

Exercices 2

Algorithmique et programmation en Python

Application : tarifs par paliers, compteurs, maximum, seuils, simulation

7 exercices · niveau ●●○ · durée conseillée : 2 h · sans ordinateur

★ À RETENIR — Ce qu'il faut savoir faire à la fin de la feuille

- Traduire un tarif par paliers par un `if / elif / else` et vérifier la cohérence aux valeurs frontières; enchaîner des appels de fonctions.
- Compter, accumuler et chercher un maximum sans liste avec `for`, atteindre un seuil avec `while`, simuler une expérience aléatoire et estimer une fréquence.

Exercice 1 Une facture d'eau

●●○

Une commune facture l'eau ainsi : un abonnement annuel de 24 €, puis 2,10 € par mètre cube jusqu'à 30 m³; 2,90 € par mètre cube pour la part comprise entre 30 et 80 m³; 3,50 € par mètre cube pour la part au-delà de 80 m³.

- 1.1 Calculer, à la main, la facture annuelle d'un abonné consommant 20 m³, puis 50 m³, puis 100 m³.
- 1.2 Dans le programme ci-contre, expliquer d'où vient le nombre 63 de la ligne 5, puis compléter la ligne 7.
- 1.3 Que renvoient `facture(30)` et `facture(80)` ? Vérifier que le tarif ne fait pas de « saut » à ces frontières.
- 1.4 Un abonné a payé 174 €. Déterminer sa consommation.

```

1 def facture(v):
2     if v <= 30:
3         p = 2.1 * v
4     elif v <= 80:
5         p = 63 + 2.9 * (v - 30)
6     else:
7         p = .....
8     return 24 + p

```

Programme 1 — v en m³, résultat en €

Exercice 2 Appels imbriqués

●●○

- 2.1 Calculer `f(4)`, `g(4)`, puis `h(4)`.
- 2.2 Calculer `g(f(4))`. Le comparer à `h(4)` : que peut-on en conclure ?
- 2.3 Exprimer `h(x)` en fonction de `x`, puis résoudre l'équation $h(x) = 7$.
- 2.4 Écrire une fonction `k` telle que `k(x)` calcule `g(f(x))`. Exprimer `k(x)` en fonction de `x` et calculer `k(2)`.

```

1 def f(x):
2     return 3 * x - 5
3
4 def g(x):
5     return x * x
6
7 def h(x):
8     return f(g(x))

```

Programme 1 — Trois fonctions

Exercice 3 Compter et accumuler

●●○

- 3.1 Expliquer le rôle des variables `c` et `s`, puis donner ce que renvoie `compte(50)`.
- 3.2 Que renvoie `compte(100)` ? On pourra utiliser $1+2+\dots+14 = 105$.
- 3.3 Modifier le programme pour ne compter que les multiples de 7 qui **ne sont pas** multiples de 5 : combien entre 1 et 100 ?
- 3.4 Écrire une fonction `moyenne(n)` renvoyant la moyenne des multiples de 7 entre 1 et `n`. Que renvoie `moyenne(100)` ?

```

1 def compte(n):
2     c = 0
3     s = 0
4     for k in range(1, n + 1):
5         if k % 7 == 0:
6             c = c + 1
7             s = s + k
8     return c, s

```

Programme 1 — Multiples de 7

Exercice 4 Un bénéfice maximal, sans liste



Une entreprise fabrique q meubles par jour, avec $0 \leq q \leq 40$. Son bénéfice quotidien, en euros, est $B(q) = -q^2 + 40q - 300$.

- 4.1 Calculer $B(0)$, $B(10)$, $B(20)$ et $B(30)$. Interpréter les valeurs de $B(0)$ et de $B(10)$.
- 4.2 Compléter la ligne 7 pour que `maximum()` renvoie la production la plus rentable et le bénéfice associé.
- 4.3 Développer $-(q - 20)^2 + 100$; en déduire ce que renvoie `maximum()`.
- 4.4 Écrire une fonction `rentable()` utilisant une boucle `while` et renvoyant le plus petit entier q tel que $B(q) \geq 0$.

```
1 def B(q):
2     return -q ** 2 + 40 * q - 300
3
4 def maximum():
5     qm = 0
6     for q in range(0, 41):
7         if ..... :
8             qm = q
9     return qm, B(qm)
```

Programme 1 — Recherche d'un maximum

Exercice 5 Une population qui diminue



Une ville compte 25 000 habitants et perd 2 % de sa population chaque année.

- 5.1 Recopier et compléter le tableau ci-dessous (population arrondie à l'unité).

Année	1	2	3
Population			

- 5.2 On donne $0,98^{11} \approx 0,8007$ et $0,98^{12} \approx 0,7847$. Que renvoie `annees(20000)` ? Justifier sans machine.
- 5.3 Pourquoi la boucle s'arrête-t-elle toujours, quel que soit le seuil strictement positif ?
- 5.4 Un élève remplace la ligne 5 par `p = p - 2`. Que renvoie `annees(20000)` ? Pourquoi ce programme ne modélise-t-il plus la situation ?

```
1 def annees(seuil):
2     p = 25000
3     n = 0
4     while p >= seuil:
5         p = 0.98 * p
6         n = n + 1
7     return n
```

Programme 1 — Nombre d'années

Exercice 6 Lancer deux dés : simulation et fréquences



- 6.1 Quelles valeurs la fonction `somme` peut-elle renvoyer ? Que représente `frequence(n)` ?
- 6.2 En dénombrant les couples possibles des deux dés, calculer la probabilité d'obtenir une somme égale à 7. Donner sa valeur décimale arrondie à 10^{-3} .
- 6.3 Trois exécutions de `frequence(1000)` ont donné 0,171; 0,158; 0,164. Commenter, puis dire ce qu'on observerait avec `frequence(100000)`.
- 6.4 Modifier le programme pour estimer la probabilité d'obtenir une somme supérieure ou égale à 10, puis calculer cette probabilité exacte.

```
1 from random import randint
2
3 def somme():
4     return randint(1, 6) + randint(1, 6)
5
6 def frequence(n):
7     c = 0
8     for i in range(n):
9         if somme() == 7:
10             c = c + 1
11     return c / n
```

Programme 1 — Simulation

Exercice 7 Trois erreurs à corriger



Chacun des trois programmes contient **une** erreur : l'indiquer et écrire le programme corrigé. Le programme b) doit afficher $1 + 2 + 3 + 4$ et le programme c) les entiers de 1 à 10.

a)

```
1 n = 5
2 if n == 5:
3     print("cinq")
```

b)

```
1 s = 0
2 for i in range(1, 5):
3     s = s + i
4     print(s)
```

c)

```
1 for i in range(1, 10):
2     print(i)
```