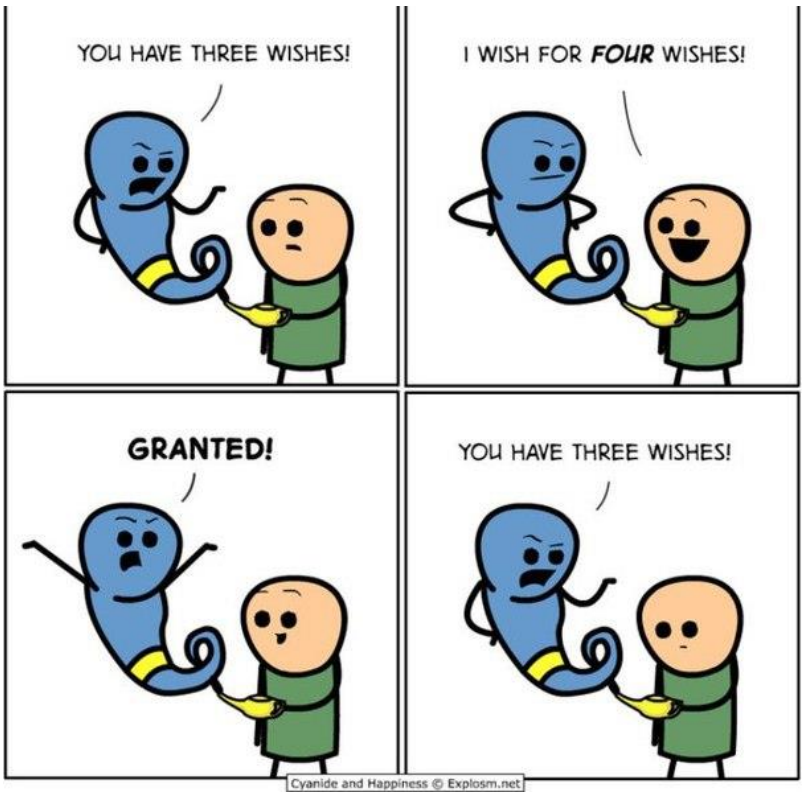
La Plateforme

Day 4 – Recursion

To understand recursion, you must first understand recursion - Stephen Hawking



Job 4.0

Create a program that asks the user to fill in a whole number. Your program will have to calculate **the factorial** of this number, without using any function other than your own. Attention, you must not use **while, for, foreach or ... loop**. Only **recursion**.

Job 4:1

Create a program that asks the user to fill in a whole number. Your program will have to calculate x^n, where n is the number provided by the user, without using any function other than your own. Attention, you must not use **while, for, foreach or ... loop**. Only **recursion**.

Job 4.2

Create a program that models a game board, square, of n x n boxes. Place on this board **n checkers** of chess, so that no checker can "take", when possible. The value of n is entered by the user. When possible, the program will have to display in the terminal the game board with the character 'O' for empty boxes and the character 'X' to represent checkers.

```
Saisir un entier (taille du tableau): 8
X O O O O O O O
O O O O O O X O
O O O O X O O O
O O O O O O O X
O X O O O O O O
O O O X O O O O
O O O O O X O O
O O X O O O O O
```

Job 4:3

Create a program that opens the maze.mz file and links the maze entrance (top left) to its output (bottom right). The program must display the maze in a "maze-out.mz" file or the boxes to follow to reach the output are represented by 'X'. The path should be as short as possible.

```
XX##################
#X#.#.#.............#
#X#.#.#.#.#######.###
#X....#.#.#...#.....#
#X#########.#######.#
#X....#.#.#.#.#.#...#
#X###.#.#.#.#.#.#.#.#
#X#.#.#.....#.....#.#
#X#.###.#########.###
#X..#.#.#.#...#.....#
#X###.#.#.#.#.#.###.#
#X....#.....#.....#.#
#X#####.###.#########
#XXX..#XXX#...#.#...#
#.#X###X#X#####.###.#
#.#X..#X#XXXXX......#
#.#X###X#####X#######
#.#XXXXX..#..X......#
#.###.#######X#######
```

Job 4.4

Write a program that asks the user to provide a first string, then a second string. The program displays 1 if the 2 channels are identical or 0 if the channels are not identical. Strings consist only of **lowercase letters**. The second character string can contain one or more ' * '. Each ' * ' can replace 0 or more characters. For example, if string 1 is "theplatform" and string 2 is "lap*", the program displays 1 because the ' * ' replaces ' lateform '. If string 1 is "theplatform" and string 2 is "l*a*pla*te*form***e" the program returns 1 because the ' * ' does not replace anything.

Rendering

In your github directory "**runtrack-python**", create a folder "**day04**" and in this folder, for each job, a folder "**jobXX**" where XX is the number of the job.
Don't forget to submit your changes as soon as a step is advanced or completed and use explicit comments.

Competencies targeted

- Understanding and applying recursion
- Go further in python...

Knowledge Base

- The Principles of Recursion
  Demystifying recursion in python
- Recursive functions
  Recursive function tutorial