Homework M5:

Matlab Primer [1] page numbers:
Conditional Control, pp. 5-2 to 5-5
Loop Control, pp. 5-5 to 5-8
Scripts and Functions, pp. 5-10 to 5-13

1. Write Matlab® code to test whether a scalar value, `x`, is larger than 1, in which case the value of `x` is set to 1, or whether `x` is less than –1, in which case the value of `x` is set to –1. (Otherwise, `x` is unchanged.)

2. Write a Matlab® function called `ll.m` (two letter el's meant to represent the parallel operator for resistors) with one argument, `Rarray`, (which is an array of resistor values). `ll.m` has the following specifications:
   i) Its return value is called `Rpar` (meaning "parallel resistance").
   ii) It has comments at the beginning that describe its use and purpose. These comments print out when `help ll` is entered at the Matlab® prompt.
   iii) It tests the values in `Rarray` to see if any of them have negative real parts. If so, it prints a warning that one or more input values has negative resistance.
   iv) It computes the parallel resistance value for all the values in `Rarray`. `Rarray` is assumed to be a horizontal vector array.

3. Use additon and the function written in problem 2 in a one-line command to find the resistance of the following network of resistors:
   10 ohms in parallel with a branch that consists of two 20 ohm resistors in parallel that are in series with a 5 ohm resistor
   Hint: the command to find the resistance of a 72 ohm resistor in parallel with a 12 ohm resistor and 24 ohm resistor in series is as follows:
   ```
   >> Req = ll([72, 12 + 24])
   ```

4. Write a Matlab® function called `spec_plot.m` that computes and plots the magnitude of the Fast Fourier Transform of its argument.

5. Write a Matlab function called `label_plot.m` that adds labels to a plot. Examples of the use of this function are as follows:
   ```
   >> label_plot('yaxis','Volts')
   >> label_plot('xaxis','time')
   >> label_plot('title','Spectrum')
   ```
   The first argument of the function is one of three terms, `'xaxis'`, `'yaxis'`, or `'title'`, and the second argument is a string containing the desired label. Use a switch statement to decide what the first argument is, and print out an error message if it doesn't match any of the three allowed choices.

6. Write a Matlab® function called `z.m` to calculate an impedance value. Examples of the use of this function are as follows:

```
>> z('R',24,0)       % Returns a value of 24 (ohms).  Third argument is frequency.
>> z('L',1e-3,100)   % Returns a value of j*0.001*100 representing jwL.
>> z('C',0.2e-6,100) % Returns a value of 1/(j*0.2uF*100r/s) representing 1/jwC.
```

7. Write functions called `aand`, and `aor` that each operate on a single argument that is a logical array of 1's and 0's. If the input array has more than one row and more than one column, `aand` performs an AND function on the numbers in each column and outputs a horizontal vector array. If the input array is a vector (horizontal or vertical), the `aand` performs an AND function on all the numbers in the array. `aor` behaves in a similar fashion but with OR functions.

8. Use your functions from problem 7 to compute the output of a logic circuit of four gates and two inputs. (Your choice of logic circuit.) Show your logic circuit, the input values, the Matlab® command line to compute the output value of the circuit, and the answer you found.

9. Write a function that has input arguments `x` (a row vector) and `n` (a scalar) and outputs a matrix, `xpow`, of the following form:

$$\texttt{xpow} = \begin{bmatrix} \text{x} \\ \text{x.\^{}2} \\ \vdots \\ \text{x.\^{}n} \end{bmatrix}$$

   The function should test for invalid `n`, (i.e., `n < 0`), and return an empty array if `n` is invalid.

10. Write a Matlab® function called `phasor_polar.m` that converts phasors in rectangular form (`a+bj`) to polar form (`[mag, phase]`). This function returns two values, `mag` (magnitude) and `phase` (in radians). The function should accept an array of phasors and return corresponding arrays for `mag` and `phase`.

REF: [1] The Mathworks, Inc, *Matlab® Primer,* Natick, MA: The Mathworks, Inc, 2012.