

## Exercises v.1 [2024-11-04]

**Note: data.zip contains the same files as the data folder, but it's easier to download**

Feel free to use whatever approach to solve these exercises, as long as it's useful to you (nobody is giving out grades, so if you blindly copy-paste a solution from the web you probably won't have achieved anything, besides shortening the life of your Ctrl key).

Difficulty rankings as always are very debatable, and everybody has different strenghts. So instead of classifying the tasks by difficulty I tried to do it based on what they involve. I suggest you start with something that seems interesting and doable, without worrying too much about difficulty.

Some programs are indeed a bit harder to do, and some are a bit lengthier. If you don't have a personal project, I'd be more than satisfied if you were to do one of these programs instead!

Remember that I'm always available to troubleshoot code together.

Good luck!

### Numbers and such

1. Write a function that solves second-degree equations, taking as arguments a list of coefficients.
2. Write a function that decomposes an integer into its prime factors (hint: a simple way for smallish numbers would be to first write a list of primes (hint: you can use the sieve of Eratostenes we saw in lesson 2)).
3. Write a function that converts roman numbers into integers.
4. Find some fractions which value remains the same by deleting the same digit from both numerator and denominator.
5. Imagine an analog clock (one with only two hands, for hours and minutes). Write a function that, given the time in 12-hour format, returns the angle between the hands of the clock. Bonus: what if the clock also had a hand for the seconds?
6. Write a function that returns the number of days between two dates.
7. Are you also annoyed by parents that insist on telling you the age of their children in months? Like *"He's 21 months old..."* instead of *"He's almost 2"*?  
Please, write a function that takes an age in months and tells you the actual age of the baby in years. Bonus: if the baby is nearing its birthday, have the function say *"They'll be XX years old in YY months"* (for example, if

the baby is 23 months old, the function should say “*They’ll be 2 years old in 1 month*”).

8. Create a function that takes an ordered list of coordinates of 2D points in the form `[[x1, y1], [x2, y2], [x3, y3], ...]` and returns the area of the corresponding polygon. It should work with any number of vertices, and also check for degenerate polygons.

## Words and characters

1. Create a function that uses numpy arrays to print a rectangle of digits of a certain width and height. The rectangle should be drawn using the character 0 for the internal space and the character 1 for the sides. For example, this is a 5 x 4 rectangle:

```
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
```

2. You’ve been put in charge of creating SPAM emails for a known online retail store. There’s a base email contained in the file `data/coupon.txt`; you must customize it so that in place of the word CUSTOMERNUMBER there’s the actual customer number. Customer numbers range from 0001 to 1234. Each personalized SPAM message has to be saved in a new txt file called `coupon\_XXXX.txt` (where XXXX is the customer number). Thank you for helping us generate SPAM!
3. *Does it rhyme?* Using the dictionary contained in `data/dictionary.txt` write a function that takes a word and returns all words that end the same (which in reality may or may not rhyme, damned the English language!). The dictionary comes from here.
4. *Is it an anagram?* Using the dictionary contained in `data/dictionary.txt` write a function that takes a word and checks if it’s the anagram of one or more words in the dictionary. The dictionary comes from here.

## Data analysis

### 1. Halloween candy

In 2017, the news and statistics website FiveThirtyEight hosted an online poll trying to find out what Halloween candy people preferred (when they say people they mostly mean North-Americans, or even just people from the USA). Starting with a list of candy types, and randomly matching them up online, they asked people to vote their preferred candy between two alternatives. Results are contained in the folder `data/candy-power-ranking`, and include some info about each candy as well. Load the data into Pandas,

and see if you can find any correlations between candy characteristics and ranking.

## 2. Population growth rate and migration

I read somewhere that population growth in high-income countries would have sunk to negative values, were it not for immigration. A shrinking population globally is good, a shrinking population in a single country often is not (for example, who'll pay your pensions?) What does the data say? In the folder `/data/population\_growth` you'll find data about natural population growth rate (which only considers births and deaths), net migration rate (considering both immigration and emigration), and the income group of each country according to the World Bank. Check if what I read is correct, and compile: a list of countries which population growth rate is saved by immigration; a list of countries which population is still shrinking, even with immigration. Data about population comes from here, and data about World Bank income groups from here. *Hint: pay attention to the units of measurement!*

## 3. Infant mortality due to diarrhoeal diseases and WASH

Unfortunately, to this day diarrhoeal diseases are still one of the leading causes of child mortality worldwide. Fortunately, these diseases are preventable, and things are generally getting better. Using data regarding diarrhea-caused deaths and WASH (WATER, Sanitation and Hygiene), can you find the strongest correlations between mortality and access to safe water and facilities? You'll find everything in folder `data/diarrhoeal\_diseases/`.

Data regarding diseases comes from the Institute for Health Metrics and Evaluation, and data about clean water and sanitation comes from here (which takes it from the JMP), slightly simplified by me.

# Games

## 1. Memory

Create a program that lets you play memory, alone or against a friend. The program should randomly “place” 32 pairs of cards face down (in a 8x8 grid), then let you choose one card to reveal, show it to you, and ask you for a second one. If they are the same, they should be removed from the grid and you should be awarded one point. Card faces can be whatever you want: numbers, colors, images, letters. You can do it with actual pictures, but I think it would already be very good doing a text-based version.

## 2. Battleship

Create a program that lets you play battleship against a friend, by exchanging CSV files of zeros and ones indicating, respectively, the position of water and ships. The program should be able to tell how many (and which) battleships are still in play, and should ask you for a coordinate to

strike on, replying with one of the following: “miss”, “hit”, “hit and sunk”.