

Algorithmique et programmation Python

Corrigé du contrôle - Terminale

Durée conseillée : 45 minutes. Barème sur 20 points. Calculatrice interdite.

1 Lecture d'un programme à seuil / 4 pts

Barème : 4 points (1,5 + 1 + 1,5).

a. (1,5 pt) Tableau des valeurs :

Passage	0 (départ)	1	2	3	4	5
n	0	1	2	3	4	5
u	3	7	15	31	63	127

La condition $u < 100$ est vraie jusqu'à $u = 63$; après le 5^e passage, $u = 127 \geq 100$: on sort.

b. (1 pt) Le programme affiche **5 127**.

c. (1,5 pt) Par récurrence : $u_0 = 3 = 2^2 - 1$ (initialisation). Si $u_n = 2^{n+2} - 1$, alors $u_{n+1} = 2 \times (2^{n+2} - 1) + 1 = 2^{n+3} - 1$: la propriété est héréditaire. Donc $u_n = 2^{n+2} - 1$. La sortie signifie que 5 est le plus petit entier n tel que $u_n \geq 100$ ($2^7 - 1 = 127 \geq 100$ alors que $2^6 - 1 = 63 < 100$).

2 Fonctions et listes / 4 pts

Barème : 4 points (1,5 + 1,5 + 1).

```
def moyenne(L):
    return sum(L) / len(L)
```

```
def nb_sup(L, s):
    c = 0
    for v in L:
        if v > s:
            c = c + 1
    return c
```

```
L = [12, 7, 15, 9, 14]
```

```
print(moyenne(L), nb_sup(L, 10), [2 * v for v in L if v % 2 ==
```

0])

Sortie obtenue :

11.4 3 [24, 28]

- (1,5 pt : 1 pt pour la fonction avec **return**, 0,5 pt pour la valeur) **moyenne(L)** = $(12 + 7 + 15 + 9 + 14) / 5 = 57 / 5 = 11,4$.
- (1,5 pt) Parmi 12, 7, 15, 9, 14, ceux qui dépassent 10 sont 12, 15, 14 : **nb_sup(L, 10)** renvoie 3.
- (1 pt) **[2 * v for v in L if v % 2 == 0]** : les éléments pairs sont 12 et 14, on obtient [24, 28].

3 Dichotomie / 4 pts

Barème : 4 points (1 + 2 + 1).

- (1 pt) f est dérivable sur $[1 ; 2]$ avec $f'(x) = 3x^2 + 1 > 0$: f est continue et strictement croissante. $f(1) = -1 < 0$ et $f(2) = 7 > 0$. D'après le théorème des valeurs intermédiaires, l'équation $f(x) = 0$ a une unique solution α dans $[1 ; 2]$.
- (2 pts : 1,5 pt pour le tableau, 0,5 pt pour l'encadrement)

Étape	a	b	m	f(m)
1	1,000	2,000	1,500	1,8750
2	1,000	1,500	1,250	0,2031
3	1,000	1,250	1,125	-0,4512

Au bout de 3 étapes : $1,125 \leq \alpha \leq 1,250$ (amplitude 0,125).

- (1 pt) L'amplitude après k étapes est $1 / 2^k$. $2^9 = 512 < 1000$ donc $1 / 2^9 > 10^{-3}$, alors que $2^{10} = 1024 > 1000$ donne $1 / 2^{10} < 10^{-3}$: il faut **10 étapes**.

4 Méthode des rectangles / 4 pts

Barème : 4 points (0,5 + 2 + 0,5 + 1).

- (0,5 pt) $h = (2 - 0) / 4 = 0,5$; abscisses 0 ; 0,5 ; 1 ; 1,5 ; 2.
- (2 pts) S_{inf} (extrémités gauches, la fonction étant croissante) = $0,5 \times (0^2 + 0,5^2 + 1^2 + 1,5^2) = 0,5 \times 3,5 = 1,75$. S_{sup} (extrémités droites) = $0,5 \times (0,5^2 + 1^2 + 1,5^2 + 2^2) = 0,5 \times 7,5 = 3,75$.
- (0,5 pt) $1,75 \leq I \leq 3,75$, et $8/3 \approx 2,67$ appartient bien à $[1,75 ; 3,75]$.
- (1 pt) $S_{\text{sup}} - S_{\text{inf}} = h \times (f(2) - f(0)) = (2 / n) \times 4 = 8 / n$ (pour $n = 4 : 2 = 3,75 - 1,75$, cohérent). $8 / n < 0,01$ équivaut à $n > 800$: le plus petit entier est **n = 801**.

5 Dénombrement et simulation / 4 pts

Barème : 4 points (1,5 + 1 + 1,5).

- a. (1,5 pt) Les couples de somme 9 sont (3 ; 6), (4 ; 5), (5 ; 4), (6 ; 3) : 4 cas favorables sur 36 cas équiprobables, donc $P(S) = 4/36 = 1/9 \approx 0,111$.
- b. (1 pt) Le programme simule 3 000 doubles lancers et calcule la fréquence de S : on obtient 0.103. Cette valeur est proche de $1/9 \approx 0,1111$ mais n'a aucune raison d'être égale à $P(S)$: une simulation fournit une *estimation* soumise aux fluctuations d'échantillonnage (de l'ordre de $1 / \sqrt{3000} \approx 0,018$ au maximum).
- c. (1,5 pt) Les 50 doubles lancers sont indépendants, chacun est une épreuve de Bernoulli de paramètre $1/9$: X suit la loi binomiale $B(50 ; 1/9)$ (0,5 pt). $E(X) = 50 / 9 \approx 5,56$ (0,5 pt). $P(X \geq 1) = 1 - P(X = 0) = 1 - (8/9)^{50} \approx 0,997$ (0,5 pt).