07 Prove Milestone: Lists

Purpose

Prove that you can write a Python program that creates and uses a compound list.

Problem Statement

In chemistry, the molar mass of a substance is the mass in grams of one mole of the substance (grams / mole). A mole is simply a fixed very large quantity, specifically 602,214,076,000,000,000,000,000 (usually written as $6.02214076 \times 10^{23}$). A molar mass calculator is a program that computes the molar mass of a substance and the number of moles of a sample of that substance. To use a molar mass calculator, a chemist enters two inputs:

- The formula for a molecule, such as H_2O (water) or $C_6H_{12}O_6$ (glucose)
- The mass in grams of a sample of the substance, such as 3.71

The calculator computes the molar mass of the molecule by doing the following for each element in the formula:

1. Sum the number of atoms of each element in the formula
2. Find the atomic mass of each element
3. Multiply the number of atoms by their atomic mass
4. Add the product into the molar mass of the molecule

Then the calculator uses this formula to compute the number of moles in the sample:

$$number\_of\_moles = \frac{sample\_mass}{molar\_mass}$$

Finally, the calculator then prints two results for the chemist to see:

- the molar mass
- the number of moles

Example

As an example, consider a sample of glucose ($C_6H_{12}O_6$) with a mass of 12.37 grams. To use a molar mass calculator, a chemist would enter

- C6H12O6
- 12.37 grams

The calculator would compute the molar mass of glucose by doing the following:

1. Sum the number of atoms of each element in the formula for glucose:
   6 carbon atoms
   12 hydrogen atoms
   6 oxygen atoms

2. Find the atomic mass of each element:

Symbol	Name	Atomic Mass
C	Carbon	12.0107
H	Hydrogen	1.00794

Symbol	Name	Atomic Mass
O	Oxygen	15.9994

3. Multiply the number of atoms by their atomic mass:
   6 × 12.0107 = 72.0642
   12 × 1.00794 = 12.09528
   6 × 15.9994 = 95.9964

4. Add the results of the multiplications to get the molar mass of glucose:
   72.0642 + 12.09528 + 95.9964 = 180.15588 grams/mole

Then the calculator would divide the mass of the sample of glucose by the molar mass of glucose which results in the number of moles in the sample:

$$\frac{12.37 \text{ grams}}{180.15588 \text{ grams/mole}} = 0.06866 \text{ moles}$$

The calculator would then print two results for the chemist to see:

- the molar mass of glucose: 180.15588 grams/mole
- the number of moles in the sample: 0.06866 moles

Table of Elements

There are 118 known elements. The symbol, name, and atomic mass of all 118 elements are shown in the following table.

Symbol	Name	Atomic Mass
"Ac",	"Actinium",	227
"Ag",	"Silver",	107.8682
"Al",	"Aluminum",	26.9815386
"Am",	"Americium",	243
"Ar",	"Argon",	39.948
"As",	"Arsenic",	74.9216
"At",	"Astatine",	210
"Au",	"Gold",	196.966569
"B",	"Boron",	10.811
"Ba",	"Barium",	137.327
"Be",	"Beryllium",	9.012182
"Bh",	"Bohrium",	272
"Bi",	"Bismuth",	208.9804
"Bk",	"Berkelium",	247
"Br",	"Bromine",	79.904
"C",	"Carbon",	12.0107
"Ca",	"Calcium",	40.078
"Cd",	"Cadmium",	112.411
"Ce",	"Cerium",	140.116
"Cf",	"Californium",	251
"Cl",	"Chlorine",	35.453
"Cm",	"Curium",	247
"Cn",	"Copernicium",	285
"Co",	"Cobalt",	58.933195
"Cr",	"Chromium",	51.9961

Symbol	Name	Atomic Mass
"Cs",	"Cesium",	132.9054519
"Cu",	"Copper",	63.546
"Db",	"Dubnium",	268
"Ds",	"Darmstadtium",	281
"Dy",	"Dysprosium",	162.5
"Er",	"Erbium",	167.259
"Es",	"Einsteinium",	252
"Eu",	"Europium",	151.964
"F",	"Fluorine",	18.9984032
"Fe",	"Iron",	55.845
"Fl",	"Flerovium",	289
"Fm",	"Fermium",	257
"Fr",	"Francium",	223
"Ga",	"Gallium",	69.723
"Gd",	"Gadolinium",	157.25
"Ge",	"Germanium",	72.64
"H",	"Hydrogen",	1.00794
"He",	"Helium",	4.002602
"Hf",	"Hafnium",	178.49
"Hg",	"Mercury",	200.59
"Ho",	"Holmium",	164.93032
"Hs",	"Hassium",	270
"I",	"Iodine",	126.90447
"In",	"Indium",	114.818
"Ir",	"Iridium",	192.217
"K",	"Potassium",	39.0983
"Kr",	"Krypton",	83.798
"La",	"Lanthanum",	138.90547
"Li",	"Lithium",	6.941
"Lr",	"Lawrencium",	262
"Lu",	"Lutetium",	174.9668
"Lv",	"Livermorium",	293
"Mc",	"Moscovium",	288
"Md",	"Mendelevium",	258
"Mg",	"Magnesium",	24.305
"Mn",	"Manganese",	54.938045
"Mo",	"Molybdenum",	95.96
"Mt",	"Meitnerium",	276
"N",	"Nitrogen",	14.0067
"Na",	"Sodium",	22.98976928
"Nb",	"Niobium",	92.90638
"Nd",	"Neodymium",	144.242
"Ne",	"Neon",	20.1797
"Nh",	"Nihonium",	284
"Ni",	"Nickel",	58.6934
"No",	"Nobelium",	259
"Np",	"Neptunium",	237
"O",	"Oxygen",	15.9994
"Og",	"Oganesson",	294
"Os",	"Osmium",	190.23
"P",	"Phosphorus",	30.973762

Symbol	Name	Atomic Mass
"Pa",	"Protactinium",	231.03588
"Pb",	"Lead",	207.2
"Pd",	"Palladium",	106.42
"Pm",	"Promethium",	145
"Po",	"Polonium",	209
"Pr",	"Praseodymium",	140.90765
"Pt",	"Platinum",	195.084
"Pu",	"Plutonium",	244
"Ra",	"Radium",	226
"Rb",	"Rubidium",	85.4678
"Re",	"Rhenium",	186.207
"Rf",	"Rutherfordium",	267
"Rg",	"Roentgenium",	280
"Rh",	"Rhodium",	102.9055
"Rn",	"Radon",	222
"Ru",	"Ruthenium",	101.07
"S",	"Sulfur",	32.065
"Sb",	"Antimony",	121.76
"Sc",	"Scandium",	44.955912
"Se",	"Selenium",	78.96
"Sg",	"Seaborgium",	271
"Si",	"Silicon",	28.0855
"Sm",	"Samarium",	150.36
"Sn",	"Tin",	118.71
"Sr",	"Strontium",	87.62
"Ta",	"Tantalum",	180.94788
"Tb",	"Terbium",	158.92535
"Tc",	"Technetium",	98
"Te",	"Tellurium",	127.6
"Th",	"Thorium",	232.03806
"Ti",	"Titanium",	47.867
"Tl",	"Thallium",	204.3833
"Tm",	"Thulium",	168.93421
"Ts",	"Tennessine",	294
"U",	"Uranium",	238.02891
"V",	"Vanadium",	50.9415
"W",	"Tungsten",	183.84
"Xe",	"Xenon",	131.293
"Y",	"Yttrium",	88.90585
"Yb",	"Ytterbium",	173.054
"Zn",	"Zinc",	65.38
"Zr",	"Zirconium",	91.224

Assignment

During this lesson and the next lesson, you will write and test a molar mass calculator. During this lesson, you will complete part of the calculator by writing code to create a compound list that contains data for all 118 known elements.

Helpful Documentation

- If you are interested in the chemistry concepts involved in a molar mass calculator, you can watch this Khan Academy video: Calculating molar mass and number of moles (6 minutes).
- The prepare content for this lesson explains how to create and use a compound list.
- The prepare content for lesson 5 explains how to use `pytest`, `assert`, and `approx` to automatically verify that functions are correct. It also contains an example test function and links to additional documentation about `pytest`.

Steps

Do the following:

1. Using VS Code, create a new file and save it as `chemistry.py`
2. In the `chemistry.py` file, write a function named `make_periodic_table` that takes no parameters and creates and returns a compound list. The compound list must contain all the data in the table of elements shown in the Problem Statement section above. The data within the compound list must be organized like this:

```
periodic_table_list = [
    # [symbol, name, atomic_mass]
    ["Ac", "Actinium", 227],
    ["Ag", "Silver", 107.8682],
    ["Al", "Aluminum", 26.9815386],
      ⋮
]
```

   We *strongly recommend* that you *do not type the data* in the table of elements but instead that you use *copy and paste* to copy the data from this assignment into your program. If you don't know how to use copy and paste to help you quickly write the `make_periodic_table` function, ask a fellow student, a tutor, a teaching assistant, or your teacher for help.

3. In the `chemistry.py` file, write the `main` function that takes no parameters and returns nothing. The `main` function should call the `make_periodic_table` function and print the returned table with each individual element on a separate line.
4. At the bottom of your `chemistry.py` file, add a call to the `main` function. Be certain to protect the call to `main` with an `if` statement as taught in the prepare content for lesson 5.

Testing Procedure

Verify that your program works correctly by following each step in this testing procedure:

1. Download the `test_chemistry_1.py` Python file and save it in the same folder where you saved your `chemistry.py` program. Run the `test_chemistry_1.py` file and ensure that the `test_make_periodic_table` function passes. If it doesn't pass, there is a mistake in your `make_periodic_table` function. Read the output from `pytest`, fix the mistake, and run the `test_chemistry_1.py` file again until the test function passes.

```
> python test_chemistry_1.py
====================== test session starts ======================
platform win32--Python 3.8.6, pytest-6.1.2, py-1.9.0, pluggy-0.13.
rootdir: C:\Users\cse111\lesson07
collected 1 item

test_chemistry_1.py::test_make_periodic_table PASSED        [100%]

======================= 1 passed in 0.14s =======================
```

2. Run your `chemistry.py` program and ensure that your program's output matches the following output. The three vertical dots (⋮) in the output below are called a vertical ellipsis and mean the output continues. Your program shouldn't print the vertical ellipsis. Instead, your program should print the rest of the output.

```
> python chemistry.py
['Ac', 'Actinium', 227]
['Ag', 'Silver', 107.8682]
['Al', 'Aluminum', 26.9815386]
['Am', 'Americium', 243]
['Ar', 'Argon', 39.948]
['As', 'Arsenic', 74.9216]
['At', 'Astatine', 210]
['Au', 'Gold', 196.966569]
  ⋮
```

Submission

On or before the due date, return to I-Learn and report your progress on this milestone.