

Plaquette 2 - Dérouler un programme et prévoir sa sortie

Algorithmique et Python — Première STI2D

Corrections détaillées

Boîte à outils

Lire un programme, c'est tenir un tableau d'états.

- **Affectation** : $x = 5$ range 5 dans x . L'instruction $x = x + 1$ lit l'ancienne valeur, ajoute 1, puis réécrit dans x .
- **Boucle bornée** : `for i in range(n)`: répète n fois, avec $i = 0, 1, \dots, n - 1$. `range(a, b)` parcourt $a, a + 1, \dots, b - 1$: la borne b est **exclue**.
- **Boucle non bornée** : `while condition`: répète **tant que** la condition est vraie. On teste **avant** chaque passage.
- **Test** : `if ... else ...`. Les comparaisons s'écrivent `==`, `!=`, `<=`, `>=`.
- **Listes** : `L[0]` est le **premier** élément, `len(L)` la longueur, `L.append(v)` ajoute v à la fin.
- **Opérations utiles** : `//` quotient entier, `%` reste, `**` puissance. Pour $n = 47$ et $d = 5$, $n // d$ vaut 9 et $n \% d$ vaut 2, car

$$n = d \times q + r \quad 47 = 5 \times 9 + 2$$

La méthode du tableau d'états. Une colonne par variable, une ligne par tour de boucle :

Tour	i	s
départ	—	0
1	0	0
2	1	1

⚠ Deux pièges classiques : `range(5)` s'arrête à 4, et l'ordre des affectations compte — $a = a + b$ puis $b = a - b$ n'utilise pas le même a .

Correction de l'exercice 1 - Suivre trois affectations

On déroule le programme ligne par ligne, en notant l'état des variables après chaque instruction.

Instruction	a	b
$a = 5$	5	—
$b = 3$	5	3
$a = a + b$	8	3
$b = a - b$	8	5

À la ligne 3, a reçoit $5 + 3 = 8$. À la ligne 4, a vaut **déjà** 8 : le membre de droite se calcule avec les valeurs courantes, donc

$$b = 8 - 3 = 5$$

Garder l'ancienne valeur $a = 5$ donnerait $b = 2$, ce qui est faux : l'ordre des affectations compte.

Réponse. $a = 8$ et $b = 5$.

Correction de l'exercice 2 - Échanger deux variables

On tient le tableau d'états, une colonne par variable.

Instruction	x	y	t
$x = 7$	7	—	—
$y = 2$	7	2	—
$t = x$	7	2	7
$x = y$	2	2	7
$y = t$	2	7	7

À la fin, $x = 2$ et $y = 7$: les deux valeurs ont été **échangées**. La variable t mémorise l'ancienne valeur de x avant que $x = y$ ne l'écrase. Sans elle, $x = y$ puis $y = x$ donnerait $x = y = 2$: la valeur 7 serait perdue.

Réponse. $x = 2$ et $y = 7$; t conserve l'ancienne valeur de x .

Correction de l'exercice 3 - Dérouler une boucle for

`range(5)` fait prendre à i les valeurs 0, 1, 2, 3, 4 : la borne 5 est exclue. À chaque tour, s augmente de i .

Tour	i	s après
départ	—	0
1	0	0
2	1	1
3	2	3
4	3	6
5	4	10

$$s = 0 + 1 + 2 + 3 + 4 = 10$$

L'erreur fréquente consiste à inclure 5 et à répondre 15.

Réponse. $s = 10$.

Correction de l'exercice 4 - Une boucle qui multiplie

La variable k prend les valeurs 1, 2, 3 et 4 : `range(1, 5)` commence à 1 et s'arrête **avant** 5. À chaque tour, p est multiplié par k :

- tour 1 : $p = 1 \times 1 = 1$;
- tour 2 : $p = 1 \times 2 = 2$;
- tour 3 : $p = 2 \times 3 = 6$;
- tour 4 : $p = 6 \times 4 = 24$.

$$p = 1 \times 2 \times 3 \times 4 = 24$$

Pour un produit, on initialise à 1, l'élément neutre de la multiplication : en partant de 0, p resterait égal à 0 à tous les tours.

Réponse. $p = 24$.

Correction de l'exercice 5 - Une évolution en boucle

La boucle s'exécute trois fois, et chaque tour multiplie u par 1,05, ce qui traduit une hausse de 5 % :

- tour 1 : $200 \times 1,05 = 210$;
- tour 2 : $210 \times 1,05 = 220,5$;
- tour 3 : $220,5 \times 1,05 = 231,525$.

Multiplier trois fois de suite par 1,05 revient à multiplier par $1,05^3$:

$$u = 200 \times 1,05^3 = 231,525$$

Vérification : $1,05^3 = 1,157\,625$ et $200 \times 1,157\,625 = 231,525$. Après trois ans, le capital atteint environ 231,53 €.

Réponse. $u = 231,525$, soit $200 \times 1,05^3$.

Correction de l'exercice 6 - Compter les tours d'une boucle while

La condition $v > 500$ est testée **avant** chaque passage ; à chaque tour, v est multiplié par 0,8 et n augmente de 1.

Tour	v avant	v après	n
1	1 000	800	1
2	800	640	2
3	640	512	3
4	512	409,6	4

Après le tour 3, $v = 512 > 500$: la boucle continue. Après le tour 4, $v = 409,6 \leq 500$: elle s'arrête.

$$n = 4$$

S'arrêter à 512 parce qu'il est « proche de 500 » serait une erreur : la condition est encore vraie.

Réponse. $n = 4$ (la valeur passe à 409,6).

Correction de l'exercice 7 - Une fonction avec un test

Une fonction avec un test n'exécute qu'une seule des deux branches, selon que la condition est vraie ou fausse.

- $\text{tarif}(80)$: la condition $80 \geq 100$ est fausse, c'est la branche *else* qui s'exécute, et la fonction renvoie 80 sans modification.
- $\text{tarif}(120)$: la condition $120 \geq 100$ est vraie, la fonction renvoie $120 \times 0,9 = 108$.

$$\text{tarif}(120) = 120 \times 0,9 = 108$$

La fonction modélise une remise de 10 % à partir de 100 € d'achat. Le cas limite se vérifie sur la condition : avec $\geq$, $\text{tarif}(100)$ renvoie 90, la remise s'applique dès 100 €.

Réponse. $\text{tarif}(80)$ renvoie 80 · $\text{tarif}(120)$ renvoie 108.

Correction de l'exercice 8 - Compter dans une liste

La boucle parcourt les **valeurs** de la liste une par une, et le compteur c , initialisé à 0 avant la boucle, augmente de 1 chaque fois que le test $v > 10$ est vrai.

Valeur	12	7	18	9	21	4
Test $v > 10$	vrai	faux	vrai	faux	vrai	faux
c après	1	1	2	2	3	3

Les valeurs comptées sont 12, 18 et 21 :

$$c = 3$$

Avec le test $v \geq 12$, on compterait encore 3 valeurs ; avec $v > 21$, aucune.

Réponse. $c = 3$.

Correction de l'exercice 9 - Somme et moyenne

La fonction calcule d'abord la somme des valeurs, puis la divise par leur nombre.

La somme. s part de 0 et reçoit chaque valeur : 14, puis $14 + 9 = 23$, puis $23 + 21 = 44$, puis $44 + 16 = 60$.

La division. La liste contient $\text{len}(L) = 4$ valeurs, donc la fonction renvoie

$$\frac{60}{4} = 15$$

Le *return* est placé **après** la boucle, au même niveau d'indentation que le *for* : s'il était à l'intérieur, la fonction s'arrêterait dès le premier tour et renverrait $\frac{14}{4} = 3,5$.

Réponse. La fonction renvoie 15.

Correction de l'exercice 10 - Quotient et reste

L'opérateur `//` donne le quotient entier et `%` le reste de la division euclidienne :

$$47 = 5 \times 9 + 2$$

donc $q = 9$ et $r = 2$: on remplit 9 cartons, et il reste 2 pièces.

Ces deux pièces demandent un carton supplémentaire, même incomplet : il faut 10 cartons. **Vérification** : $10 \times 5 = 50 \geq 47$, alors que $9 \times 5 = 45 < 47$.

En Python, le nombre de cartons s'écrit $q + 1$ lorsque $r > 0$, et q sinon. Répondre 9 revient à oublier les pièces restantes.

Réponse. $q = 9$, $r = 2$; il faut 10 cartons.

Correction de l'exercice 11 - Repérer une boucle infinie

La boucle teste la condition $v > 500$, mais ne modifie que n : v garde sa valeur 1 000, la condition reste toujours vraie, et la boucle ne s'arrête jamais. Une boucle *while* doit **modifier** la variable qu'elle teste, pour que la condition finisse par devenir fausse.

Correction : on ajoute dans la boucle la mise à jour de v , par exemple une baisse de 20 %.

```
v = 1000
n = 0
while v > 500:
    v = v * 0.8
    n = n + 1
```

v prend alors les valeurs 800, 640, 512, 409,6 : après 4 tours, $409,6 \leq 500$ et la boucle s'arrête avec $n = 4$.

Réponse. v n'est jamais modifiée : il faut ajouter $v = v * 0.8$ dans la boucle.

Correction de l'exercice 12 - Deux boucles imbriquées

Pour chacune des 3 valeurs de i (0, 1, 2), la boucle intérieure s'exécute entièrement, avec 4 valeurs de j . Le compteur augmente donc de 4 à chaque tour de la boucle extérieure :

$$c = 3 \times 4 = 12$$

On peut se représenter un tableau de 3 lignes et 4 colonnes : c compte ses cases. Plus généralement, deux boucles $\text{range}(a)$ et $\text{range}(b)$ imbriquées produisent $a \times b$ passages. L'erreur fréquente consiste à additionner, $3 + 4 = 7$, comme si les deux boucles s'exécutaient l'une après l'autre.

Réponse. $c = 12$.

Correction de l'exercice 13 - Un tableau de valeurs

La boucle fait prendre à x les valeurs 0, 1, 2, 3, et affiche à chaque fois x et $f(x)$. Dans $2 * x**2 - 3$, la puissance est calculée d'abord, puis la multiplication, puis la soustraction :

$$f(0) = -3 \quad f(1) = 2 - 3 = -1 \quad f(2) = 8 - 3 = 5 \quad f(3) = 18 - 3 = 15$$

x	0	1	2	3
f(x)	-3	-1	5	15

Le programme affiche quatre lignes : 0 -3, 1 -1, 2 5 et 3 15. Une erreur de priorité, calculer $(2x)^2$, donnerait $f(2) = 13$ au lieu de 5.

Réponse. Les couples (0 ; -3), (1 ; -1), (2 ; 5) et (3 ; 15).

Correction de l'exercice 14 - Construire une liste

La liste part vide, et *append* ajoute à la fin la valeur $3i + 1$ pour chaque i de 0 à 3 : on ajoute successivement 1, 4, 7 et 10.

$$L = [1, 4, 7, 10]$$

Ce sont les quatre premiers termes de la suite arithmétique de premier terme 1 et de raison 3. On accède aux éléments par leur indice, qui commence à 0 : $L[0]$ vaut 1 et $L[3]$ vaut 10. La liste a $\text{len}(L) = 4$ éléments ; $L[4]$ provoquerait une erreur.

Réponse. $L = [1, 4, 7, 10]$.

Correction de l'exercice 15 - Chercher le maximum

a) m part du premier élément, 23, puis prend chaque valeur plus grande que lui : $41 > 23$, donc $m = 41$; ni 18 ni 37 ne dépassent 41 ; $45 > 41$, donc $m = 45$; 29 ne change rien.

Valeur	23	41	18	37	45	29
m après	23	41	41	41	45	45

$$m = 45$$

b) Pour le minimum, on remplace le test $v > m$ par $v < m$ en gardant l'initialisation sur $L[0]$: on obtient 18. Initialiser m à 0 serait une erreur : avec des valeurs toutes positives, le « minimum » trouvé serait 0, qui n'est pas dans la liste. L'initialisation sur $L[0]$ fonctionne quels que soient les signes.

Réponse. a) $m = 45$ · b) remplacer $v > m$ par $v < m$ (on obtient 18).

Correction de l'exercice 16 - Synthèse : dérouler un barème de facturation

La fonction applique le premier tarif jusqu'à 200 kWh inclus, et le second **seulement** à la partie qui dépasse 200.

a) $150 \leq 200$: $facture(150) = 0,18 \times 150 = 27$ €.

b) $200 \leq 200$: $facture(200) = 0,18 \times 200 = 36$ €.

c) $320 > 200$: les 200 premiers kWh coûtent 36 €, les 120 suivants $0,22 \times 120 = 26,40$ € :

$$facture(320) = 36 + 26,40 = 62,40 \text{ €}$$

d) $53 > 36$: le client a dépassé 200 kWh. La part au-delà coûte $53 - 36 = 17$ €, soit $\frac{17}{0,22} \approx 77,3$ kWh : il a consommé environ 277 kWh. **Vérification** : $facture(277)$
 $= 36 + 0,22 \times 77 = 52,94$ €, très proche de 53 €.

Réponse. a) 27 € · b) 36 € · c) 62,40 € · d) environ 277 kWh.