Exercices sur l'algorithmique en Python en terminale spé maths

CAMPUS XYZ

Exercice 1 — Que renvoie ce programme ? ●○○

```
def mystere(n):
    s = 0
    for k in range(1, n + 1):
        s = s + 2*k - 1
    return s
```

a. Que renvoient `mystere(1)`, `mystere(3)` et `mystere(5)` ?

b. Conjecture ce que renvoie `mystere(n)`.

Exercice 2 — Calculer un terme de suite ●○○

La suite (u_n) est définie par $u_0 = 2$ et $u_{n+1} = 0{,}5\,u_n + 3$. Complète la fonction pour qu'elle renvoie u_n, puis donne u_3.

```
def u(n):
    v = ........
    for k in range(n):
        v = ........
    return v
```

Exercice 3 La liste des premiers termes ●○○

Écris une fonction `termes(n)` qui renvoie la liste $[u_0, u_1, \quad, u_n]$ de la suite de l'exercice 2.

Exercice 4 Calculer une somme ●○○

Écris une fonction `somme(n)` qui renvoie $1^2 + 2^2 + \cdots + n^2$, puis vérifie qu'elle donne $\dfrac{n(n+1)(2n+1)}{6}$ pour $n = 10$.

Exercice 5 — Un algorithme de seuil ●●○

Une population de 500 bactéries augmente de $20\,\%$ par heure. Écris une fonction qui renvoie le nombre d'heures nécessaires pour dépasser $10\,000$ bactéries, puis retrouve le résultat par le calcul.

Exercice 6 — Factorielle et combinaisons ●●○

a. Écris une fonction `fact(n)` qui renvoie $n!$.

b. Écris une fonction `comb(n, k)` qui renvoie $\binom{n}{k}$ à l'aide de `fact`. Que renvoie `comb(10, 3)` ?

Exercice 7 Des listes en compréhension ●●○

Donne le contenu de chaque liste.

a. `[k**2 for k in range(6)]`

b. `[k for k in range(20) if k % 3 == 0]`

c. `[2**k for k in range(1, 6)]`

Exercice 8 Tester si un nombre est premier ●●○

```
def premier(n):
    if n < 2:
        return False
    d = 2
    while d * d <= n:
        if n % d == 0:
            return False
        d = d + 1
    return True
```

a. Pourquoi suffit-il de tester les diviseurs d tels que $d^2 \leqslant n$?

b. Écris une instruction qui renvoie la liste des nombres premiers inférieurs à 30.

Exercice 9 Estimer π au hasard ●●○

```
from random import random
def estimation(n):
    c = 0
    for i in range(n):
        x = random()
        y = random()
        if x**2 + y**2 <= 1:
            c = c + 1
    return 4 * c / n
```

a. Quelle est la probabilité qu'un point choisi au hasard dans le carré $[0\,;1]^2$ soit dans le quart de disque de rayon 1 ?

b. Explique pourquoi la fonction renvoie une valeur proche de π quand n est grand.

Exercice 10 — Moyenne et variance d'une liste ●●○

Écris une fonction `moyenne(L)` et une fonction `variance(L)` pour une liste de nombres L, en utilisant $V = \frac{1}{n}\sum x_i^2 - \bar{x}^2$. Teste-les avec `[2, 4, 4, 4, 5, 5, 7, 9]`.

Exercice 11 La suite de Syracuse ●●●

On part d'un entier $n \geqslant 1$. S'il est pair, on le divise par 2 ; sinon, on le multiplie par 3 et on ajoute 1. On recommence jusqu'à obtenir 1.

a. Écris la suite obtenue en partant de 7.

b. Écris une fonction `vol(n)` qui renvoie le nombre d'étapes pour arriver à 1.

Exercice 12 La méthode d'Euler ●●●

On approche la solution de $y' = y$, $y(0) = 1$, sur $[0\,;1]$ avec un pas $h = \dfrac{1}{n}$: $y_0 = 1$ et $y_{k+1} = y_k + h\,y_k$.

a. Montre que $y_n = \left(1 + \dfrac{1}{n}\right)^n$. Quelle valeur cherche-t-on à approcher ?

b. Écris une fonction `euler(n)` qui renvoie y_n. Donne y_{10} et y_{1000}.

Exercice 13 TYPE CONTRÔLE **Une somme qui grandit lentement** ●●●

Pour $n \geqslant 1$, on pose $H_n = 1 + \dfrac{1}{2} + \dfrac{1}{3} + \cdots + \dfrac{1}{n}$.

1. Calcule H_1, H_2, H_3 et H_4.
2. Montre que (H_n) est croissante.
3. Écris une fonction `rang(A)` qui renvoie le plus petit n tel que $H_n > A$.
4. `rang(5)` renvoie 83. On admet que $H_n \approx \ln n + 0{,}577$ pour n grand. Est-ce cohérent ?
5. Déduis de cette approximation une estimation de `rang(20)`. Pourquoi ne faut-il pas lancer ce calcul ?

Exercice 14 TYPE CONTRÔLE Une marche au hasard ●●●

Un point part de 0 sur une droite graduée. À chaque étape, il avance de 1 ou recule de 1 avec la même probabilité, indépendamment. On note X sa position après 100 étapes.

1. Complète la fonction qui simule la position finale.

```
from random import random
def marche(n):
    x = 0
    for k in range(n):
        if random() < 0.5:
            x = ........
        else:
            x = ........
    return x
```

2. On note Y le nombre de pas en avant. Quelle est la loi de Y ? Montre que $X = 2Y - 100$.

3. Calcule $E(X)$ et $V(X)$.

4. Majore $P\big(|X| \geqslant 20\big)$ avec l'inégalité de Bienaymé-Tchebychev.

5. Pourquoi X est-il toujours pair ?

Exercice 15 BAC 2025 Le restaurant universitaire ●●○

Bac spé maths 2025, Amérique du Sud, jour 1, exercice 1

Un étudiant mange tous les jours au restaurant universitaire. Ce restaurant propose des plats végétariens et des plats non végétariens.

- Lorsqu'un jour donné l'étudiant a choisi un plat végétarien, la probabilité qu'il choisisse un plat végétarien le lendemain est $0{,}9$.
- Lorsqu'un jour donné l'étudiant a choisi un plat non végétarien, la probabilité qu'il choisisse un plat végétarien le lendemain est $0{,}7$.

Pour tout entier naturel n, on note V_n l'évènement « l'étudiant a choisi un plat végétarien le n^{e} jour » et p_n la probabilité de V_n.

Le jour de la rentrée, l'étudiant a choisi le plat végétarien. On a donc $p_1 = 1$.

1.

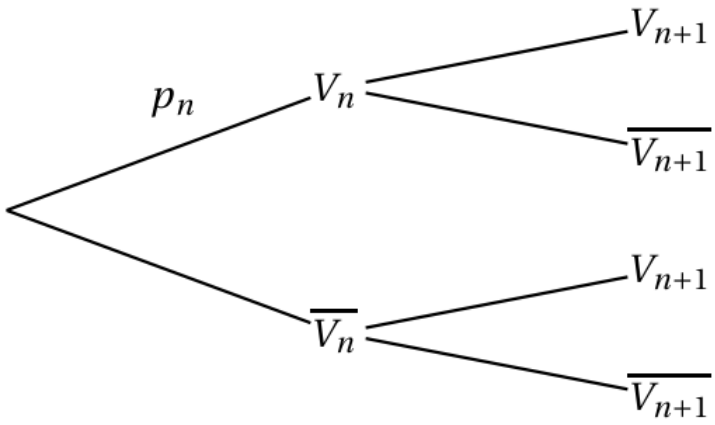
a. Indiquer la valeur de p_2.

b. Montrer que $p_3 = 0{,}88$. On pourra s'aider d'un arbre pondéré.

c. Sachant que le 3^{e} jour l'étudiant a choisi un plat végétarien, quelle est la probabilité qu'il ait choisi un plat non végétarien le jour précédent ?

On arrondira le résultat à 10^{-2}.

2. Recopier et compléter l'arbre pondéré ci-dessous:



3. Justifier que, pour tout entier naturel $n \geqslant 1$, $p_{n+1} = 0{,}2p_n + 0{,}7$.

4. On souhaite disposer de la liste des premiers termes de la suite (p_n) pour $n \geqslant 1$.

Pour cela, on utilise une fonction appelée `repas` programmée en langage Python dont on propose trois versions, indiquées ci-dessous.

Programme 1	Programme 2	Programme 3
1	def repas(n):	
2	p=1	
3	L=[p]	
4	for k in range(1,n):	
5	p = 0.2*p+0.7	
6	L. append(p)	
7	return(L)	
tabular	1	def repas(n):
2	p=1	
3	L=[p]	
4	for k in range(1,n+1):	
5	p = 0.2*p+0.7	
6	L. append(p)	
7	return(L)	
tabular	1	def repas(n):
2	p=1	
3	L=[p]	
4	for k in range(1,n):	
5	p = 0.2*p+0.7	
6	L.append(p+1)	
7	return(L)	
tabular		

1. Démontrer par récurrence que, pour tout naturel $n \geqslant 1$, $p_n = 0{,}125 \times 0{,}2^{n-1} + 0{,}875$.
2. En déduire la limite de la suite (p_n).

Exercice 16 BAC 2025 Des données binaires ●●●

Bac spé maths 2025, Polynésie, jour 2, exercice 1

Dans tout l'exercice, les probabilités seront, si nécessaire, arrondies à 10^{-3} près.

Une donnée binaire est une donnée qui ne peut prendre que deux valeurs: 0 ou 1.

Une donnée de ce type est transmise successivement d'une machine à une autre.

Chaque machine transmet la donnée reçue soit de manière fidèle, c'est-à-dire en transmettant l'information telle qu'elle l'a reçue (1 devient 1 et 0 devient 0), soit de façon contraire (1 devient 0 et 0 et devient 1).

La transmission est fidèle dans 90 % des cas, et donc contraire dans 10 % des cas.

Dans tout l'exercice, la première machine reçoit toujours la valeur 1.

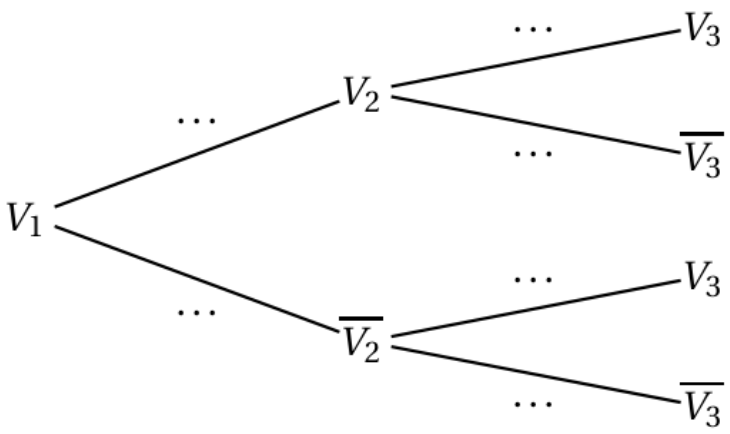
Partie A

Pour tout entier naturel $n \geqslant 1$, on note :

- V_n l'évènement : « la n-ième machine détient la valeur 1 »;
- $\overline{V_n}$ l'évènement : « la n-ième machine détient la valeur 0 ».

1.

a. Recopier et compléter l'arbre de probabilité ci-dessous.



b. Démontrer que $P(V_3) = 0{,}82$ et interpréter ce résultat dans le contexte de l'exercice.

c. Sachant que la troisième machine a reçu la valeur 1, calculer la probabilité que la deuxième machine ait aussi reçu la valeur 1.

2. Pour tout entier naturel $n \geqslant 1$, on note $p_n = P(V_n)$.

La première machine a reçu la valeur 1, on a donc $p_1 = 1$.

a. Démontrer que pour tout entier naturel $n \geqslant 1$:

$$p_{n+1} = 0{,}8p_n + 0{,}1.$$

b. Démontrer par récurrence que pour tout entier naturel $n \geqslant 1$,

$$p_n = 0{,}5 \times 0{,}8^{n-1} + 0{,}5.$$

c. Calculer la limite de p_n lorsque n tend vers l'infini. Interpréter ce résultat dans le contexte de l'exercice.

Partie B

Pour modéliser en langage Python la transmission de la donnée binaire décrite en début d'exercice, on considère la fonction `simulation` qui prend en paramètre un entier naturel n qui représente le nombre de transmissions réalisées d'une machine à une autre, et qui renvoie la liste des valeurs successives de la donnée binaire.

On donne ci-dessous le script incomplet de cette fonction.

On rappelle que l'instruction `rand()` renvoie un nombre aléatoire de l'intervalle [0 ; 1[.

```
[frame=single]
1   def simulation(n):
2       donnee = 1
3       liste = [donnee]
4       for k in range(n):
5           if rand() <0.1
6              donnee = 1 - donnee
7           liste.append(donnee)
9       return liste
```

Par exemple, `simulation(3)` peut renvoyer [1, 0, 0, 1].Cette liste traduit :

- qu'une donnée binaire a été successivement transmise trois fois entre quatre machines;
- la première machine qui détient la valeur 1 a transmis de façon contraire cette donnée à la deuxième machine ;
- la deuxième machine a transmis la donnée qu'elle détient de façon fidèle à la troisième ;
- la troisième machine a transmis de façon contraire la donnée qu'elle détient à la quatrième.

1. Déterminer le rôle des instructions des lignes 5 et 6 de l'algorithme ci-dessus.
2. Calculer la probabilité que simulation(4) renvoie la liste [1, 1, 1, 1, 1] et la probabilité que `simulation(6)` renvoie la liste [1, 0, 1, 0, 0, 1, 1].



LA FICHE MAGIQUE QUI VA AVEC

« Limites de suites »

Les limites de suites, utiles pour comprendre les algorithmes de seuil, sur une seule fiche.

Les fiches magiques terminale spé maths : campusxyz.fr/produit/les-fiches-magiques-terminale

CAMPUS XYZ

Corrigés

Exercices sur l'algorithmique en Python en terminale spé maths

Exercices sur l'algorithmique en Python en terminale spé maths : corrigés

CAMPUS XYZ

Exercice 1 — Que renvoie ce programme ? ●○○

a. La fonction additionne les nombres impairs $1 + 3 + \cdots + (2n - 1)$: $1\,;9\,;25$.

b. n^2 (se démontre par récurrence).

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

Exercice 2 — Calculer un terme de suite ●○○

```
def u(n):
    v = 2
    for k in range(n):
        v = 0.5*v + 3
    return v
```

$u_1 = 4, u_2 = 5, u_3 = 5{,}5$.

Exercice 3 — La liste des premiers termes ●○○

```
def termes(n):
    v = 2
    L = [v]
    for k in range(n):
        v = 0.5*v + 3
        L.append(v)
    return L
```

`termes(3)` renvoie **`[2, 4.0, 5.0, 5.5]`**.

Exercice 4 — Calculer une somme ●○○

```
def somme(n):
    s = 0
    for k in range(1, n + 1):
        s = s + k**2
    return s
```

`somme(10)` renvoie 385, et $\dfrac{10 \times 11 \times 21}{6} = 385$.

Exercice 5 Un algorithme de seuil ●●○

```
def heures():
    p = 500
    n = 0
    while p <= 10000:
        p = 1.2*p
        n = n + 1
    return n
```

$500 \times 1{,}2^n > 10\,000 \iff 1{,}2^n > 20 \iff n > \dfrac{\ln 20}{\ln 1{,}2} \approx 16{,}4$: **la fonction renvoie** 17.

Exercice 6 Factorielle et combinaisons ●●○

```
def fact(n):
    f = 1
    for k in range(2, n + 1):
        f = f * k
    return f

def comb(n, k):
    return fact(n) // (fact(k) * fact(n - k))
```

`comb(10, 3)` renvoie 120. On utilise la division entière `//` pour obtenir un entier.

Exercice 7 Des listes en compréhension ●●○

a. **`[0, 1, 4, 9, 16, 25]`**

b. Les multiples de 3 inférieurs à 20 : **`[0, 3, 6, 9, 12, 15, 18]`**

c. **`[2, 4, 8, 16, 32]`**

Exercice 8 Tester si un nombre est premier ●●○

a. Si $n = ab$ avec $2 \leqslant a \leqslant b$, alors $a^2 \leqslant ab = n$: un nombre non premier a toujours un diviseur d tel que $d^2 \leqslant n$.

b. `[n for n in range(30) if premier(n)]`, qui donne **`[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]`**.

Exercice 9 — Estimer π au hasard ●●○

a. Aire du quart de disque divisée par l'aire du carré : $\frac{\pi}{4}$.

b. $\frac{c}{n}$ est la fréquence des points dans le quart de disque ; d'après la loi des grands nombres, elle s'approche de $\frac{\pi}{4}$, donc $\frac{4c}{n}$ **s'approche de** π.

Exercice 10 — Moyenne et variance d'une liste ●●○

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

def variance(L):
    m = moyenne(L)
    return sum([x**2 for x in L]) / len(L) - m**2
```

Moyenne 5, variance 4 (écart type 2).

Exercice 11 — La suite de Syracuse ●●●

a. $7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1$: 16 **étapes**.

b.

```
def vol(n):
    c = 0
    while n != 1:
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3*n + 1
        c = c + 1
    return c
```

`vol(7)` renvoie 16 ; `vol(27)` renvoie 111. Personne n'a démontré que la suite atteint toujours 1 !

Exercice 12 — La méthode d'Euler ●●●

a. $y_{k+1} = (1+h)\,y_k$: suite géométrique, $y_n = (1+h)^n = \left(1+\dfrac{1}{n}\right)^n$. On approche $y(1) = \mathrm{e}$.

b.

```
def euler(n):
    y = 1
    for k in range(n):
        y = y + y / n
    return y
```

$y_{10} \approx 2{,}594$ **et** $y_{1000} \approx 2{,}717$, pour $\mathrm{e} \approx 2{,}718$.

Exercice 13 — TYPE CONTRÔLE — Une somme qui grandit lentement ●●●

1. $H_1 = 1\,;\, H_2 = 1{,}5\,;\, H_3 = \dfrac{11}{6} \approx 1{,}83\,;\, H_4 = \dfrac{25}{12} \approx 2{,}08$.

2. $H_{n+1} - H_n = \dfrac{1}{n+1} > 0$.

3.

```
def rang(A):
    n = 1
    h = 1
    while h <= A:
        n = n + 1
        h = h + 1/n
    return n
```

4. $\ln 83 + 0{,}577 \approx 4{,}996$: **c'est cohérent** ($H_{83} \approx 5{,}002$).

5. $\ln n > 19{,}42 \iff n > \mathrm{e}^{19{,}42} \approx 2{,}7 \times 10^8$: **environ 270 millions** de passages dans la boucle. H_n tend vers $+\infty$, mais extrêmement lentement.

Exercice 14 — TYPE CONTRÔLE — Une marche au hasard ●●●

1. `x = x + 1` puis `x = x - 1`.

2. Y **suit** $\mathcal{B}(100\,;0{,}5)$. Avec Y pas en avant et $100 - Y$ en arrière : $X = Y - (100 - Y) = 2Y - 100$.

3. $E(Y) = 50$, $V(Y) = 25$: $E(X) = 0$ et $V(X) = 4 \times 25 = 100$.

4. $P\big(|X| \geqslant 20\big) \leqslant \dfrac{100}{400} = \mathbf{0{,}25}$.

5. $X = 2Y - 100 = 2(Y - 50)$: **c'est un multiple de** 2.



La méthode pas à pas est dans la fiche magique **« Limites de suites »** (Les fiches magiques terminale spé maths, page 14).

Exercice 15 BAC 2025 Le restaurant universitaire ●●○

Un étudiant mange tous les jours au restaurant universitaire. Ce restaurant propose des plats végétariens et des plats non végétariens.

- Lorsqu'un jour donné l'étudiant a choisi un plat végétarien, la probabilité qu'il choisisse un plat végétarien le lendemain est $0{,}9$.
- Lorsqu'un jour donné l'étudiant a choisi un plat non végétarien, la probabilité qu'il choisisse un plat végétarien le lendemain est $0{,}7$.

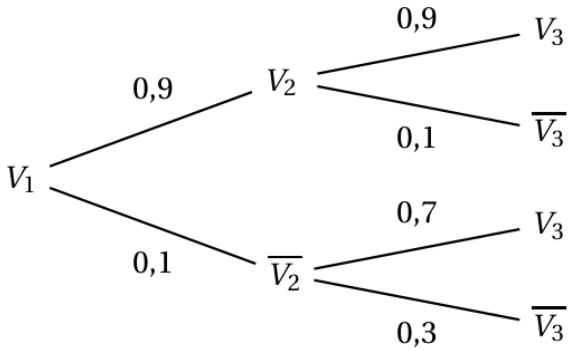
Pour tout entier naturel n, on note V_n, l'évènement « l'étudiant a choisi un plat végétarien le n-ième jour » et p_n la probabilité de V_n.

Le jour de la rentrée, l'étudiant a choisi le plat végétarien. On a donc $p_1 = 1$.

1.

a. Lorsqu'un jour donné l'étudiant a choisi un plat végétarien, la probabilité qu'il choisisse un plat végétarien le lendemain est $0{,}9$. Or $p_1 = 1$ donc $p_2 = 0{,}9$.

b. $p_3 = P(V_3)$. On représente la situation par un arbre pondéré.



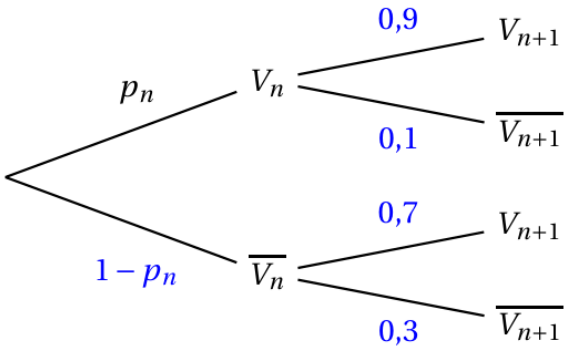
$\{V_2\ ,\ \overline{V_2}\}$ forme une partition donc, d'après la formule des probabilités totales:

$$p_3 = P(V_3) = P(V_2 \cap V_3) + P\left(\overline{V_2} \cap V_3\right) = P(V_2) \times P_{V_2}(V_3) + P\left(\overline{V_2}\right) \times P_{\overline{V_2}}(V_3)$$
$$= 0{,}9 \times 0{,}9 + 0{,}1 \times 0{,}7 = 0{,}88$$

c. Sachant que le 3^{e} jour l'étudiant a choisi un plat végétarien, la probabilité qu'il ait choisi un plat non végétarien le jour précédent est:

$$P_{V_3}\left(\overline{V_2}\right) = \frac{P\left(\overline{V_2} \cap V_3\right)}{P(V_3)} = \frac{0,1 \times 0{,}7}{0,88} = \frac{7}{88} \approx 0{,}08 \text{ au centième près.}$$

2. On complète l'arbre pondéré ci-dessous:



3. $\{V_n\ ,\ \overline{V_n}\}$ forme une partition donc, d'après la formule des probabilités totales:

$$p_{n+1} = P(V_{n+1}) = P(V_n \cap V_{n+1}) + P\left(\overline{V_n} \cap V_{n+1}\right) = P(V_n) \times P_{V_n}(V_{n+1}) + P\left(\overline{V_n}\right) \times P_{\overline{V_n}}(V_{n+1})$$
$$= p_n \times 0{,}9 + (1 - p_n) \times 0{,}7 = 0{,}9p_n + 0{,}7 - 0{,}7p_n = 0{,}2p_n + 0{,}7$$

4. On souhaite disposer de la liste des premiers termes de la suite (p_n) pour $n \geqslant 1$.

Pour cela, on utilise une fonction appelée `repas` programmée en langage Python dont on propose trois versions, indiquées ci-dessous.

Programme 1	Programme 2	Programme 3

```
1  def repas(n):
2      p=1
3      L=[p]
4      for k in range(1,n):
5          p = 0.2*p+0.7
6          L.append(p)
7      return(L)
tabular
```

```
1  def repas(n):
2      p=1
3      L=[p]
4      for k in range(1,n+1):
5          p = 0.2*p+0.7
6          L.append(p)
7      return(L)
tabular
```

```
1  def repas(n):
2      p=1
3      L=[p]
4      for k in range(1,n):
5          p=0.2*p+0.7
6          L.append(p+1)
7      return(L)
tabular
```

a. Le programme qui permet d'afficher les n premiers termes de la suite (p_n) est le n° 1.

En effet, le n° 2 donne $n+1$ termes, et le n° 3 donne des valeurs supérieures à 1.

b. Avec le programme n° 1, le résultat affiché pour $n=5$ sera la liste $[p_1\ ;\ p_2\ ;\ p_3\ ;\ p_4\ ;\ p_5]$.

$p_1 = 1$; $p_2 = 0{,}9$; $p_3 = 0{,}88$; $p_4 = 0{,}2p_3 + 0{,}7 = 0{,}2 \times 0{,}88 + 0{,}7 = 0{,}876$ et $p_5 = 0{,}2p_4 + 0{,}7 = 0{,}2 \times 0{,}876 + 0{,}7 = 0{,}8752$

Le résultat affiché par `repas(5)` sera donc $[1\ ;\ 0{,}9\ ;\ 0{,}88\ ;\ 0{,}876\ ;\ 0{,}8752]$.

5. On va démontrer par récurrence que $p_n = 0{,}125 \times 0{,}2^{n-1} + 0{,}875$, pour tout naturel $n \geqslant 1$.

- **6.Initialisation**

  Pour $n=1$, on a: $0{,}125 \times 0{,}2^{1-1} + 0{,}875 = 0{,}125 + 0{,}875 = 1$.

Or $p_1 = 1$, donc la propriété est vraie pour $n = 1$.

- **7.Hérédité**

  On suppose la propriété vraie au rang n c'est-à-dire que $p_n = 0{,}125 \times 0{,}2^{n-1} + 0{,}875$; c'est l'hypothèse de récurrence.

  $$\begin{aligned} p_{n+1} &= 0{,}2p_n + 0{,}7 = 0{,}2\left(0{,}125 \times 0{,}2^{n-1} + 0{,}875\right) + 0{,}7 \\ &= 0{,}125 \times 0{,}2^n + 0{,}2 \times 0{,}875 + 0{,}7 = 0{,}125 \times 0{,}2^n + 0{,}175 + 0{,}7 \\ &= 0{,}125 \times 0{,}2^n + 0{,}875 \end{aligned}$$

  La propriété est donc vraie au rang $n + 1$.

- **8.Conclusion**

  La propriété est vraie au rang 1 et elle est héréditaire pour tout $n \geqslant 1$; d'après le principe de récurrence, la propriété est vraie pour tout entier n non nul.

On a donc démontré par récurrence que $p_n = 0{,}125 \times 0{,}2^{n-1} + 0{,}875$ pour tout $n \geqslant 1$.

9. $-1 < 0{,}2 < 1$ donc $\lim\limits_{n \to +\infty} 0{,}2^{n-1} = 0$ donc $\lim\limits_{n \to +\infty} 0{,}125 \times 0,2^{n-1} + 0{,}875 = 0{,}875$

La limite de la suite (p_n) est donc $0{,}875$.

Le sujet complet du bac 2025 (Amérique du Sud, jour 1) corrigé →

Exercice 16 BAC 2025 Des données binaires ●●●

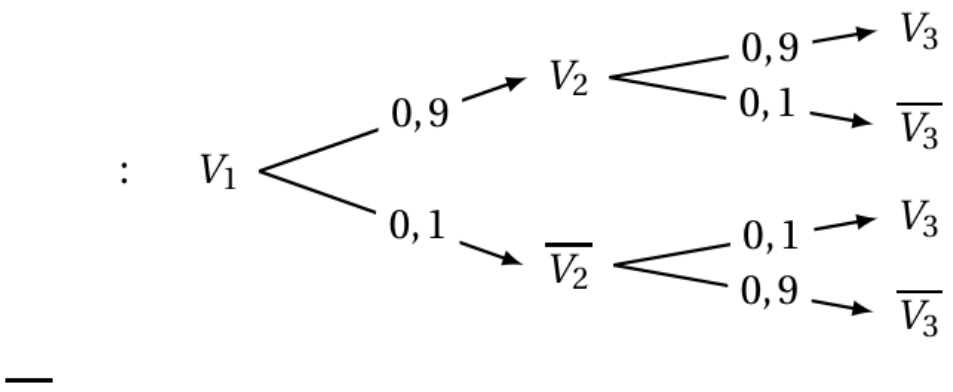
L'énoncé annonce des proportions, on va modéliser la situation de probabilité en assimilant les proportions à des probabilités.

Partie A

1.

a. D'après l'énoncé, on a donc : $P(V_1) = 1$ et, pour tout n naturel supérieur ou égal à 2 : $P_{V_{n-1}}(V_n) = 0{,}9$ et aussi $P_{\overline{V_{n-1}}}\left(\overline{V_n}\right) = 0{,}9$. (une transmission fidèle, c'est quand les machines $n-1$ et n détiennent la même valeur).

On obtient l'arbre ci-contre :



b. Les évènements V_2 et $\overline{V_2}$ partitionnent l'univers, donc, d'après la loi des probabilités totales :

$P(V_3) = P\left(V_3 \cap V_2\right) + P\left(V_3 \cap \overline{V_2}\right) = 0{,}9 \times 0{,}9 + 0{,}1 \times 0{,}1 = 0{,}81 + 0{,}01 = 0{,}82.$

On arrive bien à $P(V_3) = 0{,}82$.

c. On demande de calculer : $P_{V_3}(V_2)$.

D'après la définition, on a : $P_{V_3}(V_2) = \dfrac{P(V_3 \cap V_2)}{P(V_3)} = \dfrac{0,9 \times 0{,}9}{0,82} = \dfrac{81}{82} \approx 0{,}988.$

À 10^{-3} près, la probabilité que la machine 2 détienne 1, sachant que la machine 3 détient 1 est d'environ 0,988.

2.

a. Soit n un entier naturel non nul.

Les évènements V_n et $\overline{V_n}$ forment une partition de l'univers, donc, d'après la loi des probabilités totales :

$$\begin{aligned} p_{n+1} &= P(V_{n+1}) \\ &= P\left(V_{n+1} \cap V_n\right) + P\left(V_{n+1} \cap \overline{V_n}\right) \\ &= P_{V_n}\left(V_{n+1}\right) \times P\left(V_n\right) + P_{\overline{V_n}}\left(V_{n+1}\right) \times P\left(\overline{V_n}\right) \\ &= 0{,}9 \times P\left(V_n\right) + 0{,}1 \times P\left(\overline{V_n}\right) \\ &= 0{,}9 \times p_n + 0{,}1 \times (1 - p_n) \\ &= 0{,}9p_n + 0{,}1 - 0{,}1p_n \\ &= 0{,}8p_n + 0{,}1 \end{aligned}$$

On arrive donc bien à la relation de récurrence annoncée pour la suite (p_n).

b. Pour tout entier naturel non nul n, on pose :

P_n est l'affirmation : « $p_n = 0{,}5 \times 0{,}8^{n-1} + 0{,}5$ ».

Initialisation : on a, d'après l'énoncé : $p_1 = 1$,

et, par ailleurs, on a : $0{,}5 \times 0{,}8^{1-1} + 0{,}5 = 0{,}5 + 0{,}5 = 1.$

On constate donc que l'affirmation P_1 est vraie.

Hérédité : soit n naturel non nul tel que l'affirmation P_n est vraie, soit :

$$\begin{aligned} p_n = 0{,}5 \times 0{,}8^{n-1} + 0{,}5 &\implies 0{,}8p_n = 0{,}8\left(0{,}5 \times 0{,}8^{n-1} + 0{,}5\right) \\ &\implies 0{,}8p_n = 0{,}5 \times 0{,}8^{n-1} \times 0{,}8 + 0{,}4 \\ &\implies 0{,}8p_n = 0{,}5 \times 0{,}8^{n} + 0{,}4 \\ &\implies 0{,}8p_n + 0{,}1 = 0{,}5 \times 0{,}8^{n} + 0{,}4 + 0{,}1 \\ &\implies 0{,}8p_n + 0{,}1 = 0{,}5 \times 0{,}8^{n} + 0{,}5 \\ &\implies p_{n+1} = 0{,}5 \times 0{,}8^{n} + 0{,}5 \quad \text{c'est } P_{n+1} \end{aligned}$$

Conclusion : on a prouvé que l'affirmation P_1 est vraie, et que, n étant un naturel non nul, la véracité de P_n entraîne celle de P_{n+1}. En vertu du principe de récurrence, on peut conclure que :

$\forall n \in \mathbb{N}^*, \quad u_n = 0{,}5 \times 0{,}8^{n-1} + 0{,}5.$

Autrement dit, on a établi par récurrence une expression explicite du terme général p_n.

c. Pour tout n entier naturel non nul, on a :

$$\begin{aligned} 0{,}5 \times 0{,}8^{n-1} &= \frac{0,5}{0,8} \times 0{,}8^{n} \\ &= 0{,}625 \times 0{,}8^{n}. \end{aligned}$$

Comme on a : $-1 < 0{,}8 < 1$ la propriété des limites de suites géométriques donne :

$\lim\limits_{n \to +\infty} 0{,}8^n = 0$, puis $\lim\limits_{n \to +\infty} 0{,}625 \times 0{,}8^n = 0$.

Puis, par limite de la somme, on a : $\lim\limits_{n \to +\infty} p_n = \lim\limits_{n \to +\infty} 0{,}625 \times 0{,}8^n + 0{,}5 = 0{,}5.$

Ainsi, quand le nombre de machines tend vers plus l'infini, la probabilité que la dernière machine détienne la valeur 1 tend vers 0,5.

Partie B

1. L'instruction de la ligne 5 est une instruction conditionnelle : elle indique l'instruction de la ligne suivante ne s'exécutera qu'à la condition que le test `rand() < 0.1` renvoie la valeur `True`.

   Or, l'instruction `rand()` renvoie un nombre aléatoire dans $[0\ ;\ 1[$, donc la probabilité que ce nombre soit strictement inférieur à 0,1, c'est-à-dire dans l'intervalle $[0\ ;\ 0{,}1[$ est proportionnelle à l'amplitude de cet intervalle, la probabilité que cela arrive est donc de $\dfrac{0{,}1-0}{1-0}=0{,}1$.

   La ligne 5 va rendre l'exécution de la ligne 6 aléatoire, avec une probabilité que la ligne 6 s'exécute égale à 0,1.

   C'est-à-dire que la ligne 6 doit avoir pour effet de "mal transmettre" la dernière donnée détenue, qui est stockée dans la variable `donnee`. Cette variable contient 0 ou 1. Si elle contenait 0, après la ligne 6 elle contiendra $1-0=1$, soit la valeur contraire. Si elle contenait 1, après la ligne 6, elle contiendra $1-1=0$, là encore, la valeur contraire.

   La ligne 6 a donc pour effet de modifier la donnée, pour simuler une transmission contraire.

2. Si l'appel `simulation(4)` renvoie `[1,1,1,1,1]`, cela signifie que l'on a eu 4 transmissions fidèles, entre 5 machines.

   La probabilité que cela arrive est donc :

   $$0{,}9\times 0{,}9\times 0{,}9\times 0{,}9=0{,}9^4=0{,}6561\approx 0{,}656.$$

   De façon analogue l'appel `simulation(6)` renvoie une simulation de 6 transmissions entre 7 machines.

   Si l'appel renvoie `[1,0,1,0,0,1,1]`, cela signifie que les trois premières transmissions sont contraires (de 1 à 0, puis de 0 à 1, puis de 1 à 0), la quatrième transmission est fidèle (de 0 à 0), la cinquième est contraire (de 0 à 1) et la sixième et dernière est fidèle (de 1 à 1).

   La probabilité que cela arrive est donc :

   $$0{,}1\times 0{,}1\times 0{,}1\times 0{,}9\times 0{,}1\times 0{,}9=0{,}9^2\times 0{,}1^4=8{,}1\times 10^{-5}\approx 0{,}000.$$

   À 10^{-3} près, la probabilité d'obtenir `[1,1,1,1,1]` est d'environ 0,656, celle d'obtenir `[1,0,1,0,0,1,1]` est d'environ 0,000.

Le sujet complet du bac 2025 (Polynésie, jour 2) corrigé →



La méthode pas à pas est dans la fiche magique **« Limites de suites »** (Les fiches magiques terminale spé maths, page 14).

LES FICHES MAGIQUES TERMINALE SPÉ MATHS

Révise toute la terminale spé maths avec les fiches magiques



- **41 fiches claires et illustrées** : tout le programme de spécialité maths de terminale, 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-terminale

ANNALES BAC MATHS TERMINALE 2027

Prépare le bac avec les vrais sujets



- **75 sujets types** tirés des vrais sujets du bac, classés par chapitre.
- **3 bacs blancs complets** pour s'entraîner en conditions réelles.
- Des **corrigés avec la rédaction attendue**, des conseils et un aide-mémoire.

campusxyz.fr/produit/annales-bac-2027-terminale-spe-maths