**Exercise 1: The Calendar Module.**

In the following exercises, you will work with Python's calendar module:

1. Visit the Python documentation website at <http://docs.python.org/3.1/modindex.html>, and look at the documentation on the calendar module.
2. Import the calendar module.
3. Read the description of the function isleap(). Use isleap() to determine the next leap year.
4. Find and use a function in the module calendar to determine how many leap years there will be between years 2000 and 2050, inclusive.
5. Find and use a function in module calendar to determine which day of the week July 29, 2016 will be.

**Exercise 2: Alkaline Earth Metals.**

Assign a list that contains the atomic numbers and the names of the six alkaline earth metals---barium (56), beryllium (4), calcium (20), magnesium (12), radium (88), and strontium (38)---to a variable called alkaline\_earth\_metals.

1. Write a one-liner that returns the highest atomic number in alkaline\_earth\_metals.
2. Using one of the list methods, sort alkaline\_earth\_metals in ascending order (from the lightest to the heaviest).
3. Transform the alkaline\_earth\_metals into a dictionary using the name of the metals as the dictionary's key.
4. Create a second dictionary containing the noble gases -- helium (2), neon (10), argon (18), krypton (36), xenon (54), and radon (86) -- and store it in the variable noble\_gases.
5. Merge the two dictionaries and print the result as couples (name, atomic number) sorted in ascending order on the element names.

Note that Python's dictionaries neither preserve the insertion order nor are sorted in some way.

**Exercise 3: Temperature Conversion System.**

Beyond the well-known Celsius and Fahrenheit, there are other six temperature scales: Kelvin, Rankine, Delisle, Newton, Réaumur, and Rømer (Look at:

<http://en.wikipedia.org/wiki/Comparison_of_temperature_scales>

to read about them).

1. Write a function (table) that given a pure number returns a conversion table for it (as a string) among any of the 8 temperature scales (remember that functions are objects as well).
2. Write a function (toAll) that given a temperature in a specified scale returns a string for all the corresponding temperatures in the other scales, the result must be sorted on the temperatures and the scale must be specified.

**Exercise 4: Matrix Calculi.**

A matrix can be represented as a list of lists (rows and columns).

1. Use the comprehensions to define a function (identity) that returns the identity matrix (the one with all 0s but the 1s on the diagonal) of given size.
2. Use the comprehensions to define a function (square) that returns a square matrix filled with the first n\*n integers with n given as an argument.
3. Write the function transpose to transpose a generic matrix independently of the size and content.
4. Write the function multiply to multiply two matrices non necessarily square matrices.

**Exercise 5: Shell Commands Simulation.**

Similarly to the ls-l example please implement:

1. The cat command, i.e., a command that given a list of files prints their content on the terminal (man cat to get more info).
2. The chmod command, i.e., a command that permits to change the access mode of a given group of files (man chmod to get more info)
3. The more command, i.e., a command that given a file prints its content 30 rows at a time and wait a keystroke every 30 rows before printing the next 30.