Corrigé – Python : boucles for et while

Mathématiques – Seconde

Algorithmique et programmation · exercices ★ à ★★★, pièges et QCM

Pour chaque erreur : relis la ligne du corrigé, trouve l'étape où tu t'es trompé, puis refais le calcul sans regarder.

Exercice 1 – Les valeurs de k ★

Pour `for k in ...`, donne les valeurs prises par `k` et le nombre de tours.

a) `range(4)` part de 0 et s'arrête avant 4.
0, 1, 2, 3 : **4** tours.

b) De 1 à $5-1$: $5-1=4$ tours.
1, 2, 3, 4 : **4** tours.

c) De 3 à $8-1$: $8-3=5$ tours.
3, 4, 5, 6, 7 : **5** tours.

d) `range(10)` part de 0 et s'arrête avant 10.
0, 1, 2, 3, 4, 5, 6, 7, 8, 9 : **10** tours.

e) De 5 à $6-1$: $6-5=1$ tour.
5 : **1** tour.

f) $5-5=0$: la boucle ne fait aucun tour.
aucune valeur : **0** tour *(et non* 1*)*.

g) De -2 à $3-1$: $3-(-2)=5$ tours.
-2, -1, 0, 1, 2 : **5** tours.

h) On part de 1, on avance de 3 à chaque tour, et on s'arrête avant 10.
1, 4, 7 : **3** tours.

i) On part de 0, on avance de 5 à chaque tour, et on s'arrête avant 20.
0, 5, 10, 15 : **4** tours.

j) On part de 9, on recule de 1 à chaque tour, et on s'arrête avant 4.
9, 8, 7, 6, 5 : **5** tours.

Exercice 2 – Qu'affiche le programme ? ★

Écris ce que chaque programme affiche, dans l'ordre.

a) Suivi, tour par tour :

Tour	**Avant**	**1**	**2**	**3**	**4**
k		`0`	1	2	3
affiche		`0`	1	4	9

Le programme affiche `0` puis 1 puis 4 puis 9.

b) Suivi, tour par tour :

Tour	Avant	1	2	3
k		1	2	3
affiche		3	5	7

Le programme affiche 3 puis 5 puis 7.

c) Suivi, tour par tour :

Tour	Avant	1	2	3
k		0	1	2
x	1	2	4	8

Le programme affiche 8.
`print` est hors de la boucle : il affiche une seule fois, après le dernier tour.

d) Suivi, tour par tour :

Tour	Avant	1	2	3
k		0	1	2
x	1	2	4	8
affiche		2	4	8

Le programme affiche 2 puis 4 puis 8.

e) Suivi, tour par tour :

Tour	Avant	1	2	3
i		0	1	2
affiche		ok	ok	ok

Le programme affiche ok puis ok puis ok.

f) Suivi, tour par tour :

Tour	Avant	1	2	3
k		0	1	2
n	10	7	4	1

Le programme affiche 1.
`print` est hors de la boucle : il affiche une seule fois, après le dernier tour.

g) Suivi, tour par tour :

Tour	Avant	1	2	3	4	5
k		2	3	4	5	6
affiche		2		4		6

Le programme affiche 2 puis 4 puis 6.

h) Suivi, tour par tour :

Tour	Avant	1	2	3	4
k		1	2	3	4
s	0	1	3	6	10
affiche		1	3	6	10

Le programme affiche 1 puis 3 puis 6 puis 10.

Exercice 3 – Accumulateurs ★

Donne la valeur de la variable de la première ligne à la fin du programme.

a) S prend les valeurs $0 \to 1 \to 3 \to 6 \to 10 \to 15$.
À la fin, $S = 15$: $1 + 2 + 3 + 4 + 5$.

b) P prend les valeurs $1 \to 1 \to 2 \to 6 \to 24$.
À la fin, $P = 24$: $1 \times 2 \times 3 \times 4$.

c) S prend les valeurs $0 \to 1 \to 5 \to 14 \to 30 \to 55$.
À la fin, $S = 55$: $1 + 4 + 9 + 16 + 25$.

d) c augmente de 1 seulement aux tours où k est un multiple de 3.
À la fin, $c = 6$: multiples de 3 : $3, 6, 9, 12, 15, 18$.

e) P prend les valeurs $0 \to 0 \to 0 \to 0 \to 0$.
À la fin, $P = 0$: *parti de* 0, *le produit reste* 0.

f) S prend les valeurs $0 \to 2 \to 4 \to 6 \to 8 \to 10$.
À la fin, $S = 10$: 5 tours, $+2$ à chaque tour.

g) u prend les valeurs $8 \to 12.0 \to 18.0 \to 27.0$.
À la fin, $u = 27.0$: $8 \times 1{,}5^3$.

h) S prend les valeurs $0 \to 1 \to 4 \to 9 \to 16 \to 25 \to 36 \to 49 \to 64 \to 81 \to 100$.
À la fin, $S = 100$: $1 + 3 + 5 + \cdots + 19$ (10 impairs).

Exercice 4 – Tableau de suivi ★★

Donne chaque variable avant la boucle, puis à la fin de chaque tour (des colonnes peuvent rester vides).

a)

```
u = 3
n = 0
while u < 100:
    u = 2 * u + 1
    n = n + 1
```

Tour	Avant	1	2	3	4	5
u	3	7	15	31	63	127
n	0	1	2	3	4	5

À chaque tour : `u = 2 * u + 1`, puis n augmente de 1.
Après le tour 5, $u = 127$: la condition `u < 100` devient fausse, la boucle s'arrête.
À la fin : $u = 127$, $n = 5$.

b)

```
a = 1
b = 1
for k in range(5):
    c = a + b
    a = b
    b = c
```

Tour	Avant	1	2	3	4	5
k		0	1	2	3	4
a	1	1	2	3	5	8
b	1	2	3	5	8	13

À chaque tour : `c` reçoit `a + b`, puis `a` prend la valeur de `b` et `b` celle de `c`.

5 tours : k prend les valeurs 0 à 4.

À la fin : `a` $= 8$, `b` $= 13$.

Exercice 5 – while : au bout de combien… ? (calculatrice autorisée) ★★

a) À chaque tour, `P = P * 1.03` et `n` compte les tours.

Après 6 tours, `P` $\approx 5\,970{,}26$: `P < 6000` est encore vrai.

Après 7 tours, `P` $\approx 6\,149{,}37$: `P < 6000` devient faux, la boucle s'arrête.

`n` $= 7$ et `P` $\approx 6\,149{,}37$.

Augmenter de 3 %, c'est multiplier par 1,03.

La population dépasse 6000 habitants pour la première fois au bout de 7 ans.

b) À chaque tour, `V = V * 0.85` et `n` compte les tours.

Après 4 tours, `V` $\approx 10\,440{,}12$: `V >= 10000` est encore vrai.

Après 5 tours, `V` $\approx 8\,874{,}11$: `V >= 10000` devient faux, la boucle s'arrête.

`n` $= 5$ et `V` $\approx 8\,874{,}11$.

Baisser de 15 %, c'est multiplier par 0,85.

La voiture vaut moins de la moitié de son prix au bout de 5 ans.

c) Augmenter de 2 %, c'est multiplier par 1,02.

On continue **tant que** $C < 1200$: c'est le contraire de l'arrêt ($C \geqslant 1200$).

```
C = 1000
n = 0
while C < 1200:
    C = C * 1.02
    n = n + 1
```

Après 9 ans, $C \approx 1\,195{,}09 < 1200$; après 10 ans, $C \approx 1\,218{,}99$.

À la fin, `n` $=$ `10` : il faut 10 ans.

d) Doubler, c'est multiplier par 2.

« Dépasser un million » s'arrête quand $B > 1\,000\,000$: on continue tant que $B \leqslant 1\,000\,000$.

```
B = 500
n = 0
while B <= 1000000:
    B = B * 2
    n = n + 1
```

Après 10 heures, $B = 512\,000$; après 11 heures, $B = 1\,024\,000$.

À la fin, `n` $= 11$ et `B` $=$ `1024000` : il faut 11 heures.

Exercice 6 – Programmes à trous ★★

Complète les **...** : a) $1 + 2 + \cdots + 50$; b) nombre de multiples de 7 de 1 à 100 ; c) $1 \times 2 \times \cdots \times 6$; d) nombre de

divisions par 2 pour que $u < 1$.

a)

```
S = 0
for k in range(1, 51):
    S = S + k
```

Pour aller jusqu'à 50, la borne de droite est 51.
À chaque tour, on ajoute k à la somme : `S = S + k`.
À la fin, `S` $= 1275$.

b)

```
c = 0
for k in range(1, 101):
    if k % 7 == 0:
        c = c + 1
```

k est un multiple de 7 quand le reste de sa division par 7 est nul : `k % 7 == 0`.
À la fin, `c` $= 14$.

c)

```
P = 1
for k in range(1, 7):
    P = P * k
```

Un produit démarre à 1.
Pour aller jusqu'à 6, la borne de droite est 7.
À la fin, `P` $= 720$.

d)

```
u = 1000
n = 0
while u >= 1:
    u = u / 2
    n = n + 1
```

On s'arrête quand $u < 1$, donc on continue tant que $u \geqslant 1$.
n compte les divisions : `n = n + 1`.
À la fin, `n` $= 10$.

Exercice 7 – Deux placements ★★★

1. Augmenter de 3 %, c'est multiplier par $1 + \dfrac{3}{100} = 1{,}03$.
   Après 1 an : $2000 \times 1{,}03 = 2060$ €.
   Après 2 ans : $2060 \times 1{,}03 = 2121{,}8$ €.

2. ```
   C = 2000
   n = 0
   while C < 3000:
       C = C * 1.03
       n = n + 1
   ```

   Le capital est multiplié par 1,03 chaque année : `C = C * 1.03`.
   On s'arrête quand $C \geqslant 3000$, donc on continue **tant que** $C < 3000$: `while C < 3000:`.
   ```

3. On ajoute 80 € chaque année : `C = C + 80`.

```
C = 2000
n = 0
while C < 3000:
    C = C + 80
    n = n + 1
```

Il donne `n` = 13.
Vérification : $2000 + 80 \times 12 = 2960 < 3000$ et $2000 + 80 \times 13 = 3040 \geqslant 3000$.

4. B dépasse 3000 € le premier (13 ans contre 14).
Mais au bout de 20 ans, A donne $2000 \times 1{,}03^{20} \approx 3\,612{,}22$ € et B donne $2000 + 80 \times 20 = 3600$ €.
A est devenu plus intéressant : un pourcentage s'applique à un capital qui grandit, alors que B ajoute toujours 80 €.

Exercice 8 – Le lac ★★★

1. Retirer 10 %, c'est multiplier par 0,9 ; puis on ajoute 200 t.
Après 1 an : $0{,}9 \times 5000 + 200 = 4500 + 200 = 4700$ t.
Après 2 ans : $0{,}9 \times 4700 + 200 = 4230 + 200 = 4430$ t.

2. `u = 0.9 * u + 200` calcule la masse de l'année suivante, et `n` compte les années.
La boucle tourne tant que la masse dépasse 2500 t.
La masse passe sous 2500 t (ou l'atteint) pour la première fois au bout de 18 ans.

3. La nouvelle valeur est $0{,}9u + 200$; on lui retire 2000 :
$0{,}9u + 200 - 2000 = 0{,}9u - 1800 = 0{,}9(u - 2000)$.
Donc $u - 2000$ est multiplié par 0,9 chaque année.
Au départ $u - 2000 = 5000 - 2000 = 3000 > 0$, et un nombre positif multiplié par 0,9 reste positif.
u reste toujours au-dessus de 2000, donc de 1500.
La condition `u > 1500` reste vraie : la boucle est infinie.

Exercice 9 – Trouve l'erreur ★★

Chaque programme ne fait pas ce qui est annoncé. Trouve l'erreur, puis corrige.

a) Le programme faux : il donne `P` = 8.
Ligne fautive (l. 2) : `for k in range(1, 4):`
Ligne corrigée : `for k in range(1, 5):`
`range(1, 4)` ne fait que 3 *tours.*

```
P = 1
for k in range(1, 5):
    P = P * 2
```

Version corrigée : `P` = 16.

b) Le programme faux : il donne `S` $= 8$.

Ligne fautive (l. 2) : `S = 0` est dans la boucle ; elle doit passer avant le `for`.

`S = 0` est dans la boucle : la somme repart de 0 *à chaque tour.*

```
S = 0
for k in range(1, 9):
    S = S + k
```

Version corrigée : `S` $= 36$.

c) Le programme faux : il ne s'arrête jamais.

Ligne fautive (l. 3) : `while u != 100:`

Ligne corrigée : `while u < 100:`

u vaut 81 *puis* 243 *: il ne vaut jamais* 100.

```
u = 1
n = 0
while u < 100:
    u = u * 3
    n = n + 1
```

Version corrigée : `n` $= 5$.

d) Le programme faux : il donne `c` $= 30$.

Ligne fautive (l. 5) : `c = c + 1` est mal décalée.

`c = c + 1` n'est pas décalé sous le `if` : il compte tous les tours.

```
c = 0
for k in range(1, 31):
    if k % 4 == 0:
        print(k)
        c = c + 1
```

Version corrigée : `c` $= 7$.

e) Le programme faux : il donne `n` $= 0$.

Ligne fautive (l. 3) : `while u > 1000:`

Ligne corrigée : `while u < 1000:`

La condition est celle de l'arrêt : $500 > 1000$ *est faux, la boucle ne tourne pas.*

```
u = 500
n = 0
while u < 1000:
    u = u * 1.1
    n = n + 1
```

Version corrigée : `n` $= 8$.

f) Le programme faux : il donne `S` $= 10$.

Ligne fautive (l. 3) : `S = k`

Ligne corrigée : `S = S + k`

`S = k` écrase la somme : il faut ajouter, `S = S + k`.

```
S = 0
for k in range(1, 11):
    S = S + k
```

Version corrigée : `S` $= 55$.

QCM – Six questions comme en contrôle

Une seule bonne réponse par question.

1. `for k in range(2, 9):` fait :

Réponse C : 7 tours
$9 - 2 = 7$: k va de 2 à 8.

2. Après `S = 0` et `for k in range(1, 4): S = S + k * k`, `S` vaut :

Réponse A : 14
k prend les valeurs 1, 2, 3 (il s'arrête avant 4).
$S = 1^2 + 2^2 + 3^2 = 1 + 4 + 9 = 14$.

3. Pour « tant que le capital C n'a pas atteint 5000 », on écrit :

Réponse D : `while C < 5000:`
La condition du `while` est le contraire de l'arrêt ($C \geqslant 5000$) ; `!=` risque une boucle infinie.

4. Une quantité baisse de 8 % par an. Dans la boucle, on écrit :

Réponse B : `Q = Q * 0.92`
Baisser de 8 %, c'est multiplier par $1 - 0{,}08 = 0{,}92$.

5. Pour calculer un produit dans une boucle, on initialise `P` :

Réponse D : à 1, avant la boucle
Partir de 0 donne toujours 0 ; dans la boucle, le produit repartirait à chaque tour.

6. Après `x = 2` et `for k in range(3): x = x * x`, `x` vaut :

Réponse A : 256
Tour 1 : $2 \times 2 = 4$; tour 2 : $4 \times 4 = 16$; tour 3 : $16 \times 16 = 256$.

Ce document est la propriété intellectuelle de Ben, professeur particulier (benprofparticulier.fr), protégée par le Code de la propriété intellectuelle (art. L. 111-1 et L. 122-4). Il est mis à disposition pour un usage personnel de révision. Toute reproduction, diffusion, mise en ligne ou revente, totale ou partielle, sans autorisation écrite est interdite et constitue une contrefaçon (art. L. 335-2).