TP d'informatique n°0
Premières révisions

PCSI 2023 – 2024

I À faire AVANT le premier TP

- Réviser la syntaxe python pour les opérations de base (affectation, addition, multiplication . . ., puissance, division entière, reste de la division entière (modulo)).
- Réviser la syntaxe python pour écrire une fonction.
- Réviser la syntaxe python pour importer un module (parfois appelé bibliothèque ou librairie dans les autres langages de programmation).
- Réviser la syntaxe python pour la structure conditionnelle (parfois appelé boucle si-alors, if-then).

Vous pouvez éventuellement regarder les vidéos :

`https://youtu.be/hpwva109JFg` `https://youtu.be/GzSlRkGhdR0`

Exercice n°1 Mise en pratique : Écrire une fonction `duree(h, m, s)` qui prend en argument un nombre d'heures, de minutes et de secondes et qui renvoie la durée correspondante en secondes. Par exemple `duree(2, 3, 1)` doit renvoyer 7381. Testez votre fonction à l'aide de python (sur votre ordinateur ou alors en utilisant un outil sur internet permettant d'exécuter du python).

II Pendant le TP

Exercice n°2 Écrire une fonction `hms(duree:int)->(int,int,int)` qui fait l'inverse, c'est-à-dire auquel on indique une durée en secondes et qui renvoie le nombre d'heures de minutes et de secondes (on pourra commencer par faire minutes-secondes uniquement, puis rajouter heure). Il est *conseillé* d'utiliser les opérateurs modulo et division entière. Par exemple `hms(7381)` doit renvoyer $(2, 3, 1)$, `hms(4810)` doit renvoyer $(1, 20, 10)$.
Si vous avez du mal à faire cet exercice, faites le à la main sur plusieurs exemples (que doit renvoyer `hms(8 000)` ? `hms(14 000)` ? `hms(70 000)` ? Identifiez les étapes que vous appliquez pour les faire faire à l'ordinateur ensuite.

Exercice n°3 Écrire une fonction `signe(x:float)->int` qui prend en argument un nombre réel et renvoie 1 si ce nombre est positif.

Exercice n°4 Améliorer la fonction précédente pour qu'elle renvoie -1 si le nombre n'est pas positif.

Exercice n°5 Améliorer la fonction de l'exercice précédent pour renvoyer 0 si le nombre est nul.

Exercice n°6 Écrire une fonction val_abs qui renvoie la valeur absolue d'un nombre.

Exercice n°7 Écrire une fonction qui renvoie le montant du salaire d'un employé en fonction du nombre d'heures travaillées, sachant que :

- Les 39 premières heures sont sans suppléments et valent 12 euros chacune.
- De la 40^e à la 44^e heure le salaire horaire est majoré de 50% (attention, seules les heures supplémentaires sont majorées, les 39 premières heures sont toujours payées au même tarif).
- De la 45^e à la 49^e heure le salaire horaire est majoré de 75%.
- Au delà de la 50^e heure le salaire horaire est majoré de 100%.

Exemples :

h	35	39	41	44	45	47	49	50	55
salaire(h)	420	468	504	558	579	621	663	687	807

Exercice n°8 Définir une fonction `hypotenuse(x:float,y:float)->float` qui calcule la longueur de l'hypoténuse d'un triangle rectangle, où x et y sont les longueurs des deux autres cotés.

Exercice n°9 De façon générale, les années bissextiles sont celles qui sont divisibles par 4. Par exemple 2016 est bissextile, 2015 ne l'est pas.
Écrire une fonction `bissextile(n:int)->bool` qui pour une année n renvoie le booléen `True` si l'année est bissextile et `False` sinon.

Exercice n°10 La règle est en fait un peu plus complexe : les années multiples de 100 ne sont pas bissextile : modifier votre fonction pour prendre en compte cela. Par exemple 2100 n'est pas bissextile.

Exercice n°11 La règle est en fait encore plus complexe : les années multiples de 400 sont malgré tout bissextile : modifier votre fonction pour prendre en compte cela. Par exemple 2000 est bissextile.
Tester votre fonction complète avec les années 2000, 2020, 2021, et 2100.

Exercice n°12 Réaliser une fonction `trinome(a,b,c)` qui prend en argument un trinôme du second degré via ses trois coefficients ($P = aX^2 + bX + c$). La fonction renvoie la ou les racines réelles du polynôme (c'est-à-dire les solutions de l'équation $P = 0$) ou, le cas échéant, affiche un message indiquant que ce polynôme n'a pas de racine réelle et ne renvoie rien (None).

Exercice n°13 À partir des différentes informations du numéro de sécurité sociale, on veut recomposer le numéro total et vérifier la validité de la saisie. Les différents chiffres ont la signification suivante :

position	signification
1	sexe (1 pour les hommes, 2 pour les femmes)
2 et 3	deux derniers chiffres de l'année de naissance
4 et 5	mois de naissance
6 et 7	département de naissance
8, 9, 10	numéro de la commune à l'intérieur du département
11, 12, 13	numéro de naissance à l'intérieur de la commune
14 et 15	clé de contrôle

La clé de contrôle est calculée en faisant l'opération suivante : on appelle n le nombre formé des 13 premiers chiffres, la clé est $97 - n\%97$.
Par exemple : une femme (2) née en 1955 (55) au mois d'août (8) dans le département 14 (Calvados) dans la commune numéro 168 (Colombières) en 25e avec pour clé de contrôle 38. Le numéro INSEE de cette personne est :

$$255081416802538$$

Écrire une fonction qui prend en argument 7 nombres entiers : `sexe`, `annee`, `mois`, `departement`, `commune`, `numero`, `cle` et effectue les opérations suivantes (dans l'ordre) :

— calcule le nombre n défini ci-dessus et en déduit la clé à partir des 13 premiers chiffres
— affiche (print) la clé calculé et la clé rentrée par l'utilisateur
— renvoie (return) le numéro de sécurité social (donc en rajoutant la clé à la fin, on pourra choisir indifféremment la clé rentrée ou calculée).

Exercice n°14 À partir d'un numéro de sécurité sociale, redonnez les différentes informations, renvoyez `False` si la clé calculée et la clé rentrée par l'utilisateur sont incompatibles et `True` sinon.

Exercice n°15 Nous utilisons la plupart du temps la base 10 dans la vie de tous les jours. Ainsi lorsque nous écrivons le nombre 254, cela signifie : $2 \times 10^2 + 5 \times 10^1 + 4 \times 10^0$. Dans cet exercice, on travaillera uniquement avec des nombres à trois chiffres. Faites une fonction `chiffre_vers_nombre` qui prend en argument trois chiffres a, b, c et qui renvoie le nombre qui s'écrit abc.

Exercice n°16 Faites une fonction faisant l'inverse.

Exercice n°17 Refaites les deux exercices précédent, mais en base 3 (en se limitant à des nombres strictement inférieurs à 3^3 pour n'avoir que 3 chiffres).

III Pour la prochaine fois

Sans relire le TP, essayez de répondre aux questions suivantes (ne relisez le TP qu'après avoir essayé de répondre aux questions)

- Quelles opérations peut-on utiliser en Python ?
- Qu'est-ce que l'opérateur modulo ? Comment représente-t-on l'opérateur division entière ?
- Comment indiquer un commentaire ?
- Pourquoi mettre des commentaires ?
- Comment faire une fonction ? Comment faire une fonction qui prend plusieurs arguments ?
- Pourquoi faire une fonction ?

Exercice n°18 Installer python chez vous si vous le pouvez (winpython conseillé sur windows, anaconda sur mac ou linux, toute distribution acceptable tant que vous avez les bibliothèques scientifiques numpy et matplotlib).

Exercice n°19 Programmer une fonction `maxi2(a,b)` qui renvoie le nombre le plus grand entre a et b.

Exercice n°20 Programmer une fonction `maxi3(a,b,c)` qui renvoie le nombre le plus grand entre a, b et c.

Exercice n°21 Refaire l'exercice précédent en utilisant la fonction maxi2 pour alléger l'écriture de votre fonction. (Si vous l'aviez fait spontanément, c'est très bien).

Exercice n°22 Sur une feuille de papier d'abord, puis sur un ordinateur pour tester, écrivez une fonction prenant en argument un nombre (entier) et qui renvoie le dernier chiffre de ce nombre (dans son écriture en base 10).

Exercice n°23 Bob et Alice se disputent : ils ne sont pas d'accord sur la manière de faire référence aux différents éléments d'un tableau. Bob choisi de numéroter les cases en allant de gauche à droite, puis à chaque fois qu'il arrive au bout d'une ligne, il revient au début de la ligne suivante en continuant de compter (en gris sur la figure ci-dessous). À l'inverse, Alice trouve plus simple de faire référence à un élément en donnant son numéro de ligne et de colonne sous la forme d'un couple (ligne, colonne) (en noir sur la figure ci-dessous). Nos deux amis choisissent de numéroter à partir de 0.

0 (0,0)	1 (0,1)	2 (0,2)	3 (0,3)	4 (0,4)
5 (1,0)	6 (1,1)	7 (1,2)	8 (1,3)	9 (1,4)
10 (2,0)	11 (2,1)	12 (2,2)	13 (2,3)	14 (2,4)
15 (3,0)	16 (3,1)	17 (3,2)	18 (3,3)	19 (3,4)
20 (4,0)	21 (4,1)	22 (4,2)	23 (4,3)	24 (4,4)
25 (5,0)	26 (5,1)	27 (5,2)	28 (5,3)	29 (5,4)

Ainsi le premier élément du tableau est l'élément n°0 pour Bob et le (0,0) pour Alice. Le dernier est le n°29 pour Bob et le (5,4) pour Alice. Afin de réconcilier Bob et Alice, écrire une fonction `Alice_to_Bob` permettant de réaliser la conversion de la numérotation d'Alice vers celle de Bob et une fonction `Bob_to_Alice` permettant de faire l'inverse.
Faites le d'abord sur papier, puis à l'ordinateur pour tester vos fonctions. Pour qu'une fonction renvoie plusieurs nombres (par exemple les nombres contenus dans les variables ligne et colonne) il suffit de les séparer par une virgule au niveau du return : **return** `ligne,colonne`