EECS 627 Pre Lab Assignment
Perl Introduction

1. Introduction

In this Pre Lab assignment you will be introduced to Perl. It is a very powerful language that can be used to write scripts to extract text from files, run automated tasks and many more things. It is used extensively in industry and academia.

Please refer to materials in books (many available at the library) and websites to complete this assignment. One particular website you can use is http://perldoc.perl.org/index-language.html. You will be asked to research several Perl language concepts (Section 2) that will be used in the application assignment (Section3). If you are already familiar with Perl, fell free to skip Section2 and jump in directly to the assignment.

2. Perl concepts

Please research the following Perl concepts. Read relevant material regarding the topic, and try an example on your own to test your understanding of the concept.

a. Syntax: line ending in semicolon, my variables, variable declaration, loop control (for, while, last, next), true/false evaluation, subroutines

b. Variable data types: scalar, arrays, lists, hashes

c. Basic Perl functions: m// (match), s/// (substitute), push, pop, shift

d. I/O Perl functions: file open/close, print

e. Regular expressions: character patterns

f. Command line operands: @ARGV

3. Application assignment

The goal of this assignment is to test your abilities with Perl. It will test concepts of pattern matching, file opening, reading, writing, variable arrays, language control flow and syntax. When doing HSPICE simulations the most useful data is usually obtained by post processing pre-set measurements (e.g. delay of a signal, currents, and voltages). This data is written by default into a *.mt0 file. In this pre-lab, we will use the file prelab.mt0 in *ctools*. The syntax of this file will be explained. The first lines of the file are:

```
$DATA1 SOURCE='HSPICE' VERSION='X-2005.09 '
.TITLE 'Pre Lab example'
bias_qb qb_fall qb_rise q_fall
q_rise temper alter#
0.5500000 1.399960e-09 failed failed
1.400254e-09 25.0000000 1.0000000
0.5510000 1.411731e-09 failed failed
1.411852e-09 25.0000000 1.0000000
```

* The first line is a comment; starts with $.
* The second line just contains the title of the simulation.
* The third and fourth lines contain column labels for each of the variables that were measured.

In this case: bias_qb, qb_fall, qb_rise, q_fall, q_rise, temper, and alter# were measured. The left-most variable (bias_qb) is swept for a series of simulations and is unique for each simulation iteration and each set of measured data. Each pair of lines in the remainder of the file contains measurements for each variable for each simulation iteration.

In order to post-process the data, it must be imported to MATLAB. For this purpose, prelab.mt0 must be parsed such that each column contains the data for a single variable and each row contains the data for a single simulation iteration. Columns should be separated by a comma with a carriage return at the end of each row. Note that many of the values listed in prelab.mt0 are "failed," indicating that the requested measurement could not be made during the HSPICE run. MATLAB cannot process the word "failed," so each occurrence of "failed" must be replaced with "NaN" (not a number).

Your task is to write a Perl script called uniquename_prelab.pl (e.g., bharan_prelab.pl) that takes as input a list of variable names, parses a *.mt0 file to extract the requested variables, and creates a MATLAB-compatible file called prelab.txt listing the data for each of the requested variables. Note that your script should work with any *.mt0 file and should allow any subset of variable names to be requested in any order. The ordering of the columns in prelab.txt should reflect the ordering specified at the command line. The syntax for calling uniquename_prelab.pl should be as follows:

```
perl uniquename_prelab.pl <var1> <var2> … <var n>
```

Suppose the function call looks like:

```
perl bharan_prelab.pl bias_qb qb_fall q_fall
```

The first lines of prelab.txt should be:

```
bias_qb,qb_fall,q_fall
0.5500000,1.399960e-09,NaN
0.5510000,1.411731e-09,NaN
```

Notes:

- As a trick you can use the fact that alter# will always be the last variable measured in any simulation.
- You must note that if the number of variables measured is increased, the number of group of lines per simulation will grow accordingly. You must keep track of the number of lines that the variable labels occupy and use this same information for extracting the values.
- The lines that contain comments or the title information must be ignored.
- You can assume the input file is called "prelab.mt0"
- The output file must be called "prelab.txt"
- The Perl script must be named "uniqname_prelab.pl"
- The file used for the evaluation of your script will be different from the one provided as example so your script must be as generic as possible.
- The Perl script must be fully commented.

4. Deliverables:

Please submit the Perl script file on *Canvas* by **11:59pm on Sunday, January 10th, 2016**. Late submissions will be deducted 25% per day. The file must be named uniqname_prelab.pl and must be ready to run with the command:

```
perl uniqname_prelab.pl <var1> <var2> … <var_n>
```