Homework #1 - Part 2 Programming Assignment (Due Oct. 12)

1. This is the starting point for a simple circuit simulator, *myspice*, that each of you will build. First we address the reading in (*readin*) of a circuit and the setup of the circuit matrix. For this assignment you are required to write the software to stamp the E (VCVS), F (CCCS), and H (CCVS) elements. Test examples are provided as test[1-7].ckt. As you progress with this course it will be your responsibility to add C, L, D, Q, and M elements. C-source code templates are provided that show how R, I, V, and G elements are readin and stamped. Readin code is provided for the other elements. The syntax used for the simulator is as follows:

```
Rname  node node value
Iname  node node value
Vname  node node value
Gname  node node node node value
Ename  node node node node value
Fname  node node vsrc value
Hname  node node vsrc value
Cname  node node value
Lname  node node value
Dname  node node model value <value refers to area factor>
Qname  node node node model value <value refers to area factor>
Mname  node node node node model value value <the two value fields are the W and L of the MOSFET, respectively>
```

where *node* are node names (integers or strings), *value* is a floating-point number, *vsrc* is the name of a voltage source whose current is used as a controlling variable for current controlled sources, and *model* is the name of the model associated with a semiconductor device.

a) Add the code to stamp the E, F, and H elements in the circuit matrix. The circuit matrix is allocated as a dense matrix of size equal to the number of equations + 1. The *Setup<Element>* and *Stamp<Element>* functions should be developed to stamp the contributions for each element. Print the circuit matrices for test cases test[1-7].ckt using the *printMatrix* function. Note that the branch currents are added after the node numbers.

b) Suggest one way to verify that your stamps are correct.
c) Implement the readin, print, setup, and stamp functions of the following two-port descriptions. Your input syntax should be as shown with the definition of the two ports.

1. An ideal transformer \((n\) is the transformer turns ratio): 
   \[
   \begin{align*}
   N & = \text{node node node node n<value>} \\
   v_1 &= n v_2 \\
   i_1 &= -\frac{1}{n} i_2
   \end{align*}
   \]

2. An ideal gyrator \((g\) is the gyration constant): 
   \[
   \begin{align*}
   T & = \text{node node node node g<value>} \\
   i_1 &= -g v_2 \\
   i_2 &= g v_1
   \end{align*}
   \]


d) Implement the readin, print, and stamp functions for an ideal opamp. The terminal characteristics are described below.

\[
\begin{align*}
O & = \text{node node node A} \\
I_1 &= 0 \\
I_2 &= 0 \\
V_3 &= -A(V_1 - V_2)
\end{align*}
\]

e) Provide a test circuit input file for each element in (c) and (d) similar to the test[1-7].ckt files provided for the other example. Name these files testn.ckt, testt.ckt, and testo.ckt. You need to print the circuit matrices for each of these test cases.

f) How have you verified that the circuit matrices in (e) are correct?

g) List at least four error checks that must be performed during readin.