`Python` td 2 - Sommes, suites, séries

I. Les fonctions `np.prod` et `np.sum` , `np.cumprod`, `np.cumsum`

Rappelons que l'on doit charger la librairie `numpy` : `import numpy as np`

1. **Factorielle**

   Rédiger un script qui fasse les étapes suivantes :

   - Il demande un entier naturel n à l'utilisateur
   - Il fabrique un tableau A contenant les n premiers entiers naturels non nuls, puis affiche ce tableau.
   - Il calcule puis affiche les valeurs de $n!$ et de $\sum_{k=0}^{n} k$.
   - Rajouter ensuite les commandes `P=np.cumprod(A)` et `S=np.cumsum(A)`, puis afficher les valeurs de P et S. Que constate-t-on?

2. **Produit des n premiers entiers naturels impairs.**

   Ecrire un script qui effectue successivement les tâches suivantes.

   - Il saisit un entier naturel n.
   - A l'aide d'une boucle for, il calcule puis affiche la valeur de $P = \prod_{k=0}^{n-1}(2k+1)$, puis la valeur de $S = \sum_{k=0}^{n-1}(2k+1)$.
   - Rajouter ensuite les commandes `A=np.prod(np.arange(1,2*n,2))` et `B=np.sum(np.arange(1,2*n,2))`. Que constate-t-on?
   - Calculer (sur feuille !!) $\prod_{k=0}^{n-1}(2k+1)$ et $\sum_{k=0}^{n-1}(2k+1)$ en fonction de n.
   - Vérifier le résultat à l'aide de Python.

II. Calculs de sommes

1. On considère la suite $(v_n)_{n\in\mathbb{N}^*}$ définie par : $\forall n \in \mathbb{N}^*,\ v_n = \sum_{k=1}^{n} k^2$

   Ecrire un script qui demande un entier naturel n, puis affiche la valeur de v_n (sans utiliser la formule de votre cours). En utilisant la formule de votre cours, vérifier que le résultat obtenu est cohérent.

2. On considère la suite $(u_n)_{n\geq 2}$ définie par: $\forall n \geq 2,\ \ u_n = \sum_{k=2}^{n} \ln\left(1 - \dfrac{1}{k}\right)$

   (a) Ecrire un script qui demande un rang n, puis qui affiche les valeurs de u_2 , u_3 ...u_n .
   (b) Calculer "à la main" u_n en fonction de n .
   (c) Vérifier la cohérence des résultats.

III. Suite récurrente linéaire d'ordre 2

On considère la suite récurrente $(u_n)_{n\in\mathbb{N}^*}$ définie par:

$$u_1 = 0,\ \ u_2 = 9 \ \text{ et } \ \forall n \in \mathbb{N}^* \quad u_{n+2} = 6u_{n+1} - 9u_n.$$

1. Ecrire un programme qui demande le rang n à l'utilisateur puis affiche la valeur de u_n.
2. Calculer u_n en fonction de n. Vérifier les résultats donnés par l'ordinateur.

IV. Somme double

1. Ecrire un programme qui demande un entier naturel n à l'utilisateur , puis qui calcule

$$S_n = \sum_{k=1}^{n}[\sum_{j=k}^{n} \frac{k}{j+1}]$$

2. Exprimer "à la main" S_n en fonction de n puis faire des vérifications avec les résultats donnés en 1. par l'ordinateur.

V. Edhec 2016

On considère la fonction f définie sur $\mathbb{R}_+^*$ par : $\forall x \in \mathbb{R}_+^*,\ f(x) = \frac{e^{-x}}{x}$.
On considère également la suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_0 = 1$ et par la relation $u_{n+1} = f(u_n)$, valable pour tout entier naturel n.

1. (a) Dresser le tableau de variation de f, limites comprises.
   (b) Vérifier que chaque terme de la suite $(u_n)_{n\in\mathbb{N}}$ est parfaitement défini et strictement positif.

2. Les scripts suivants renvoient, pour celui de gauche, la valeur 5, et celui de droite, la valeur 6. Que sait-on de u_5 et u_6 ? Quelle conjecture peut-on émettre sur le comportement de la suite $(u_n)_{n\in\mathbb{N}}$?

```
u=1
n=0
while u>0.00001:
    u=np.exp(-u)/u
    n=n+1
print(n)
```

```
u=1
n=0
while u<100000:
    u=np.exp(-u)/u
    n=n+1
print(n)
```

3. (a) Étudier les variations de la fonction g définie sur $\mathbb{R}_+$ par :
   $$\forall x \in \mathbb{R}_+,\ g(x) = e^{-x} - x^2$$
   (b) En déduire que l'équation $f(x) = x$, d'inconnue x, possède une seule solution, que l'on notera α, sur $\mathbb{R}_+^*$.
   (c) Montrer que $\frac{1}{e} < \alpha < 1$.

4. (a) Établir les deux inégalités : $u_2 > u_0$ et $u_3 < u_1$.
   (b) En déduire les variations des suites $(u_{2n})_{n\in\mathbb{N}}$ et $(u_{2n+1})_{n\in\mathbb{N}}$.

5. On pose : $h(x) = \begin{cases} (f\circ f)(x) & \text{si } x > 0 \\ 0 & \text{si } x = 0 \end{cases}$

   (a) Déterminer $h(x)$ pour tout réel x strictement positif et vérifier que h est continue en 0.
   (b) Résoudre l'équation $h(x) = x$, d'inconnue x élément de $\mathbb{R}_+$.
   (c) En déduire la limite de la suite $(u_{2n+1})_{n\in\mathbb{N}}$.
   (d) Montrer par l'absurde que la suite $(u_{2n})_{n\in\mathbb{N}}$ diverge puis déterminer $\lim\limits_{n\to+\infty}(u_{2n})$.

VI. Une série alternée

Notons $(u_n)_{n\in\mathbb{N}^*}$ la suite définie par : $\forall n \in \mathbb{N}^* \quad u_n = \dfrac{(-1)^n \ln(n)}{n}$

et la fonction f définie par : $\forall x \in [e, +\infty[\quad f(x) = \dfrac{\ln(x)}{x}$.

Notons : $\forall n \in \mathbb{N}^*, \ S_n = \sum_{k=1}^n u_k, \ x_n = S_{2n}$ et $y_n = S_{2n+1}$

1. Montrer que f est décroissante sur $[e, +\infty[$.
2. (a) Ecrire un script en Python qui saisit un entier naturel n, puis qui calcule et affiche la valeur de S_n en utilisant une boucle for.
   (b) Ecrire un script en Python qui saisit un entier naturel n, puis qui calcule et affiche la valeur de S_n sans utiliser de boucle for.
3. (a) Montrer que les suites $(x_n)_{n\in\mathbb{N}^*}$ et $(y_n)_{n\in\mathbb{N}^*}$ sont adjacentes.
   (b) En déduire que la série de terme général u_n est convergente et justifier que :
$$\forall n \in \mathbb{N}^*, \quad S_{2n+1} \leq \sum_{k=1}^{+\infty} \frac{(-1)^k \ln(k)}{k} \leq S_{2n}$$
4. Déterminer, en utilisant Python, une valeur approchée de $\displaystyle\sum_{k=1}^{+\infty} \frac{(-1)^k \ln(k)}{k}$ à 10^{-3} près.

VII. Formule des rectangles

On note $\forall t \in [0,1] \quad f(t) = \dfrac{1}{1+t^3}$. On va s'intéresser à l'intégrale suivante $I = \displaystyle\int_0^1 f(t)dt$.

1. Rappeler la formule des rectangles (ou sommes de Riemann) pour une fonction définie sur un intervalle $[a,b]$.
2. Dresser le tableau de variation de f sur $[0,1]$.
3. Justifier que : $\forall n \in \mathbb{N}^*, \forall k \in [[0, n-1]]$, $\dfrac{1}{n} f\left(\dfrac{k+1}{n}\right) \leq \displaystyle\int_{\frac{k}{n}}^{\frac{k+}{n}} f(t)\ dt \leq \dfrac{1}{n} f\left(\dfrac{k}{n}\right)$
4. En déduire que $\forall n \in \mathbb{N}^*$,
$$\frac{1}{n}\sum_{k=1}^{n} f\left(\frac{k}{n}\right) \leq \int_0^1 f(t)dt \leq \frac{1}{n}\sum_{k=0}^{n-1} f\left(\frac{k}{n}\right)$$
5. Ecrire un script donnant une valeur approchée de I à 10^{-4} près.
6. *Des exercices complémentaires sur les sommes de Riemann*

   Pour chacune des suites suivantes, démontrer sa convergence et calculer sa limite:

   a. $u_n = \sum_{k=1}^n \dfrac{1}{\sqrt{kn+n^2}}$ **b.** $v_n = \sum_{k=1}^{n-1} \dfrac{k}{n^2} \sin\left(\dfrac{k\pi}{n}\right)$ **c.** $w_n = \sum_{k=1}^n \dfrac{1}{2n+3k}$