

Terminale ES/Algorithmes

1. Structure répétitive :

Exercice 6145



On considère la suite (u_n) définie pour tout entier naturel :
 $u_0 = 115$; $u_{n+1} = 0.4 \cdot u_n + 120$

On considère les trois parties d'algorithmes ci-dessous présentant chacune une fonction `terme()` prenant un entier n supérieur ou égal à 1 pour argument :

Algorithme 1

```
Fonction terme(n)
  Pour i de 1 à n
    U ← 0,4×U+120
  Fin Pour
  Renvoyer U
```

Algorithme 2

```
Fonction terme(n)
  Pour i de 1 à n
    U ← 115
    U ← 0,4×U+120
  Fin Pour
  Renvoyer U
```

Algorithme 3

```
Fonction terme(n)
  U ← 115
  Pour i de 1 à n
    U ← 0,4×U+120
  Fin Pour
  Renvoyer U
```

Expliquer pourquoi les fonctions `terme()` des deux premiers algorithmes, appelées avec l'entier n , ne renvoient pas le terme de la suite (u_n) de rang n .

Exercice 6147



On considère la suite (u_n) définie pour tout entier naturel :
 $u_0 = 50000$; $u_{n+1} = 0,95 \cdot u_n + 3000$ pour tout $n \in \mathbb{N}$

On considère les trois parties d'algorithmes présentant chacune une fonction `f` :

Fonction `terme1(A)`

```
n ← 0
U ← 50000
Tant que U < A
  n ← n+1
  U ← 0,95·U+3000
Fin Tant que
Renvoyer n
```

Fonction `terme2(n)`

```
U ← 50000
Pour i allant de 1 à n
  U ← 0,95·U+3000
Fin Tant que
Renvoyer U
```

Fonction `terme3(n)`

```
U ← 50000
Pour i allant de 0 à n
  U ← 0,95·U+3000
Fin Tant que
Renvoyer U
```

Parmi tous les fonctions ci-dessus, lequel permet, par une exécution pas à pas et en observant les valeurs successives prises par la variable `U`, d'obtenir toutes les valeurs des termes de la suite (u_n) pour les rangs allant de 0 à n .

2. Structure répétitive: étude de la condition d'arrêt :

Exercice 6150



On étudie l'évolution de la population d'une ville, depuis le 1^{er} janvier 2008.

On considère la population de cette ville à partir du 1^{er} janvier 2008 par la fonction f définie sur $[0; +\infty[$ par :

$$f(x) = \frac{3}{1 + 2e^{-0,05x}}$$

où x désigne le nombre d'années écoulées depuis le 1^{er} janvier 2008 et $f(x)$ le nombre d'habitants en centaines de milliers.

On admet que f est croissante sur $[0; +\infty[$

On considère l'algorithme suivant :

```
X ← 0
Tant que f(X) ≤ 2
  X ← X+1
Fin Tant que
```

Si on exécute cet algorithme alors en fin d'exécution la variable `X` aura pour valeur 28. Interpréter ce résultat dans le contexte de ce problème.

3. Structure répétitive: étude du seuil d'une suite :

Exercice 6146



On considère la suite (a_n) définie par :

$$a_0 = 2500$$
 ; $a_{n+1} = 0,8 \cdot a_n + 400$

1. On admet que le terme général de la suite (a_n) admet pour expression :

$$a_n = 500 \times 0,8^n + 2000$$

En déduire la limite de la suite (a_n) .

2. On propose l'algorithme suivant :

```
N ← 0
A ← 2500
Tant que A-2000 > 50
  A ← A × 0,8 + 400
  N ← N + 1
Fin Tant que
```

- Expliquer ce que représente la valeur de la variable N à la fin de l'exécution de cet algorithme.
- A l'aide de la calculatrice, déterminer la valeur de la variable N à la fin de l'exécution de cet algorithme et interpréter le résultat.

Exercice 7017



On considère l'algorithme suivant :

4. Structure répétitive : trouver la condition d'arrêt :

Exercice 6149



On considère la suite (u_n) définie par :
 $u_0 = 20$; $u_{n+1} = 0,92 \cdot u_n + 3$

- On admet que le terme général de la suite (u_n) admet pour expression :
 $u_n = -17,5 \times 0,92^n + 37,5$

En déduire la limite de la suite (u_n) .

- Recopier et compléter l'algorithme suivant afin qu'à la fin de son exécution la variable N représente le rang à partir duquel les termes de la suite auront une valeur supérieur ou égale à 25.

```
U ← 20
N ← 0
Tant que ...
  U ← 0,92 × U + 3
  N ← N + 1
Fin Tant que
```

- A l'aide de la calculatrice, déterminer le rang à partir duquel les termes de la suite (u_n) seront pour la première fois supérieur ou égal à 25.

Exercice 6151



- Déterminer par le calcul la plus petite valeur de l'entier naturel n telle que :
 $250 + 1250 \times 0,8^n < 500$

- On considère la suite (u_n) définie par :
 $u_0 = 1500$; $u_{n+1} = 0,8 \cdot u_n + 50$ pour tout $n \in \mathbb{N}$

```
U ← 4
N ← 0
Tant que U < 40
  U ← 0,92 × U + 8
  N ← U + 1
Fin Tant que
```

- Recopier le tableau suivant et le compléter en ajoutant autant de colonnes que nécessaire.

Les valeurs de U seront arrondies au dixième.

Valeur de U	4
Valeur de N	0
Condition $U < 40$	vraie

- Donner la valeur de la variable N à la fin de l'exécution de cet algorithme.

Compléter l'algorithme ci-dessous afin qu'à la fin de son exécution la variable u a pour valeur la solution obtenue à la question précédente :

```
u ← 1500
n ← 0
Tant que ..... faire
  u ← .....
  n ← .....
Fin Tant que
```

Exercice 7856



On considère la suite (u_n) définie par $u_0 = 65$ et pour tout entier naturel n : $u_{n+1} = 0,8 \cdot u_n + 18$

On admet que : $u_n = 90 - 25 \times 0,8^n$

On considère l'algorithme ci-dessous :

```
ligne 1 u ← 65
ligne 2 n ← 0
ligne 3 Tant que .....
ligne 4   n ← n + 1
ligne 5   u ← 0,8 × u + 18
ligne 6 Fin Tant que
```

- Recopier et compléter la ligne 3 de cet algorithme afin qu'il détermine le plus petit entier naturel n tel que :
 $u_n \geq 85$.
- Quelle est la valeur de la variable n à la fin de l'exécution de l'algorithme ?
- Retrouver par le calcul le résultat de la question précédente en résolvant l'inéquation $u_n \geq 85$