

EXERCICES

1. Echanger deux variables

Étant donné deux variables x et y , on veut permuter leurs valeurs. Par exemple si au départ $x = 81$ et $y = 31$ alors, après échange, on doit avoir : $x = 31$ et $y = 81$. On pourra compléter le code ci-dessous :

```
1 x=81
2 y=31
3 print(x,y)
4
5 # VOTRE
6 # CODE
7 # ICI
8
9 print(x,y)
```

Bien sûr votre code doit fonctionner correctement si on donne d'autres valeurs à x et y aux lignes 1 et 2.

On fera comme lorsqu'on veut échanger deux meubles A et B de place : on utilise un emplacement de stockage temporaire.

2. Calculs

Faire effectuer par Python les calculs suivants :

$$a = 10^{2 \times 3} \quad b = \frac{666}{74 \times 9} \quad c = 2 \times 3,14159 + 2,71828 + 1,4142 \times 5$$

3. Priorités des opérateurs

L'expression $2 + 3 * 5$ pourrait a priori s'interpréter $(2 + 3) * 5$ ou $2 + (3 * 5)$. Demandons à Python la valeur des trois expressions :

```
1 x=2 + 3 * 5
2 y=(2 + 3) * 5
3 z=2 + (3 * 5)
4 print (x==y, x==z)
5 False True
```

Donc la bonne interprétation avec les parenthèses est $2 + (3 * 5)$.

Faire de même pour les cinq expressions suivantes autrement dit placer toutes les parenthèses utiles dans l'expression pour obtenir une expression non ambiguë et de même valeur :

```
1 - 1 ** 2
2 10 ** 2 ** 3
3 7 * 29 / 7
4 2 / 1 * 2
5 60 / 6 / 2 / 5
```

4. La finance aime Python

Traduire en utilisant une variable `milliard` et une variable `million` l'opération suivante :

200 milliards de millions augmenté de 2020 millions et privé de 925 milliards.

Afficher le résultat (on trouvera 199999077020000000).

5. *Vulnerant omnes, ultimat ne cat*

- ① Combien y-a-t-il de secondes dans un siècle ? (utiliser des variables `une_heure`, `j` et `an` et supposer qu'une année est constituée de 365 jours 1/4)
- ② À partir de quel âge (entier) un individu a vécu au moins 1 milliard de secondes (réponse : 32 ans) ?

6. Diviseur, multiple

Utiliser massivement des variables et les possibilités d'affichage multiple de chaînes et de nombres (cf. cours) pour avoir un code le plus homogène possible.

- ① Vérifier que 101 est un diviseur de 2020.
- ② Montrer que 192642212037519289 et 138633362400520449 ont même reste dans la division entière par 2020.

7. Multiples de 421

Ecrire la liste des 5 premiers multiples de 421. On donnera deux codes :

- un code affichant les nombres les uns en-dessous des autres,
- un code affichant les nombres les uns à côtés des autres.

8. Nombre de tours

Utiliser des variables pour résoudre le problème suivant :

un coureur fait un tour de circuit en c minutes. Quel est le nombre x de tours complets qu'il effectuera en n minutes ? Appliquer avec $c = 7$, $n = 240$.

9. Reproduire un affichage

Soit une variable `x`, représentant un entier. On cherche à fournir un affichage du calcul de $10x + 1$. Si, par exemple, $x = 42$, on veut que l'affichage ait EXACTEMENT la forme suivante

```
1 x = 42 => 10 * x + 1 = 421
```

ou encore (si $x = 3$) :

```
1 x = 3 => 10 * x + 1 = 31
```

Écrire un code qui produise ce type d'affichage.

10. Conditions sur un entier

Dans cette question, `x` est supposé être un entier. Créer une variable booléenne `cond` (prenant comme valeur `True` ou `False`) qui teste si `x` est ou bien multiple de 10 ou bien à la fois impair, non multiple de 3 et compris, au sens large, entre 42 et 421. Ainsi `x = 2020` passe le test, `x = 301` aussi, `x = 420` également mais pas `x = 81`.

11. Vacances

On donne deux entiers `j` et `m` représentant une date valide où `m` représente le numéro du mois (entre 1 et 12) et `j` le jour du mois (entre 1 et 31). Ainsi,

- la donnée `j = 25` et `m = 4` représente le 25 avril
- le 31 septembre est représenté par `j = 31` et `m = 9`.

En France, les vacances d'été 2017 auront lieu du dimanche 8 juillet au dimanche 3 septembre 2017. On donne le jour j et le mois m d'une date et on vous demande de construire une variable booléenne `enVacances` valant `True` si la date correspond à un jour de vacances d'été 2017 et `False` sinon. Exemples :

```
1 j = 1 et m = 5 -> False
2 j = 6 et m = 7 -> False
3 j = 7 et m = 7 -> False
4 j = 14 et m = 7 -> True
5 j = 2 et m = 8 -> True
6 j = 31 et m = 8 -> True
7 j = 3 et m = 9 -> True
8 j = 4 et m = 9 -> False
9 j = 7 et m = 9 -> False
10 j = 8 et m = 12 -> False
```

12. Test de la non-égalité de cinq entiers

On donne cinq entiers a , b , c , d et e . Soit la condition C suivante : *les 5 entiers ne sont pas tous égaux*. Par exemple

- si $a=5$, $b=5$, $c=5$, $d=4$ et $e=4$ alors C est `True`
- si $a=5$, $b=5$, $c=5$, $d=5$ et $e=5$ alors C est `False`

Ecrire une variable booléenne qui vaut `True` si la condition C est vraie et `False` sinon.

13. Au moins un pair et au moins un supérieur à 100

On donne trois entiers positifs a , b et c . Construire un booléen `ok` qui vaut `True` si parmi les trois nombres, il y a au moins un nombre pair et aussi s'il y a au moins un nombre supérieur ou égal à 100. Sinon, `ok` vaut `False`

Exemples :

```
1 7, 42, 142 -> True
2 42, 60, 71 -> False
3 111, 125, 2015 -> False
4 80, 111, 31 -> True
```

14. Entiers qui se suivent

On vous donne deux variables x et y représentant des entiers. Définir une variable booléenne `ok` valant `True` si

- les entiers sont entre 1 et 4 ET
- ou bien sont consécutifs
- ou bien l'un vaut 1 et l'autre vaut 4.

Sinon, `ok` vaudra `False`.

Exemples

```
1 x=2, y=3 -> ok = True
2 x=3, y=2 -> ok = True
3 x=4, y=4 -> ok = False
4 x=7, y=8 -> ok = False
5 x=4, y=1 -> ok = True
6 x=3, y=1 -> ok = False
```

15. Formule

On connaît la formule mathématique suivante $1 + x + x^2 + x^3 = \frac{x^4 - 1}{x - 1}$.

- ① Introduire une variable `x`, par exemple `x = 42` et traduire les deux expressions (celle de gauche et celle de droite) de la formule ci-dessus par du code Python avec deux variables `gauche` et `droite`. ATTENTION aux parenthèses!
Afficher un booléen qui indique si les valeurs de `gauche` et `droite` sont égales.
- ② Vérifiez l'exactitude de cette formule pour `x = 208066`. Si la formule n'est pas exacte, modifiez votre code en observant que le quotient exact au membre de droite doit être un entier puisque le membre de gauche est un entier.

16. Entier ayant n chiffres

On vous donne un entier $x > 0$ et un autre entier $n > 0$. Créer une variable booléenne `avoir_n_chiffres` (prenant comme valeur `True` ou `False`) qui teste si l'entier x admet exactement n chiffres dans son écriture en base 10. Par exemple, si $x = 421$ et $n = 2$, la condition vaut `False` et si $n = 3$, la condition vaut `True`. Pour trouver la condition, on pourra observer, en utilisant des puissances de 10, quel est le plus petit et le plus grand entier ayant n chiffres.

17. Nombre intermédiaire

On donne trois entiers a , b et c , par exemple $a = 42$, $b = 100$ et $c = 10$. On demande de déterminer et d'afficher le nombre qui est encadré par les deux autres. Dans l'exemple précédent, on a $c \leq a \leq b$ donc le nombre demandé est $a = 42$.

18. Transport en bus

Un bus peut contenir p passagers. Combien faut-il de bus pour transporter n passagers? Coder en Python la réponse à cette question et appliquer à différents exemples (attention, la réponse dépend d'une condition laquelle, bien sûr, doit être exprimée à l'aide d'une instruction `if`).

19. Excédent payé

Soit à régler un montant de `m` euros avec des billets de 20 euros. Par exemple,

- si `m = 85` alors il faut 5 billets de 20 euros
- si `m = 120` alors il faut 6 billets de 20 euros

On demande de calculer le nombre `exc` représentant l'excédent payé en fonction de `m`. Par exemple,

- si `m = 85` alors `exc = 15`.
- si `m = 120` alors `exc = 0`.

20. Jeu à deux nombres

Dans cet exercice, on appellera *combinaison* la donnée de deux entiers a et b . Ainsi, on peut parler de :

- la combinaison 42, 17;
- la combinaison 81, 81.

À partir d'une combinaison, on a les règles suivantes :

- Si $a=b$, la combinaison rapporte 10 points.
- Si a et b sont consécutifs (comme $a=5$ et $b=4$), alors la combinaison rapporte 3 points.
- sinon la combinaison ne rapporte rien.

Écrire un code Python qui, étant donné une combinaison de deux entiers a et b , affiche le nombre de points que rapporte la combinaison.

21. Heures d'ouverture

Dans l'exercice, on supposera que

- les jours de la semaine sont codés à l'aide d'un entier j avec $j = 1$ pour lundi, $j = 2$ pour mardi, et ainsi de suite jusqu'à dimanche ($j = 7$).
- une heure est codée par un entier entre 0 (inclus) et 24 (exclu).

Un magasin est ouvert du lundi au vendredi de 8h à 18h et le samedi de 9h à 12h et il est fermé le reste du temps.

On demande d'écrire une fonction f prenant en paramètres un jour j et une heure h et qui renvoie **True** si le magasin est ouvert le jour j à l'heure h et **False** sinon.

Voici quelques exemples de comportements de la fonction f :

- $f(2, 17)$ vaut **True**
- $f(6, 10)$ vaut **True**
- $f(5, 18)$ vaut **True**
- $f(2, 19)$ vaut **False**
- $f(6, 17)$ vaut **False**

22. Arrondir l'heure

- ① On donne un entier $n \geq 0$. Construire une variable N valant le multiple de 5 le plus proche de n . Vous raisonnerez en fonction la valeur du reste de la division de n par 5.

Par exemple :

- si $n = 42$ alors $N = 40$
- si $n = 15$ alors $N = 15$
- si $n = 64$ alors $N = 65$
- si $n = 90$ alors $N = 90$
- si $n = 0$ alors $N = 0$

- ② Cette question fait appel à la question précédente. Si vous n'avez pas réussi à calculer m , vous pourrez utiliser le code ci-dessous pour le multiple de 5 le plus proche de l'entier n :

```
1 m = round(n/5)*5
```

Vous allez devoir écrire un code qui arrondit une heure donnée aux 5 minutes les plus proches. Une heure de la journée sera codée par deux nombres entiers h et m , où h est le nombre d'heures ($0 \leq h < 24$) et m le nombre de minutes ($0 \leq m < 60$). Par exemple, 14 h 05 est codée par $h = 14$ et $m = 5$ ou encore 4 h sera codée par $h = 4$ et $m = 0$.

On se donne une heure de la journée, représentée par deux nombres entiers h et m . Écrire un code qui calcule l'heure arrondie à 5 minutes près. Plus précisément, votre code devra construire deux variables H et M correspondant à l'heure arrondie. Voici quelques exemples de comportements attendus :

```

1 14h 53m -> [14, 55]
2 18h 31m -> [18, 30]
3 02h 10m -> [2, 10]
4 01h 02m -> [1, 0]
5 09h 58m -> [10, 0]
6 23h 58m -> [0, 0]
7 23h 57m -> [23, 55]

```

On fera attention de ne pas écrire des nombres de minutes ou d'heures avec un zéro initial (comme 02), Python 3 considérant cette syntaxe comme une erreur.

23. Équation du second degré

L'exercice consiste à écrire un code permettant de résoudre dans l'ensemble des réels une équation du second degré $ax^2 + bx + c = 0$. On supposera que a, b, c sont des entiers, que a est non nul et que l'inconnue x est un nombre réel.

Rappels : résolution de l'équation du second degré

On rappelle la résolution de l'équation. On calcule $\Delta = b^2 - 4ac$. Il y a alors trois cas :

— 1^{er} cas : $\Delta > 0$. L'équation admet deux solutions réelles, données par les formules

$$x_1 = \frac{-b - \sqrt{\Delta}}{2a} \quad \text{et} \quad x_2 = \frac{-b + \sqrt{\Delta}}{2a}.$$

— 2^e cas : $\Delta = 0$. L'équation admet une seule solution réelle, $x = \frac{-b}{2a}$, donnée par n'importe laquelle des deux formules du 1^{er} cas.

— 3^e cas : $\Delta < 0$. L'équation n'admet aucune solution.

- ① Écrire un code Python qui détermine le nombre de solutions d'une équation du second degré.
- ② Affiner le code précédent pour résoudre complètement dans les réels l'équation $ax^2 + bx + c = 0$ et afficher les solutions éventuelles sous forme de nombres flottants (et pas de fractions). Tester les trois équations suivantes :

Equation	Solutions
$6x^2 - 5x + 1 = 0$	1/2 et 1/3
$4x^2 - 12x + 9 = 0$	3/2
$6x^2 + 7x + 7 = 0$	Aucune