CS30 Spring 2015
Lab 3

Use the command `diary` to record your answers and submit them. Also submit code for the functions you write.

1. (25 points) Array construction and indexing. Let

\[
\text{array1} = [10, 20, 30, 40, 50];
\]

\[
\text{array2} = [10, -20, 30, -40, 50];
\]

\[
\text{array3} = [50, 40, 30, 20, 10];
\]

(a) Use the double color operator to construct `array1`.

(b) Use the double colon operator to index all the odd numbered elements of `array1`.

(c) Use the double colon operator to index all the even numbered elements of `array1`.

(d) Return the last element of `array1`. Do this in such a way that your code would work regardless of the length of `array1`.

(e) Return elements 1, 2, and 4 of `array1`.

(f) Construct `array2`. Use one statement to copy `array1` into the variable `array2` and another statement to negate the even elements of `array2`. Do this in such a way that the same code would work for arbitrary arrays `array1` and `array2`.

(g) Construct `array3` from `array1` using the double colon operator.

2. (30 points) Array operations. Use the definitions of `array1`, `array2`, and `array3` from Problem 1.

(a) Compute the elementwise sum of `array1` and `array2`. Compute the elementwise product of `array2` and `array3`.

(b) Write a function `ElementwiseMax` that takes two arrays and returns a single array where each element in the resulting array is the maximum of the two elements in the corresponding positions of the input arrays. E.g., `ElementwiseMax([1, 2, 3],[1, -1, 5])` would return `[1, 2, 5]`. Do the following test cases

\[
\text{>> ElementwiseMax(array1, array2);
} \]

\[
\text{>> ElementwiseMax(array1, array3);
} \]

\[
\text{>> ElementwiseMax(array2, array3);
} \]

Note that this gives the same behavior as using the builtin Matlab function `max` (but you are not allowed to use `max` in your solution).
An array can be said to be monotonically increasing if its elements are of non-decreasing value (i.e., array(1) ≤ array(2) ≤ ... ≤ array(n)) and monotonically decreasing if its elements are of non-increasing value (i.e., array(1) ≥ array(2) ≥ ... ≥ array(n)). The array is monotonic if it is either monotonically increasing or monotonically decreasing. Write a function IsMonotonic whose input is an array and whose output is true if the array is monotonic and false otherwise. You may find it useful to use the Matlab function all which checks if all the elements in a logical array are true. Run your function on array1, array2, and array3.

3. (25 points) Formatted output. Given the arrays

```matlab
data = 
  ids = [ 10, 20, 30, 40, 50 ];
initial = [ 'T', 'S', 'R', 'B', 'P' ];
ages = [ 15, 24, 19, 18, 30 ];
weights = [ 130.24, 145.2341, 190.123, 126.1, 215.12 ];
```

Write a script called `FormatTable.m` that uses `fprintf` to create the following table:

```
ID  Initial  Age    Weight
----- ------- ------- ------
  10    T       15   130.24
  20    S       24   145.23
  30    R       19   90.12
  40    B       18  126.10
  50    P       30  215.12
```

Make your result match the table above exactly, including field width, alignment, and precision.

4. (20 points) Functions and simple plotting.

(a) Define an array `x` that samples the interval [0, 2π] with 10 evenly spaced points.
(b) Define an array `y` where each element is the sin of the corresponding element of `x`.
(c) Draw a plot of `x` vs. `y`.
(d) Repeat steps (a) - (c), but now sample the interval [0, 2π] using 100 evenly spaced points. How are your results different?