AE1205 Assignment 1: International Standard Atmosphere

For this assignment, first check if you understand the example program below:

ABC solver Python code

```
from math import sqrt

print('Solve ax^2 + bx + c = 0')

a = float(input('Enter a:'))
b = float(input('Enter b:'))
c = float(input('Enter c:'))

# Solve equation    (Note: even for rounded numbers, we use 2.0
# (instead of 2) to make sure it is a float)

# Is this a second-order equation?
if not a == 0:
    # or a != 0
    # or (a < 0 or a > 0)

    # Second-order equation: check determinant D
    D = b ** 2.0 - 4.0 * a * c

    # Two solutions:
    if D > 0:
        x1 = (-b - sqrt(D)) / (2.0 * a)
        x2 = (-b + sqrt(D)) / (2.0 * a)
        print('There are two solutions:', round(x1, 3), 'and', round(x2, 3))

    elif D < 0:
        print('There are no solutions')

    else:
        x = -b / (2.0 * a)
        print('There is one solution:', round(x, 3))

else:
    # This is a first-order equation:
    x = -c / b
    print('There is one solution:', round(x, 3))

print('\nReady.')
```

When run, this program results in the following output (with user inputs in red):

ABC solver example output

```
Solve ax^2 + bx + c = 0
Enter a:2
Enter b:3
Enter c:-4
There are two solutions: -2.351 and 0.851

Ready.
```

Study example program

You should be able to answer the following questions before starting the assignment:

- Why is the first line needed in this program?
- Type `help("math")` in the Python shell to see which functions and constants have been imported by the asterisk.
- What does the "`\n`" mean in the pieces of text that are printed?
- What is the meaning of the change of the left margin?
- Under what conditions will the code x1= and x2= be executed?
- What would happen a and b are both equal to zero?
- What does `b**2` mean, so what does the operator ** do? What would **0.25 do?
- What does the `round(x,n)` function do?
- Also note that all numbers, when they are float have a decimal point to signal the Python interpreter that it is indeed a float and not an integer (2.0, 4.0 and 2.)

ISA background information

Before we start with the actual assignment a short reminder of how to calculate the ISA, with the constants you need: There are two types of layers: isotherm, with a constant temperature, and gradient, with a fixed rate of temperature change per altitude, often referred to as a. We always start at sea level, and work our way up until we are in the layer where we need to be. We use the values at the base of a layer, which is the top of the lower, previous layer as zero-values. We first calculate the temperature at our altitude and then we use the ratio with the base temperature to calculate the pressure and finally the gas equation to calculate the density.

The formulae and constants needed are given below:

$$\begin{aligned} g_0 &= 9.80665 m/s^2 \\ R &= 287.0 J/kgK \end{aligned}$$

Sea level values:

$$\begin{aligned} T &= 15^\circ C = 288.15K \\ p &= 101325.0 Pa \end{aligned}$$

Layers for geopotential altitude h:

$$T_1 = T_0 + a(h_1 - h_0)$$

Troposphere	$0 < h <= 11km$:	$a = -6.5K/km = -0.0065K/m$
Tropopause	$11km < h <= 20km$:	isotherm ($a = 0$)
Stratosphere	$20km < h <= 32km$:	$a = +1.0K/km = +0.0010K/m$
Stratosphere	$32km < h <= 47km$:	$a = +2.8K/km = +0.0028K/m$
Stratopause	$47km < h <= 51km$:	isotherm ($a = 0$)
Mesosphere	$51km < h <= 71km$:	$a = -2.8K/km = -0.0028K/m$
Mesosphere	$71km < h <= 86km$:	$a = -2.0K/km = -0.0020K/m$

Formulae for pressure:

Gradient layer: $\frac{p_1}{p_0} = \left(\frac{T_1}{T_0}\right)^{-\frac{g_0}{aR}}$

Isothermal layer: $\frac{p_1}{p_0} = e^{-\frac{g_0}{RT}(h_1-h_0)}$

For both layer types, density with: $\rho = \frac{p}{RT}$

The assignment: make your own basic ISA calculator

In this assignment you will make your own ISA calculator. We will always start at sea level and recalculate the base values every time, so that when somebody changes the value used for g and/or R, which are used, it will still be a continuous atmosphere.

Step 1: Troposphere calculation (h < 11 km)

Using the example program on the front page as inspiration, make a program with the following input/output for the troposphere (so up to 11 km) first (user input in red):

```
    **** ISA calculator troposphere ****

Enter altitude [m]:9000

Temperature: 229.65 K (-43.5 'C)
Pressure : 30736 Pa ( 30 % SL)
Density : 0.4663 kg/m3 ( 38 % SL)

Ready.
```

Also add a check to see whether the user entered a valid altitude:

```
    **** ISA calculator troposphere ****

Enter altitude [m]:15000

Sorry, I can only do altitudes up to 11000 m.

Ready.
```

Step 2: The next layer

As a next step, add one more layer. For this we still first need to calculate the 11 km. Check this:

```python
if h <= 11000.0:
    h1 = h
else:
    h1 = 11000.0
```

is equivalent to this:

```python
h1 = min(h, 11000.0)
```

Similarly, the `max( )` function can be used to set a lower limit: `x = max(x, 0.0)` will apply a lower limit of zero on the variable x (not necessary now). So `min` can be used to set an **upper** limit and `max` for a **lower** limit.

Use the `min`-function in the line above to adapt the calculation of the first layer. After this has been done, check whether the input value h for altitude was actually larger than 11 km, if so, copy the calculated values T_1, p_1, h_1 to the base values p_0, T_0 and h_0 for the next layer. Then calculate in a similar way the isothermal layer (limited to an upper limit of 20 km for h_1). This will overwrite the previous values for T_1, p_1 and h_1 and prepare them as new base values, if we even need to go higher. In this way the program can be extended, even without ever further indenting.

Step 3: Add all layers

Now add more layers in a similar fashion. For the advanced programmers, who have previous knowledge. Use an if-statement, and try to avoid further indenting if it is not needed. Use copy-paste freely.

Step 4: Add a menu

Add a menu to present the user with a choice of altitude inputs as shown below:

ISA calculator menu

```
    *** ISA calculator ***

1. Calculate ISA for altitude in meters
2. Calculate ISA for altitude in feet
3. Calculate ISA for altitude in FL

Enter your choice: 2
```

So based on this choice as entered by the user, first check whether you need to convert the input to meters or not.

```
ft = 0.3048      # [m]
FL = 100.0 * ft # [m]
hinft = float(input("Enter altitude in ft:"))
h   = hinft * ft # Conversion from feet to meters: multiply by one foot
hft = h / ft     # Conversion from meters to ft:
                 # divide how many feet fit in this number of meters
```

If you end your program with the following line, then you can double click the Python file directly from Windows and still read its output before the window closes:

```
dummy = input('Press enter to end the ISA calculator.')
```

Optional extras for advanced programmers

* Add a question whether the user wants to define a different sea level temperature. When he presses enter then the standard ISA value will be used.

*** We still see a lot of repeating code. There are ways to avoid this and make the program shorter. We will look at this later, but maybe you can also find how you could shorten the program using function definitions, lists and a while loop? Make sure you still start from sea level to allow temperature offsets and slightly different constants and to ensure a continuous atmosphere.

***** You could also add options 4 and 5 to the menu which returns the altitude in ft, FL and meters for a given p and density. You can check the answers of option 3 and 4 by entering the altitude in 1 (or 2). And add the required calculations, note how you can combine choice 1 and 2, the only difference is the unit conversion before you start the calculation. Example of a conversion, first define the odd unit in SI units, then multiply when a variable is given in feet or divide when you want to know how many feet fit within the number of meters.