

Informatique tronc commun – TP n° 4

18 octobre 2018

1. **Lisez attentivement tout l'énoncé avant de commencer.**
2. Commencez la séance en créant un dossier au nom du TP dans le répertoire dédié à l'informatique de votre compte.
3. Après la séance, vous devez rédiger un compte-rendu de TP et l'envoyer au format électronique à votre enseignant.
4. Ce TP est à faire en binôme, vous ne rendrez donc qu'un compte-rendu pour deux.
5. Ayez toujours un crayon et un papier sous la main. Quand vous réfléchissez à une question, utilisez les !
6. Vous devez être autonome. Ainsi, avant de poser une question à l'enseignant, merci de commencer par :
 - relire l'énoncé du TP (beaucoup de réponses se trouvent dedans) ;
 - relire les passages du cours¹ relatifs à votre problème ;
 - effectuer une recherche dans l'aide disponible sur votre ordinateur (ou sur internet) concernant votre question.

Il est alors raisonnable d'appeler votre enseignant pour lui demander des explications ou une confirmation !

Le but de ce TP est d'apprendre à écrire des fonctions évoluées en **Python**.

Instructions de rendu

Attention : suivez précisément ces instructions. Votre fichier portera un nom du type

`tp04_durif_kleim.py`,

où les noms de vos enseignants sont à remplacer par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe, ni majuscule. Dans ce fichier, vous respecterez les consignes suivantes.

- Écrivez d'abord en commentaires (ligne débutant par #), le titre du TP, les noms et prénoms des étudiants du groupe.
- Commencez chaque question par son numéro écrit en commentaires.
- Les questions demandant une réponse écrite seront rédigées en commentaires.
- Les questions demandant une réponse sous forme de fonction ou de script respecteront pointilleusement les noms de variables et de fonctions demandés.

1. Dans le cas fort improbable où vous ne vous en souviendriez pas.

Exercices

Q1 Dans le jeu de la bataille navale, on représente chaque case par un couple d'entiers entre 0 et 9.

Un navire a ses extrémités sur les cases **a** et **b**. Un joueur tire sur la case **x**.

Écrire une fonction `touche(a,b,x)` qui renvoie un booléen indiquant si le navire est touché ou non.

Q2 Un banquier vous propose un prêt de 400 000 euros sur 40 ans «à 3% par an» — ce qui, dans le langage commercial des banquiers, veut dire 0,25% par mois — avec des mensualités de 1431,93 euros. Autrement dit, vous contractez une dette de 400 000 euros. Chaque mois, cette dette augmente de 0,25% puis est diminuée du montant de votre mensualité. À la fin des 40×12 mensualités, il ne vous reste plus qu'à vous acquitter d'une toute petite dette, que vous remboursserez aussitôt.

a) Écrire une fonction `reste_a_payer(p, t, m, d)` renvoyant le montant de cette somme à rembourser immédiatement après le paiement de la dernière mensualité, où p est le montant total du prêt en euros (dans l'exemple, 400 000), t son taux mensuel (dans l'exemple, $0,25 \times 10^{-2}$), m le montant d'une mensualité en euros (dans l'exemple, 1431,93) et d la durée en années (dans l'exemple, 40).

Indice : dans le cas donné dans cet énoncé, vous devez trouver un montant restant d'un peu moins de 7,12 euros.

b) Écrire une fonction `somme_totale_payee(p, t, m, d)` renvoyant la somme totale (mensualités plus le dernier paiement) que vous aurez payé au banquier.

c) Écrire une fonction `cout_total(p, t, m, d)` renvoyant le coût total du crédit, c'est-à-dire le total de ce que vous avez payé moins le montant du prêt.

Q3 Un banquier vous propose de vous prêter p euros, à un taux de $12t\%$ par an — ce qui, dans le langage commercial des banquiers, veut dire $t\%$ par mois — avec des mensualités de m euros. Autrement dit, vous contractez une dette de p euros. Chaque mois, cette dette augmente de $t\%$ puis est diminuée du montant de votre mensualité. Lorsque votre dette, augmentée du taux, est inférieure à la mensualité, il suffit de régler le solde en une fois.

Écrire une fonction `duree_mensualite(p,t,m)` renvoyant le nombre de mensualités nécessaires au remboursement total du prêt.

Attention : que se passe-t-il si la mensualité est trop petite ?

Indice : dans le cas où le prêt est $p = 4 \times 10^5$, le taux est $t = 0,25 \times 10^{-2}$ et la mensualité est $m = 1431,93$, on trouvera une durée de remboursement de 480 mois.

Q4 Écrire une fonction `comb(p,n)` renvoyant $\binom{n}{p}$ (nombre de combinaisons de p éléments parmi n). On pourra bien entendu introduire une fonction auxiliaire².

2. C'est-à-dire, comme l'indique l'étymologie, une autre fonction dont le but sera de vous aider à répondre à la question.

Q5 On appelle suite de Fibonacci la suite F définie par $F_0 = 0$, $F_1 = 1$ et pour tout $n \in \mathbb{N}$, $F_{n+2} = F_{n+1} + F_n$.

Écrire une fonction `fib(n)` calculant et renvoyant la valeur de F_n .

Pensez à vérifier le résultat de votre fonction en 0, 1 et en 5 (vous calculerez à la main F_5 *avant* de le faire calculer par votre fonction).

Q6 On pose $u_0 = 1$ et pour tout $n \in \mathbb{N}$,

$$u_{n+1} = \frac{1}{2} \left(u_n + \frac{n+1}{u_n} \right)$$

et $v_n = \sum_{k=0}^n \frac{1}{u_k^5}$

Écrire une fonction `f(n)` renvoyant la valeur de v_n .

On peut montrer que $(v_n)_{n \in \mathbb{N}}$ converge.

Vous pouvez calculer u_n pour de grandes valeurs de n .

Attention : on fera attention à ce que le calcul de `f` ne demande pas trop de (re)calculs inutiles. Pour fixer les idées, vous pouvez considérer que `f(10**6)` doit être calculé en (largement) moins d'une minute.

Q7 Écrire une fonction `somme1(n)` et une fonction `somme2(n)` prenant en argument un entier naturel `n` et renvoyant respectivement

$$\sum_{1 \leq i, j \leq n} \frac{1}{i + j^2}, \quad (1)$$

$$\sum_{1 \leq i < j \leq n} \frac{1}{i + j^2}. \quad (2)$$

Au besoin, on introduira des fonctions auxiliaires.