FONCTIONS - EXERCICES

1. RETOUR SUR LE PH:

fichier 4-pH.py

Le code élaboré dans les exercices de la partie 2 a besoin d´être amélioré...

```
pH=float(input("Entrez le pH: "))
if pH<=0 or pH>=14:
    print("Valeur impossible, recommencez...")
elif pH<7:
    print("La solution est acide")
elif pH==7:
    print("La solution est neutre")
else:
    print("La solution est basique")
```

Modifiez-le afin que l´utilisateur soit invité à recommencer la saisie du pH:

- en cas d´entrée non valide (ValueError)
- en cas de saisie de pH en dehors de l´intervalle $[0,14]$

2. SOMAVAMOS:

fichier 4-palindrome.py

Si on oublie l´accent, ce mot est un palindrome. Il peut se lire indifféremment de gauche à droite ou de droite à gauche en gardant le même sens.

Rédigez le programme qui permet de tester si un mot entré par l´utilisateur est un palindrome.

- Définir tout d´abord la fonction **is_palindrome(x)** avec un argument d´entrée, la saisie utilisateur.
- Dans le corps du programme, écrire le code qui permet de définir la saisie utilisateur, l´appel de la fonction et l´affichage du résultat.

Rappel: si x est un variable de type string, x[::-1] permet de renverser la chaine de caractères.

3. FACTORIELLE ET NOMBRE e

fichiers 4-factorial.py et 4-euler.py

La factorielle d´un nombre entier n est le produit de tous les entiers de 1 à n soit:

$$n! = 1 \times 2 \times 3 \times \ldots \times n$$

a) Définissez la fonction **factorielle**(n) permettant d´obtenir la factorielle d´un nombre entier n. On pourra utiliser une boucle for. Dans le corps du programme nous devrons:

- Gérer les entrées utilisateur et ses éventuelles erreurs de saisie.
- Afficher le résultat.

b) Cette méthode fonctionne également:

```
def factorielle(n) :
    # Retourne le produit des entiers de 1 à n
    if n <= 1:
        return 1
    else:
        return n * factorielle(n - 1)
```

Testez et décrivez la particularité de cette méthode dite "récursive".

c) Le nombre **e** ne connaît pas la célébrité du nombre π. Pourtant on lui trouve de très nombreuses ressemblances. Comme son congénère, *e* est un nombre irrationnel, c'est à dire qu'il s'écrit avec un nombre infini de décimales sans suite logique.
Ses premières décimales sont: **e = 2,7182818284 5904523536 0287471352 662497....**

Dans *« Introductio in Analysin infinitorum »* publié en 1748, le mathématicien suisse Leonhard EULER montre que l´on peut obtenir une valeur approchée de e en utilisant la formule:

$$e = \sum_{i=0}^{n} \frac{1}{i!}$$

En utilisant la fonction **factorielle**(n) rédigez dans le corps du programme le code permettant d´obtenir une valeur approchée de e.

L´utilisateur choisit la valeur de n. Plus n est grand, plus la valeur de **e** obtenue est précise.

Pensez à gérer les erreurs de saisie.