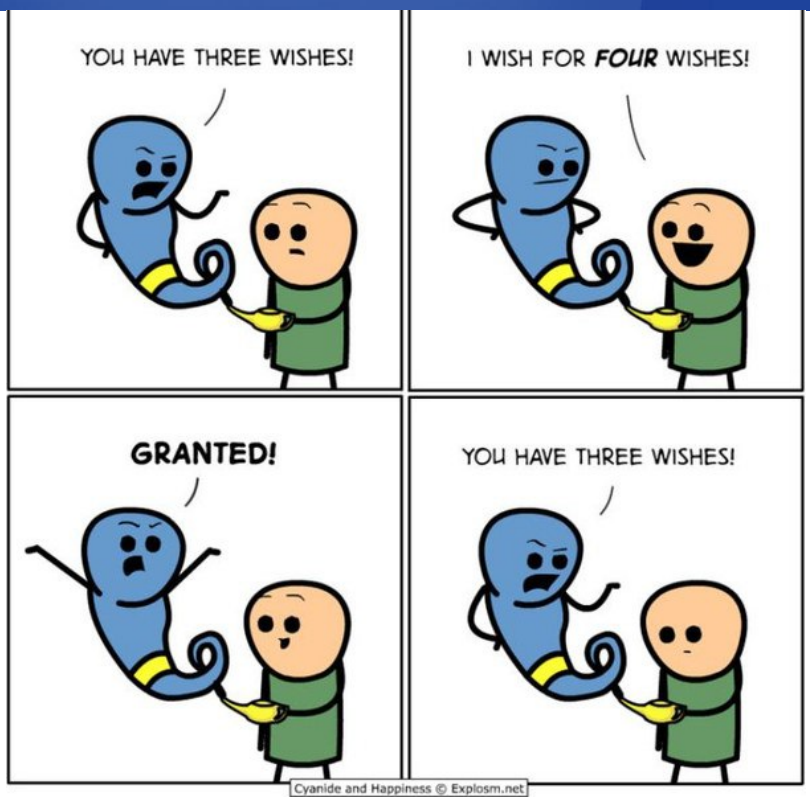
La Plateforme

Jour 4 - Récursivité

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



Job 0

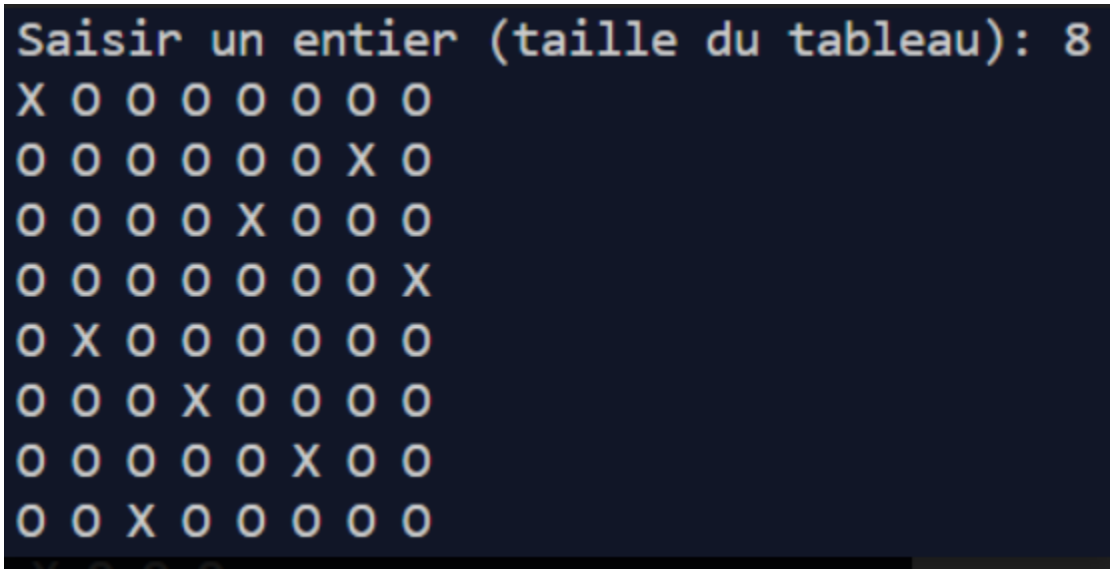
Créer un programme demandant à l'utilisateur de renseigner un nombre entier. Votre programme devra calculer **la factorielle** de ce nombre, sans utiliser de fonction autre que les vôtres. Attention, vous ne devez utiliser **ni while, ni for, ni foreach ni … boucle**. Seulement de la **récursivité**.

Job 0.1

Créer un programme demandant à l'utilisateur de renseigner un nombre entier. Votre programme devra calculer x^n, où n est le nombre fourni par l'utilisateur, sans utiliser de fonction autre que les vôtres. Attention, vous ne devez utiliser **ni while, ni for, ni foreach ni … boucle**. Seulement de la **récursivité**.

Job 03

Créer un programme qui modélise un plateau de jeu, carré, de n x n cases. Placez sur ce plateau **n dames** de jeu d'échecs, de manière à ce qu'aucune dame ne puisse se "prendre", quand cela est possible. La valeur de n est renseignée par l'utilisateur. Quand cela est possible, le programme devra afficher dans le terminal le plateau de jeu avec le caractère 'O' pour les cases vides et le caractère 'X' pour représenter les dames.



Job 08

Créer un programme qui ouvre le fichier maze.mz et qui relie l'entrée du labyrinthe (en haut à gauche) a sa sortie (en bas à droite). Le programme doit afficher le labyrinthe dans un fichier “maze-out.mz” ou les cases à suivre pour atteindre la sortie sont représentées par des ‘X’. Le chemin doit être le plus court 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#######
#.#.........#XXXXXXXX
####################X
```

Job 15

Écrire un programme qui demande à l’utilisateur de fournir une première chaîne de caractères, puis une seconde. Le programme affiche 1 si les 2 chaines sont identiques ou 0 si les chaînes ne sont pas identiques. Les chaînes ne sont constituées que de **lettres minuscules**. La deuxième chaîne de caractères peut contenir un ou plusieurs ‘ * ‘. Chaque ‘ * ‘ peut remplacer 0 ou plusieurs caractères. Par exemple, si la chaîne 1 est “laplateforme” et la chaine 2 “lap*”, le programme affiche 1 car l’ ‘ * ‘ remplace ‘ lateforme ‘. Si la chaîne 1 est “laplateforme” et la chaîne 2 “l*a*pla*te*form***e” le programme renvoie 1 car les ‘ * ‘ ne remplace rien.

Rendu

Dans votre répertoire github “**runtrack-python**”, créez un dossier “**jour04**” et dans ce dossier, pour chaque job, un dossier “**jobXX**” où XX est le numéro du job.
N’oubliez pas d’envoyer vos modifications dès qu’une étape est avancée ou terminée et utilisez des commentaires explicites.

Compétences visées

- Comprendre et appliquer la récursivité
- Aller plus loin en python…

Base de connaissances

- Les principes de la Récursivité
  Démystification de la récursivité en python
- Fonctions récursives
  Tutoriel de fonction récursive