Welcome

...to the 2020 Spring semester of CS 1371! Before beginning your first homework, you should take a look at the **CS 1371 Homework Guide Spring 2020.pdf** and **Testing Your Code.pdf** files included in the .zip file for this homework. These documents detail everything you need to know about completing this and future homeworks. The document about testing your code also includes information about testing functions with output types that have not been taught yet. You can ignore this part of the document for now, but will probably want to save it as a reference for later in the semester.

If you have not yet downloaded MATLAB you can click here to download it from GT OIT. Once you have MATLAB installed, and you have read and understood the documents above, you can start on this homework.

Happy coding!
~Homework team

Function Name: `youngsModulus`

Inputs:
1. (*double*) Force
2. (*double*) Area
3. (*double*) Change in length
4. (*double*) Original length

Outputs:
1. (*double*) Young's modulus

Background:

Young's modulus is a mechanical property that measures a material's resistance to atomic separation (more commonly known as stiffness). The higher the Young's modulus, the stiffer the material. You're completing your MSE 2001 homework and decide to use MATLAB to make your life a little easier!

Function Description:

Write a function that outputs the Young's modulus (*E*) given the force (*F*), area (*A*), change in length (*ΔL*), and original length (*L*).

$$Stress = \mathbf{F/A}$$

$$Strain = \Delta L/L_0$$

$$E = \frac{Stress}{Strain}$$

Example:

```
E = youngsModulus(20, 3, 3, 6)
E → 13.33
```

Notes:
- Round your final output to 2 decimal places.

Function Name: `nernstPotential`

Inputs:

1. (*double*) The valence of the ion
2. *(double)* The extracellular concentration
3. *(double)* The intracellular concentration

Outputs:

1. (double) The equilibrium potential for the ion

Background:

The Nernst equation describes the equilibrium potential for any ion□that is, the electrical potential necessary to balance a given ionic concentration gradient across a membrane so that the net flux of the ion is zero.

Function Description:

Given the valence, intracellular concentration, and extracellular concentration of an ion, write a function that outputs the calculated equilibrium potential, rounded **down** to the nearest whole number.

$$E_{\text{ion}} = \frac{61}{Z} \log\left(\frac{C_{\text{out}}}{C_{\text{in}}}\right)$$

where E_{ion} = equilibrium potential for a particular ion, in mV
C_{in} = intracellular concentration of the ion
C_{out} = extracellular concentration of the ion
Z = valence of the ion

Example:

```
eqPot = nernstPotential(1, 5, 150)
eqPot → -91
```

Notes:

- Be sure to use log base 10, `log10()`, and not `log()`, which is natural log, in your code!

Hints:

- The function `floor()` will be useful.

Function Name: `newtonsLaw`

Inputs:

1. (*double*) The initial velocity of an object (m/s)
2. (*double*) The final velocity of an object (m/s)
3. (*double*) The change in time (s)
4. (*double*) The mass of the object (kg)

Outputs:

1. (*double*) The acceleration of the object (m/s^2)
2. (*double*) The force required to accelerate the object (N)

Function Description:

Given the initial velocity and the final velocity of an object, the change in time for the velocity change to take place and the mass of the object, use the following equation to calculate and output the acceleration of the object.

$$acceleration = \frac{final\ velocity - initial\ velocity}{change\ in\ time}$$

Then use the following equation to calculate and output the force required to accelerate the body.

$$force = mass \times acceleration$$

Example:

```
[acc, force] = newtonsLaw(5, 20, 6, 75)
acc → 2.50
force → 187.50
```

Notes:

- Round both outputs to two decimal places. However, you must calculate force **BEFORE** rounding the acceleration value.
- The change in velocity could be negative, and in that case both the acceleration and the required force would also be negative.
- In this problem we are assuming constant acceleration/deceleration.

Function Name: `arrhenius`

Inputs:

1. (*double*) A value of k, the rate constant
2. (*double*) A value of k_o, the pre-exponential factor
3. (*double*) A value of E_a, the activation energy

Outputs:

1. (*double*) The value of T, the temperature of the reaction

Background:

In chemistry, reactions proceed at different rates according to the reaction's rate law. All rate laws have a factor *k*, the rate constant, applied to them. However, this value is not a constant across different temperatures. It is in fact a function of several values inherent to that reaction.

Function Description:

Given values of k, k_o, and E_a. Find the temperature (T) of the reaction. The relation is as follows:

$$k = k_0 e^{\frac{-E_a}{RT}}$$

Example:

```
T = arrhenius(5000000, 500000, -5000)
T → 261.18
```

Notes:

- Use R = 8.314 in this problem.
- Round your answer to 2 decimal places.

Hints:

- You will need to rearrange the equation algebraically to where you have T as a function of k, k_0, and E_a.