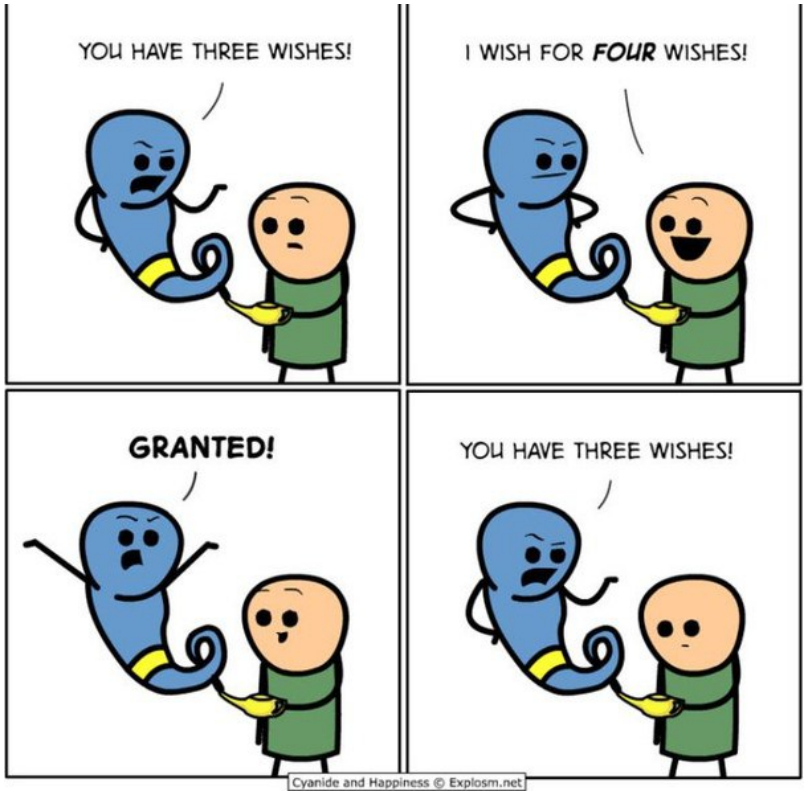
La Plateforme

Jour 5 : La récursivité

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



Job 1

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 2

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 3

Créer une fonction qui prend en paramètre une chaine de caractère. Écrire une fonction qui permet de retourner sa longueur, sans utiliser de fonction système . Attention, vous ne devez utiliser **ni while, ni for, ni foreach**

Job 4

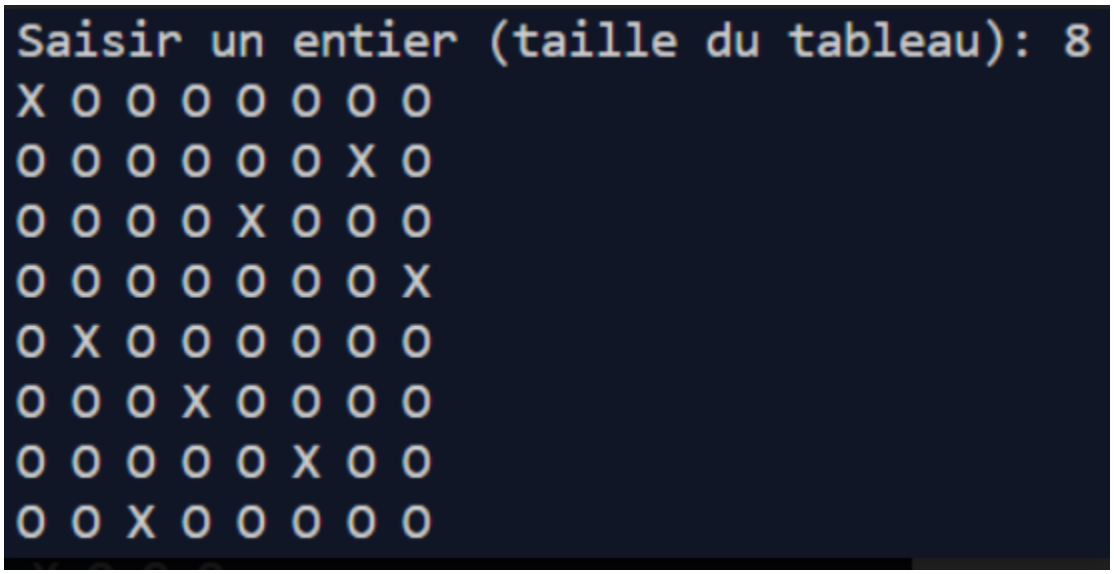
Écrire une fonction récursive permettant de retourner le plus grand chiffre d'une liste.

Job 5

Écrire une fonction récursive qui calcule le n-ième nombre de la suite de Fibonacci.
La suite de Fibonacci est une suite de nombres où chaque nombre est la somme des deux nombres précédents. Elle commence par 0 et 1, et les premiers termes sont donc : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, etc.
Attention, vous ne devez utiliser **ni while, ni for, ni foreach,ni ... boucle**. Seulement de la **récursivité**.

Job 6

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 7

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

Le projet est à rendre sur https://github.com/prenom-nom/runtrack-python-poo. Pour chaque jour, créer un dossier nommé “jourXX” ou XX est le numéro du jour et pour chaque job, créer un fichier “jobXX” ou 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.
Pensez à donner les droits sur le répertoire à **deepthoughtlaplateforme** !

Compétences visées

- Maîtriser la récursivité

Base de connaissances

- La récursivité
- Utiliser la récursivité