

TP Informatique 6

Évaluation automatique de l'exercice 1 (20 minutes)

Télécharger le fichier TP06_exo1 sur cahier de prépa, l'ouvrir et le compléter. Pour cela, supprimer l'instruction `pass` et compléter les fonctions.

Au bout de 20 minutes au maximum, vous déposerez votre fichier dans le répertoire prévu sur cahier de prépa. **Le fichier déposé doit pouvoir être exécuté sans message d'erreur.** Puis vous poursuivrez le sujet de TP.

Exercice 1

Vous respecterez l'utilisation d'une boucle conditionnelle `while` ou inconditionnelle `for` imposée dans chaque question.

1. On souhaite écrire une fonction `somme(n)` renvoyant la valeur de la somme $\sum_{k=0}^n k$. On propose :

```
def somme(n):  
    """n : entier"""  
    k,res=0,0  
    while k<n:  
        res+=k  
        k+1  
    return res
```

Identifier la (ou les) erreur(s) de ce code et écrire une version corrigée de la fonction `somme`. Cette fonction utilisera nécessairement une boucle `while`.

2. En vous inspirant du code précédent (toujours avec une boucle `while`), écrire une fonction `fact(n)` renvoyant $n!$.
3. Écrire une fonction `somme1(n)`, avec boucle conditionnelle, d'argument n un entier et qui renvoie $\sum_{0 \leq 2k+1 \leq n} (2k+1)^2$.
4. Écrire une fonction `geom(n,q)`, avec boucle inconditionnelle, d'arguments n un entier et q un flottant et qui renvoie la valeur de $\sum_{k=0}^n q^k$. On veillera à calculer efficacement les puissances successives de q^k en s'appuyant sur la valeur de q^{k-1} déjà calculée.

Exécuter `pytest` (rappel)

1. Lancer l'invite de commande (`cmd` dans la recherche Windows).
2. Se placer dans le répertoire où se trouve votre fichier `.py` en tapant `U:` puis `cd` suivi du chemin.
3. Exécuter le fichier `python nom_du_fichier.py`.
4. Exécuter `pytest` en tapant `pytest nom_du_fichier.py`. On rappelle que si `pytest` n'existe pas, on peut l'installer en tapant `pip install pytest`.

Exercice 2

On définit la suite $(w_n)_n$ par

$$\forall n \in \mathbb{N} \quad w_0 = \frac{\pi}{2} \quad w_1 = 1 \quad \text{et} \quad \forall n \in \mathbb{N} \quad w_{n+2} = \frac{n+1}{n+2} w_n$$

1. Écrire une fonction `w1(n)`, avec boucle conditionnelle, d'argument `n` un entier et qui renvoie la valeur de w_n .
2. Écrire une fonction `test_w1()` utilisant l'instruction `assert` testant la fonction `w1` pour différentes valeurs de n . On rappelle qu'il faut proposer un jeu de tests suffisants pour tester tous les blocs constituant la fonction `w1`.
3. Écrire une fonction `w2(n)`, avec boucle inconditionnelle, d'argument `n` un entier et qui renvoie la valeur de w_n .
4. Écrire une fonction `test_w2()` utilisant l'instruction `assert` testant la fonction `w2` pour différentes valeurs de n .
5. On donne la fonction `w3(n)` d'argument `n` un entier et qui renvoie la valeur de $\int_0^{\frac{\pi}{2}} (\sin t)^n dt$:

```
import numpy as np, scipy.integrate as integr
def w3(n):
    return integr.quad(lambda t:np.sin(t)**n,0,np.pi/2)[0]
```

Afficher les 20 premiers termes des suites codées par `w1`, `w2` et `w3`. Qu'observe-t-on ?

Exercice 3

On a
$$\sum_{k=0}^n k = \frac{n(n+1)}{2} \xrightarrow{n \rightarrow \infty} +\infty$$

Par conséquent, on peut définir

$$\forall p \in \mathbb{R} \quad \varphi(p) = \min \left\{ n \in \mathbb{N} \mid \sum_{k=0}^n k \geq p \right\}$$

1. Écrire une fonction `seuil(p)` d'argument `p` un flottant et qui renvoie le couple $\left(\sum_{k=0}^{\varphi(p)} k, \varphi(p) \right)$.
2. Écrire une fonction `yest_seuil()` utilisant l'instruction `assert` testant la fonction `seuil` pour différentes valeurs de p .

Exercice 4

Dans l'éditeur, effectuer l'importation `import numpy as np`.

1. Écrire une fonction `somme_log(n)`, avec boucle conditionnelle, d'argument `n` un entier et qui renvoie la valeur de $\sum_{1 \leq k, k \ln k \leq n} k$. On rappelle que la fonction `ln` correspond à la fonction `log` dans `numpy`.
2. Peut-on écrire une version avec boucle inconditionnelle ? Si oui, le faire en nommant cette fonction `somme_log_bis(n)`