HW06 Python numpy, scipy and sympy

Python has a huge set of libraries that you will find invaluable in your school and career. This assignment introduces you to some basic tools you will use in most of your ECE classes so make sure to do it well and save these files for later to use as templates.

You generally want to define library functions in your code, but you may also want to execute them directly by defining a main function like this:

```
def main():
    print("Hello World!")

if __name__ == "__main__":
    main()
```

To make a python script directly executable you must include the execution command as the first line

```
#!/usr/bin/env python3
```

And you must tell linux to allow it to be executed by giving it executable permission by executing the chmod command

```
chmod +x hw06a.py
```

For this assignment all python scripts MUST be written as function definitions and an executable main function. The functions must all be properly documented using docstrings for full credit. I will check by calling help() on each function.
https://peps.python.org/pep-0257/

In your circuit analysis classes you will always be writing simultaneous sets of equations to find node voltages or currents. This is a pain if done by hand and some equations can only be solved numerically so that is where sympy and scipy come in handy.

1: HW06a.py Solving transcendental equations with scipy:

1. Given the equation for the intrinsic carrier density $n_i^2 = BT^3 exp(-\frac{E_G}{kT})$ find the value of T (in degrees Kelvin) for a given value of ni
    a. Write a python script with a function def FindT(ni, min, max, element) which returns the corresponding value for T
        i. It includes optional arguments min and max to limit the search space for the solution T. Min should default to 1 and max to 1000
        ii. Optional argument element is either “Si” “Ge” or “GeAs”. It should default to “Si” and choose the correct constants based on its value.
        iii. If no solution is found it should raise a ValueError exception
    b. Include in your python script a main function which takes ni min max element as command line arguments and prints the value for T as “T={:.2e}”

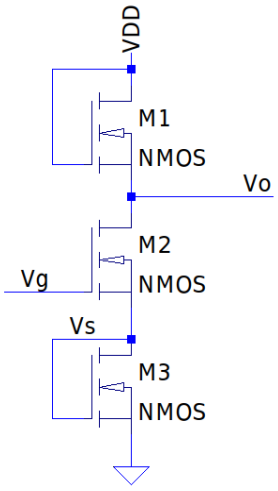
i. Call it from the command line for example:
    1. ./hw06a.py 10e10 100 400 Si
ii. min, max and element are optional and should only be used if provided.
iii. If it fails it should print “No solution”, an explanation of what went wrong and return -1

Here is a table of constants: use Boltzmans constant for eV/K

c	Speed of light (vacuum)	3.0×10^{-10}	$\frac{cm}{s}$
k	Boltzmanns's Constant	8.62×10^{-5}	$\frac{eV}{K}$
k	Boltzmanns's Constant	1.38×10^{-23}	$\frac{J}{K}$
V_T	Thermal voltage at 300K	26	mV
V_T	Thermal voltage	$\frac{kT}{q}$	V
q	Electron charge	1.6×10^{-19}	C
B	Material parameter Si	1.08×10^{31}	$K^{-3}cm^{-6}$
B	Material parameter Ge	2.31×10^{30}	$K^{-3}cm^{-6}$
B	Material parameter GaAs	1.27×10^{29}	$K^{-3}cm^{-6}$
E_G	Bandgap Energy Si	1.12	eV
E_G	Bandgap Energy Ge	0.66	eV
E_G	Bandgap Energy GaAs	1.42	eV

2: hw06b.py solving simultaneous equations using sympy

Given the following 3 simultaneous current equations for this transistor circuit, solve for Id, Vo and Vs



$$I_D = K_n(VDD - V_o - V_{TN})^2 * (1 + \lambda(VDD - V_o))$$

$$I_D = K_n(V_g - V_s - V_{TN})^2 * (1 + \lambda(V_o - V_s))$$

$$I_D = K_n(V_s - V_{TN})^2 * (1 + \lambda V_s)$$

1. A: write a function solve(Kn, VDD, Vg, Vtn, lambda) which returns (Id, Vo, Vs) as a tuple.
    a. It should default Vtn=1 and lambda=0 if they are not given.
    b. Since it is a polynomial function it will have many solutions. Some will be complex some will be real, but invalid, but only one should solve the circuit. Use a for loop to go through the solutions returned by sym.solve and only return teh valid one.
        i. Make sure it only returns real valued answers. Sometimes the value will be reported as complex due to rounding errors even though it is real. So

if it is complex look at the imaginary part and see if it is negligibly small (1+j1e-25) is actually real

ii. Verify that the answers are valid for the given problem.
    1. Since this circuit has no negative supply, Vs, Vo and Ig must be positive.
    2. Since this circuit is powered by a DC supply = VCC, no voltages can be above VCC

iii. If no solution is found it should raise a ValueError exception.

2. Include a main function that takes Kn Vdd Vg Vtn and Lambda as command line arguments and prints the answer as
    a. "Id={:.2e} Vo={:.2e} Vs={:.2e}"
    b. If no valid solution is found, then main program should print "No solution found" and return -1
    c. For example: ./hw06b.py 1e-3 5 3 1 0.1
        i. Should print "Id=2.94e-04 Vo=3.49e+00 Vs=1.51e+00"

3: HW06c.py Creating graphs with matplotlib, reading data from files:

1. Use numpy or pandas to read a spice output file and plot it as the three currents vs time. With the x axis being time and the y axis being the current in mA.
    a. You could also read the data with a simple for loop, but numpy and pandas makes it much easier.
2. The data is in the file "data.correct" in the test directory
3. Label the plot "Spice Output" and the x-axis "time (sec)" and the y-axis "current (mA)"

4: HW06d.py Complex math with scipy or python controls libraries

In circuits, signals and controls you will come across complex numbers and algebra quite often. Scipy and its associated libraries and python controls library will help you make quick work of these jobs.

Circuit responses and physical systems are often modeled using Laplace transforms and the complex "s" domain.

Given a complex polynomial $H(s) = \frac{5(s+5)(s+3)(s+1)}{(s+6)(s+12)(s+5)}$

1. Plot the Bode Plot response from frequencies 10mHz to 10Hz
2. Plot the poles and zeros locations

The input files I will test against are in the "test" directory