Exercice 1 — Lire une boucle for ●○○

Quelle valeur ce programme affiche-t-il ? Justifie.

```
a = 2
for i in range(10):
    a = a * 2
print(a)
```

Exercice 2 — Calculer un terme d'une suite ●○○

La suite (u_n) est définie par $u_0 = 10$ et $u_{n+1} = 0{,}5\,u_n + 3$. Complète la fonction pour qu'elle renvoie u_n, puis donne la valeur de `terme(2)`.

```
def terme(n):
    u = ........
    for i in range(........):
        u = ........
    return u
```

Exercice 3 La liste des premiers termes ●○○

Que renvoie `liste(4)` ?

```
def liste(n):
    u = 1
    L = [u]
    for i in range(n):
        u = 3*u - 1
        L.append(u)
    return L
```

Exercice 4 Une somme avec une boucle ●○○

Écris une fonction `somme(n)` qui renvoie $1 + 2 + \cdots + n$. Que renvoie `somme(100)` ?
Vérifie avec une formule du cours.

Exercice 5 — Un algorithme de seuil ●●○

On considère la suite $u_n = 3 \times 1{,}1^n$ et le programme ci-dessous.

```
n = 0
u = 3
while u <= 10:
    u = u * 1.1
    n = n + 1
print(n)
```

a. Que représente la valeur affichée ?

b. Sachant que $1{,}1^{12} \approx 3{,}14$ et $1{,}1^{13} \approx 3{,}45$, quelle valeur est affichée ?

Exercice 6 — Listes en compréhension ●○○

Donne le contenu de chaque liste.

```
A = [n**2 for n in range(6)]
B = [2*k + 1 for k in range(5)]
C = [x for x in range(20) if x % 3 == 0]
```

Exercice 7 — Trouver l'erreur ●●○

Un élève veut calculer u_n pour la suite géométrique de premier terme $u_0 = 5$ et de raison 2. Il écrit :

```
def u(n):
    v = 5
    for i in range(n+1):
        v = 2*v
    return v
```

Il obtient `u(3) = 80`. Explique l'erreur et corrige-la.

Exercice 8 — Simuler des lancers de dé ●●○

```
from random import randint
def frequence(N):
    c = 0
    for i in range(N):
        if randint(1, 6) == 6:
            c = c + 1
    return c / N
```

a. Que calcule `frequence(N)` ?

b. De quelle valeur `frequence(100000)` devrait-elle être proche ?

Exercice 9 Encadrer une solution par balayage ●●○

On cherche une solution de $x^3 + x - 1 = 0$ entre 0 et 1.

```
def f(x):
    return x**3 + x - 1
x = 0
while f(x) < 0:
    x = x + 0.1
print(x)
```

a. Calcule $f(0{,}6)$ et $f(0{,}7)$.

b. Quelle valeur le programme affiche-t-il (aux arrondis près) ? Qu'en déduit-on sur la solution ?

Exercice 10 Deux variables à la fois ●●○

Quelle valeur ce programme affiche-t-il ? Présente les valeurs de a et b après chaque passage dans un tableau.

```
a, b = 1, 1
for i in range(5):
    a, b = b, a + b
print(a)
```

Exercice 11 — Moyenne et maximum d'une liste ●●○

Sans utiliser les fonctions `sum` et `max` de Python, écris :

a. une fonction `moyenne(L)` qui renvoie la moyenne des nombres de la liste `L` ;

b. une fonction `maximum(L)` qui renvoie le plus grand nombre de `L`.

Exercice 12 — Compter les diviseurs ●●○

```
def nb_div(n):
    c = 0
    for d in range(1, n+1):
        if n % d == 0:
            c = c + 1
    return c
```

a. Que renvoie `nb_div(12)` ? `nb_div(13)` ?

b. Comment utiliser cette fonction pour savoir si un nombre est premier ?

Exercice 13 La méthode de Héron ●●●

La suite définie par $u_0 = 1$ et $u_{n+1} = \frac{1}{2}\left(u_n + \frac{2}{u_n}\right)$ donne des valeurs approchées de $\sqrt{2}$.

a. Calcule u_1 et u_2 sous forme de fractions.

b. Écris un programme qui calcule les termes tant que $|u_n^2 - 2| \geqslant 10^{-6}$ et affiche le dernier terme et son rang.

Exercice 14 TYPE CONTRÔLE **L'élevage de bactéries** ●●●

Dans un laboratoire, la masse de bactéries augmente de $20\,\%$ chaque heure, puis on en prélève 1 kg. Au départ, il y a 10 kg de bactéries. On note m_n la masse, en kg, au bout de n heures ; $m_0 = 10$.

1. Calcule m_1 et m_2 et justifie que $m_{n+1} = 1{,}2\,m_n - 1$.
2. Écris une fonction Python `masse(n)` qui renvoie m_n.
3. On pose $v_n = m_n - 5$. Montre que (v_n) est géométrique et déduis-en m_n en fonction de n.
4. Écris un programme qui affiche le nombre d'heures au bout duquel la masse dépasse 50 kg.
5. Sachant que $1{,}2^{12} \approx 8{,}92$ et $1{,}2^{13} \approx 10{,}70$, quelle valeur affiche ce programme ?

Exercice 15 TYPE CONTRÔLE Simulation et probabilité ●●●

On lance trois fois une pièce équilibrée et on s'intéresse à l'évènement A : « obtenir au moins un pile ».

```
from random import randint
def experience():
    for i in range(3):
        if randint(0, 1) == 1:
            return 1
    return 0

def freq(N):
    s = 0
    for k in range(N):
        s = s + experience()
    return s / N
```

1. Dans la fonction `experience`, que représente `randint(0, 1) == 1` ? Que renvoie la fonction ?
2. Que calcule `freq(N)` ?
3. Calcule la probabilité de A à l'aide de l'évènement contraire.
4. Un élève obtient `freq(10000) = 0.8741`. Ce résultat est-il cohérent ?
5. Modifie la fonction `experience` pour simuler 4 lancers et compte le nombre de piles obtenus (la fonction renverra ce nombre).

CAMPUS XYZ

Corrigés

Exercices sur l'algorithmique en première spé maths

Exercices sur l'algorithmique en première spé maths : corrigés

CAMPUS XYZ

Exercice 1 — Lire une boucle for ●○○

`range(10)` donne 10 passages ; à chaque passage, a est multiplié par 2. On part de 2 : $a = 2 \times 2^{10} = 2^{11}$.

Le programme affiche $2\,048$.

Exercice 2 — Calculer un terme d'une suite ●○○

```
def terme(n):
    u = 10
    for i in range(n):
        u = 0.5*u + 3
    return u
```

$u_1 = 0{,}5 \times 10 + 3 = 8$ et $u_2 = 0{,}5 \times 8 + 3 = 7$: **`terme(2)` renvoie 7.**

Exercice 3 — La liste des premiers termes ●○○

La liste commence par $u_0 = 1$, puis on ajoute 4 termes : $3 \times 1 - 1 = 2$, $3 \times 2 - 1 = 5$, 14, 41.

`liste(4)` renvoie `[1, 2, 5, 14, 41]`.

Exercice 4 — Une somme avec une boucle ●○○

```
def somme(n):
    s = 0
    for k in range(1, n+1):
        s = s + k
    return s
```

`somme(100)` renvoie $5\,050$, et la formule $\dfrac{n(n+1)}{2} = \dfrac{100 \times 101}{2} = 5\,050$ donne le même résultat.

> `range(1, n+1)` va de 1 à n : la borne de droite est exclue.

Exercice 5 — Un algorithme de seuil ●●○

a. **Le plus petit entier n tel que $u_n > 10$.**

b. $u_{12} \approx 3 \times 3{,}14 = 9{,}42 \leqslant 10$ et $u_{13} \approx 3 \times 3{,}45 = 10{,}35 > 10$: **le programme affiche 13.**

Exercice 6 — Listes en compréhension ●○○

- `A = [0, 1, 4, 9, 16, 25]` (les carrés de 0 à 5).
- `B = [1, 3, 5, 7, 9]` (les 5 premiers nombres impairs).
- `C = [0, 3, 6, 9, 12, 15, 18]` (les multiples de 3 inférieurs à 20 : `%` donne le reste).

Exercice 7 — Trouver l'erreur ●●○

$u_3 = 5 \times 2^3 = 40$. Avec `range(n+1)`, la boucle tourne $n + 1$ fois : on multiplie une fois de trop par 2.

Il faut écrire `range(n)` : n multiplications pour passer de u_0 à u_n.

Exercice 8 — Simuler des lancers de dé ●●○

a. **La fréquence d'apparition du 6 sur N lancers simulés d'un dé.**

b. Pour un grand nombre de lancers, la fréquence se rapproche de la probabilité : $\dfrac{1}{6} \approx 0{,}167$.

Exercice 9 — Encadrer une solution par balayage ●●○

a. $f(0{,}6) = 0{,}216 + 0{,}6 - 1 = -0{,}184$ et $f(0{,}7) = 0{,}343 + 0{,}7 - 1 = 0{,}043$.

b. La boucle s'arrête dès que $f(x) \geqslant 0$: **le programme affiche $0{,}7$.** f change de signe entre $0{,}6$ et $0{,}7$: la solution est comprise entre ces deux nombres.

Exercice 10 — Deux variables à la fois ●●○

À chaque passage, a prend l'ancienne valeur de b et b prend la somme des deux anciennes valeurs.

passage	départ	1	2	3	4	5
a	1	1	2	3	5	8
b	1	2	3	5	8	13

Le programme affiche 8 (ce sont les termes de la suite de Fibonacci).

Exercice 11 Moyenne et maximum d'une liste ●●○

```
def moyenne(L):
    s = 0
    for x in L:
        s = s + x
    return s / len(L)

def maximum(L):
    m = L[0]
    for x in L:
        if x > m:
            m = x
    return m
```

Par exemple, `moyenne([12, 8, 15, 9])` renvoie 11 et `maximum([12, 8, 15, 9])` renvoie 15.

Exercice 12 Compter les diviseurs ●●○

a. Diviseurs de 12 : $1, 2, 3, 4, 6, 12$, **`nb_div(12)` renvoie** 6. 13 n'a que 1 et 13 : **`nb_div(13)` renvoie** 2.

b. **n est premier si et seulement si `nb_div(n) == 2`.**

Exercice 13 La méthode de Héron ●●●

a. $u_1 = \dfrac{1}{2}(1+2) = \dfrac{3}{2}$; $u_2 = \dfrac{1}{2}\left(\dfrac{3}{2} + \dfrac{4}{3}\right) = \dfrac{17}{12} \approx 1{,}4167$.

b.

```
u = 1
n = 0
while abs(u**2 - 2) >= 10**(-6):
    u = (u + 2/u) / 2
    n = n + 1
print(n, u)
```

Il s'arrête à $n = 4$ avec $u_4 \approx 1{,}4142136$: la méthode converge très vite.

Exercice 14 TYPE CONTRÔLE L'élevage de bactéries ●●●

1. $m_1 = 1{,}2 \times 10 - 1 = 11$ et $m_2 = 1{,}2 \times 11 - 1 = 12{,}2$. Augmenter de $20\,\%$, c'est multiplier par $1{,}2$, puis on retire 1.

2.

```
def masse(n):
    m = 10
    for i in range(n):
        m = 1.2*m - 1
    return m
```

3. $v_{n+1} = 1{,}2\,m_n - 6 = 1{,}2\,(m_n - 5) = 1{,}2\,v_n$: **géométrique de raison** $1{,}2$, $v_0 = 5$. $m_n = 5 + 5 \times 1{,}2^n$.

4.

```
m = 10
n = 0
while m <= 50:
    m = 1.2*m - 1
    n = n + 1
print(n)
```

5. $m_n > 50 \iff 1{,}2^n > 9$. $1{,}2^{12} \approx 8{,}92 \leqslant 9$ et $1{,}2^{13} \approx 10{,}70 > 9$: **il affiche** 13.

Exercice 15 TYPE CONTRÔLE Simulation et probabilité ●●●

1. Un lancer qui donne pile (une chance sur deux). **La fonction renvoie 1 dès qu'un pile apparaît, et 0 si les trois lancers donnent face**.

2. **La fréquence de l'évènement A sur N expériences simulées**.

3. $\overline{A}$: « trois faces », de probabilité $\left(\frac{1}{2}\right)^3 = \frac{1}{8}$. $P(A) = \frac{7}{8} = 0{,}875$.

4. **Oui** : $0{,}8741$ est très proche de $0{,}875$, comme prévu pour un grand nombre de simulations.

5.

```
def experience():
    c = 0
    for i in range(4):
        if randint(0, 1) == 1:
            c = c + 1
    return c
```

LES FICHES MAGIQUES 1RE SPÉ MATHS

Révise toute la première spé maths avec les fiches magiques



- **40 fiches claires et illustrées** : tout le programme de spécialité maths de première, une notion par page.
- **Des vidéos méthode** accessibles par QR code sur les fiches.
- **Des exercices corrigés** après chaque fiche pour vérifier que tout est acquis.

campusxyz.fr/produit/les-fiches-magiques-1re