Correction 1

Dans la boucle de l'algorithme, le fait d'inscrire l'instruction "*u prend la valeur* $u\times1{,}05$" permet de définir les termes de la suite (u_n) de premier terme 2 et de raison 1,05.
Ainsi, le terme de la suite (u_n) a pour expression :
$u_n = 2\times1{,}05^n$

L'algorithme initialise la variable S avec la valeur 2 *(qui est aussi la valeur du premier terme de la suite (u_n))*. Lors de l'exécution de la boucle, la variable S s'incrémente de la valeur de u à chacune de ses exécutions.

Ainsi, lors de la fin de l'exécution de la boucle, la variable S aura pour valeur :

$$\begin{aligned} S &= u_0 + u_1 + u_2 + \cdots + u_{20} \\ &= 2 + 2\times1{,}05 + 2\times1{,}05^2 + \cdots + 2\times1{,}05^{20} \\ &= 2\cdot\left(1 + 1{,}05 + 1{,}05^2 + \cdots + 1{,}05^{20}\right) = 2\cdot\frac{1-1{,}05^{20+1}}{1-1{,}05} \\ &= 2\cdot\frac{1-1{,}05^{21}}{-0{,}05} = -40\cdot\left(1-1{,}05^{21}\right) \\ &\approx 71{,}4385 \approx 71{,}44 \end{aligned}$$

Correction 2

Les deux lignes de l'algorithme :

```
U ← 50
U ← 1,2× U
```

Ainsi, les valeurs successives de la variable U représentent les termes de la suite (u_n) arithmétique de premier terme 50 et de raison 1,2.
Les termes de la suite (u_n) admettent pour valeur en fonction de leur rang n :
$u_n = 50\times1{,}2^n$

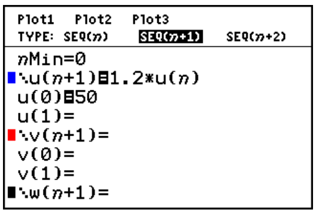
La boucle de l'algorithme va être exécuté tant que la clause :

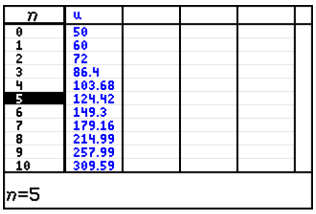
```
Tant que  U<120
```

est vérifiée.

Deux méthodes sont possibles :

- A l'aide de la calculatrice :





  On remarque que c'est à partir du rang 5 que les termes de la suite sont supérieurs à 120. La valeur de la variable `n` en fin d'exécution sera 5.

- A l'aide du logarithme népérien :

  Ainsi, la valeur de la variable `n` en fin d'exécution de l'algorithme est le rang du premier terme de la suite ne vérifiant plus cette clause. Résolvons l'inéquation :

$$\begin{aligned} u_n &\geqslant 120 \\ 50\times1{,}2^n &\geqslant 120 \\ 1{,}2^n &\geqslant \frac{120}{50} \\ 1{,}2^n &\geqslant 2{,}4 \end{aligned}$$

  La fonction logarithme étant strictement croissante :

$$\begin{aligned} \log\left(1{,}2^n\right) &\geqslant \log\left(2{,}4\right) \\ n\cdot\log\left(1{,}2\right) &\geqslant \log\left(2{,}4\right) \\ n &\geqslant \frac{\log\left(2{,}4\right)}{\log\left(1{,}2\right)} \end{aligned}$$

  On a la valeur approchée : $\frac{\ln 2{,}4}{\ln 1{,}2}\approx4{,}801$

$$n \geqslant 5$$

  Ainsi, en fin d'exécution de l'algorithme, la variable `n` aura la valeur 5.

La réponse correcte est la réponse **c.**.

Correction 3

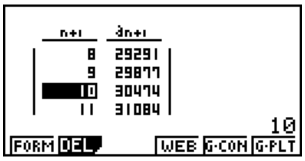
Valeur de `i`		2	3	4	5
Valeur de `u`	2 000	2016	2032,13	2048,39	2064,77
Valeur de `S`	2 000	4016	6048,13	8096,51	10161,29

Correction 4

1. Une augmentation de 2 % correspond à un coefficient multiplicateur valant :
$1 + \dfrac{2}{100} = 1 + 0{,}02 = 1{,}02$
Ainsi, les termes de la suite (u_n) vérifient la relation :
$u_{n+1} = 1{,}02\cdot u_n$

On en déduit que la suite (u_n) est la suite géométrique de premier terme 25 000 et de raison 1,02.

2. a. En utilisant la calculatrice, on a :



Ainsi, les termes de la suite (u_n) dépasseront la valeur 30 000 euros dès que le rang du terme dépassera 10.

b. Voici l'algorithme complété :

```
ℓ.1    n ← 0
ℓ.2    u ← 25 000
ℓ.3    Tant que  u<=30 000 faire
ℓ.4        n ← n+1
ℓ.5        u ← 1,02× u
ℓ.6    Fin Tant que
```

Correction 5

1. a. Voici l'algorithme complété :

```
U ← 50
Pour N allant de 1 à 24
    U ← U×0,9
Fin Pour
```


(b.) La suite (u_n) étant une suite géométrique de premier terme 50 et de raison 0,9, son terme de rang n admet pour expression :

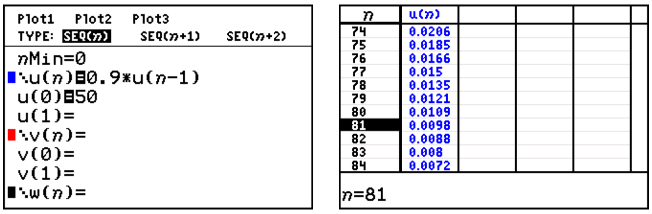
$$u_n = 50\times 0{,}9^n$$

(c.) Le terme de rang 24 admet pour expression :

$$u_{24} = u_0\times q^{24} = 50\times 0{,}9^{24}$$
$$\approx 3{,}9883 \approx 3{,}988$$

2. Deux approches sont possibles :

- A l'aide de la calculatrice :



Ainsi, le plus petit entier nature n vérifiant la comparaison $u_n < 0{,}01$ est 81.

- A l'aide d'une résolution d'une inéquation et de la fonction logarithme :

$$u_n < 0{,}01$$
$$50\times 0{,}9^n < 0{,}01$$
$$0{,}9^n < \frac{0{,}01}{50}$$
$$0{,}9^n < \frac{0{,}01}{50}$$
$$0{,}9^n < 0{,}0002$$

La fonction logarithme étant strictement croissante :

$$\ln\left(0{,}9^n\right) < \ln\left(0{,}0002\right)$$
$$n\cdot \ln\left(0{,}9\right) < \ln\left(0{,}0002\right)$$

Le nombre $\ln 0{,}9$ est strictement négatif :

$$n > \frac{\ln\left(0{,}0002\right)}{\ln\left(0{,}9\right)}$$

De la valeur approchée $\frac{\ln 0{,}0002}{\ln 0{,}9} \approx 80{,}83$, on a :

$$n \geqslant 81$$

Ainsi, le plus petit entier nature n vérifiant la comparaison $u_n < 0{,}01$ est 81.

3. (a.)
- Ce n'est pas l'algorithme 2, car S est affecter à chaque passage de la boucle de la valeur du terme u_n. S ne représentera pas la somme des termes de la suite (u_n).
- Ce n'est pas l'algorithme 3, car S est inialisé par la valeur 50 et lors du premier traitement de la boucle *(pour N affecter de* 0*)*, on ajoutera la valeur $50\times 0{,}9^0 = 50$. Ainsi, dans la variable S, on aura ajouter deux fois la valeur de u_0.

(b.) La somme S_{24} est égale :

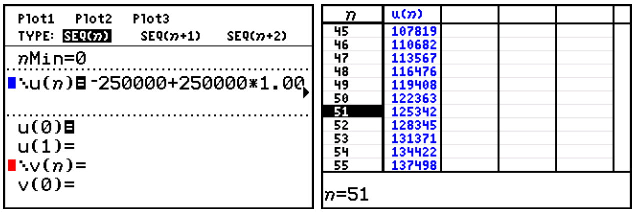
$$S_{24} = u_0 + u_1 + u_2 + \cdots + u_{24}$$
$$= 50 + 50\times 0{,}9 + 50\times 0{,}9^2 + \cdots + 50\times 0{,}9^{24}$$
$$= 50\cdot\left(1 + 0{,}9 + 0{,}9^2 + \cdots + 0{,}9^{24}\right) = 50\cdot\frac{1-0{,}9^{25}}{1-0{,}9}$$
$$= 50\cdot\frac{1-0{,}9^{25}}{0{,}1} = 500\cdot\left(1-0{,}9^{25}\right) \approx 464{,}10 \approx 464$$

Correction 6

1. De la comparaison $1{,}008 > 1$, on en déduit la limite :

$$\lim_{n\mapsto+\infty} 1{,}008^n = +\infty \Longrightarrow \lim_{n\mapsto+\infty} 250\,000\times 1{,}008^n = +\infty$$
$$\Longrightarrow \lim_{n\mapsto+\infty} -250\,000 + 250\,000\times 1{,}008^n = +\infty$$

2. D'après la calculatrice :



C'est à partir du rang 51 que la suite S_n a une valeur supérieure à 125 000.

