

DEVOIR SURVEILLÉ D'INFORMATIQUE N°4

L'utilisation de toute calculatrice et de tout matériel électronique est **interdite**. Les candidats ne peuvent utiliser qu'un document, le cours "Bilan de l'informatique : Ta fiche!". Dans l'écriture de vos programmes en Python, **respectez la ponctuation et l'indentation**.

Exercice 1 — Un petit pot pourri...

Aucune cohérence dans cet exercice, ce sont juste des extraits de vos TP!

1. Écrire une fonction prenant en entrée un entier naturel n et retournant $\sum_{k=0}^n k^7$ avec une boucle.
2. Écrire une fonction récursive prenant en entrée un entier naturel n et retournant $\sum_{k=0}^n k^7$.
3. Soit $(w_n)_{n \in \mathbb{N}}$ la suite définie par :

$$w_0 = 1 \text{ et } , \forall n \in \mathbb{N}, w_{n+1} = 12 \ln(w_n) + w_n^2.$$

Écrire une fonction qui prend en entrée un entier n qui renvoie la liste $[w_0, w_1, \dots, w_n]$.

4. Écrire une fonction prenant en entrée un entier naturel non nul n et traçant le graphique de la fonction $f : x \mapsto x \sin(x^2 + 1)$ sur l'intervalle $[-5, 5]$ avec n points bien répartis.
5. On considère l'équation différentielle (E) : $y'(t) = \cos(t) \times y^2(t)$ d'inconnue y fonction dérivable sur \mathbb{R}_+ . Soit φ la solution de (E) telle que $\varphi(0) = 2$ Écrire une fonction Euler qui prend en entrée un entier naturel non nul n et qui renvoie $g_n(5)$ avec g_n la fonction approchant φ en utilisant la méthode d'Euler sur $[0, 5]$ avec n points.
6. On pose $I = \int_0^1 \exp(-t^2) dt$. Écrire une fonction rectangles prenant en entrée un entier naturel n non nul et renvoyant une approximation de I en utilisant la méthode des rectangles à n pas.

Exercice 2 — La clé du voyage



Clovis part en voyage géologie avec sa nouvelle acquisition, une superbe clé USB Harry Potter. Elle est très belle mais ne dispose que de 8 Go. Pour s'occuper le soir, Clovis souhaite la remplir de films. Il a 9 fichiers vidéo à disposition sur son ordinateur. Chaque fichier a une taille et a une durée, ce tableau (cf. page suivante) présente les fichiers disponibles avec la taille en Go et les durées en minutes :

Nom	Vidéo 1	Vidéo 2	Vidéo 3	Vidéo 4	Vidéo 5	Vidéo 6	Vidéo 7	Vidéo 8	Vidéo 9
Temps	113	102	83	12	39	138	143	92	104
Taille	1,7	2,3	1,2	0,5	1,1	2	0,7	0,8	1

Clovis souhaite remplir sa clé de façon à ce que la durée totale de films emportés soit la plus grande.

1. La durée des vidéos est-elle proportionnelle à la taille? Pourquoi?
2. Pour répondre à la question, le plus naturel est d'utiliser un dictionnaire dont les clés sont les noms des fichiers et, pour chaque clé, la valeur correspondante est un dictionnaire dont les clés sont les chaînes "Durée" et "Taille" et dont les valeurs correspondantes sont la durée en minute et la taille en Go. Écrire le dictionnaire de notre problème.
3. Pour simplifier, on va utiliser une liste. Cette liste contiendra des listes de triplets, chaque triplet étant une liste de trois termes, le premier est le nom de la vidéo, le deuxième sa durée et le troisième sa taille. Dans tout le problème, on appellera `Lvideo` cette liste. C'est donc une liste de listes. Écrire un programme qui prend en entrée `Lvideo` et qui renvoie cette même liste complétée, chaque triplet devenant un quadruplet avec les mêmes trois premiers termes d'origine (nom, durée, taille) puis, en quatrième position, le rapport durée/taille. Le premier élément de cette liste sera donc la liste [Vidéo 1, 113, 1.7, 66.47058823529412].
4. On souhaite trier cette liste. Compléter le programme suivant de façon à ce que la liste de nombres `liste` prise en entrée soit triée dans l'ordre décroissant :

```
def Tri(liste) :
    for i in range (len (liste)-1 , 0, -1):
        for j in range (???????????) :
            if liste [j] < ??????????????:
                a=liste [j]
                liste [j] = liste [j+1]
                liste [j+1] = ???????????????
    return liste
```

5. Modifier le programme précédent de façon à ce qu'il puisse prendre en entrée `Lvideo` et renvoyer cette même liste trier par ordre décroissant du rapport durée/taille. Le premier élément de notre liste sera donc [Video 7', 143, 0.7, 204.2857142857143] et le dernier sera [Video 4', 12, 0.5, 24.0].
6. Écrire une fonction qui suit l'algorithme de remplissage de la clé suivant : on va donner la réponse dans la liste appelée `solution`, cette liste contiendra les noms des vidéos ajoutés. On ajoute les vidéos les unes après les autres en choisissant d'abord la vidéo ayant le meilleur rapport durée/taille puis on fait dans l'ordre des rapports durée/taille. On ajoute ces vidéos tant que la taille totale ne dépasse pas la taille maximale. On renvoie à la fin la liste `solution` ainsi que la durée totale de films mis sur la clé usb.
7. On voudrait tester toutes les combinaisons possibles. Pour cela, on va utiliser un 9-upplets de 0 et de 1, l'élément i de ce upplet vaut 1 si on a pris la vidéo i , 0 sinon. Ainsi, si ce upplet vaut [0,0,1,1,1,0,1,0,1], cela signifie que Clovis a mis uniquement les vidéo 3, 4, 5, 7 et 9 sur sa clé. Il y a, d'après les cours de M.Poullaouec, 2^9 combinaisons possibles. Donner un programme donnant la liste de ces 2^9 combinaisons possibles.
8. Donner un programme prenant en entrée un 9-upplets de 0 et de 1 révélant les vidéos prises (cf. question précédente) et la liste `Lvideo` et renvoyant une liste de deux termes : la durée totale et la taille totale de ces vidéos.
9. En déduire une fonction `meilleuresolution` qui prend en entrée `Lvideo`, qui teste toutes les combinaisons possibles et qui renvoie le meilleur choix possible ainsi que la durée totale.