-T(1)]?
      \[ 2.08 + 1 = 3.08 \]

3. (5 points) Write the MATLAB commands to create a vector w that has ten ordered elements equal to the cube of the first ten integers (1 through 10). Note: there is more than one way to do this.

\[
\text{w} = \left[ 1, 8, 27, 64, 125, 216, \ldots \right]
\]

4. For the vector w created above:
   a) (3 points) What is the dimension of w?
      \[ 10 \]
   b) (Bonus - 3 points) What is w(4)-w(3)?
      \[ 64 - 27 = 37 \]
5. (8 points) For each of the MATLAB commands below, write what the answer (ans = ) output would be if the commands were executed. If you think the commands would produce an error, write "syntax error."

a) \[ \begin{bmatrix} 1 & 2 \\ 3 \\ 4 \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \]
\[
\begin{bmatrix}
0 & 0 & 8 & 8
\end{bmatrix}
\]

b) \[ \begin{bmatrix} 0 & 1 & 2 & 4 \\ 1 & 0 & 4 & 2 \end{bmatrix} \]
\[ \begin{bmatrix} 0 & 1 & 2 & 4 \\ 0 & 2 & 4 & 0 \end{bmatrix} \]
\[
\begin{bmatrix}
0.5 & 1 & 4 & 5
\end{bmatrix}
\]

C) \[ \begin{bmatrix} 0 & 1 & 2 & 4 \\ 1 & 0 & 4 & 2 \end{bmatrix} \]
\[ \begin{bmatrix} 0 & 1 & 2 & 4 \\ 0.5 & 0 & 2 & 0 \end{bmatrix} \]

6. Bonus: (10 points) Write the commands to create a MATLAB function called "my_bonus.m". The function should have a time vector input (call it time), and generate a one sample-period shifted time vector output (call it delay_time). This means that delay_time(2) = time(1), delay_time(3) = time(2), and so on. Both delay_time output and time input vectors should have the same length; thus, part of the problem is to figure out what delay_time(1) should be. Hints: If you apply the function to the vector [1 2 3 4], the output should be [0 1 2 3]. Also, I recommend using a for-loop, and using the MATLAB command length(time).

```matlab
function delay_time = my_bonus(time)
    delay_time(1) = time(1) - [time(2) - time(1)]
    for k = 2 : length(time)
        delay_time(k) = time(k-1);
    end

Alternate: delay_time(2 : length(time)) =
2. time(1 : length(time) - 1);
```