

Plaquette 4 – Seuils, algorithmes et tableur

Suites numériques et modèles — Terminale STMG

Exercices

Méthode

Quand aucune formule ne donne directement la réponse, on **fait tourner** le modèle : boucle « tant que » en Python, recopie vers le bas au tableur. Cette plaquette entraîne les deux gestes.

Exercice 1 – Compléter un algorithme de seuil

Un capital de 2000 € croît de 6 % par an. Compléter l'algorithme qui renvoie le nombre d'années pour dépasser 4000 €.

```
def seuil():  
    u = 2000  
    n = 0  
    while u < ...:  
        u = u * ...  
        n = n + 1  
    return n
```

Exercice 2 – Un seuil vers le bas

Un stock de 5000 pièces baisse de 15 % par mois. Écrire l'algorithme qui renvoie le nombre de mois pour passer sous 1000 pièces.

Exercice 3 – Trace d'exécution

Un abonné démarre à 100 et croît de 15 % par période. Dresser le tableau des cinq premières valeurs, arrondies au centième.

Exercice 4 – Formules de tableur

Une production démarre à 1200 unités et augmente de 150 par mois. La colonne A contient les rangs à partir de A2, la colonne B les productions.

- Quelle formule écrire en B2 ?
- Quelle formule en B3, à recopier vers le bas ?
- Quelle formule donnerait directement p_n depuis A ?

Exercice 5 – Tableur et modèle géométrique

Un capital de 2000 € croît de 6 % par an, la colonne B portant les capitaux.

- Donner la formule de B3.
- Donner la formule directe à partir du rang en A3.
- Le taux est en cellule D1. Comment écrire B3 ?

Exercice 6 – Que renvoie cet algorithme ?

```
def mystere():  
    u = 500  
    n = 0  
    while u < 800:  
        u = u + 60  
        n = n + 1  
    return n
```

Déterminer la valeur renvoyée.

Exercice 7 – Modifier un algorithme

L'algorithme de l'exercice 1 renvoie le rang. Le modifier pour qu'il renvoie le **capital** obtenu, puis pour qu'il renvoie les **deux**.

Exercice 8 – Un cumul par boucle

La production vaut 1200 le premier mois puis augmente de 150 par mois. Écrire un algorithme qui renvoie la production cumulée sur n mois, puis l'appliquer à $n = 20$.

Exercice 9 – Un seuil sur un cumul

Avec le même modèle, écrire un algorithme qui renvoie le nombre de mois pour que la production cumulée dépasse 50 000 unités.

Exercice 10 – Corriger un algorithme

```
def seuil():  
    u = 2000  
    n = 0  
    while u < 4000:  
        n = n + 1  
        u = u * 1.06  
    return u
```

L'objectif était de renvoyer le **rang**. Le corriger et dire ce que renvoie la version actuelle.

Exercice 11 – Un modèle mixte au tableur

Une base de clients suit $u_{n+1} = 0,9u_n + 500$, avec $u_0 = 8000$.

- Donner la formule de tableur.
- Calculer u_1 , u_2 et u_3 .
- Que se passe-t-il à long terme ?

Exercice 12 – Une épargne annuelle

On verse 15 000 € par an sur un compte à 3 % : $u_{n+1} = 1,03u_n + 15\,000$, avec $u_0 = 0$.

Écrire l'algorithme qui renvoie le nombre d'années pour dépasser 100 000 €, et donner la réponse.

Exercice 13 – Un parc à renouveler

Un parc de 12 000 machines perd 12 % par an. Écrire l'algorithme qui renvoie le nombre d'années pour passer sous 6000 machines.

Exercice 14 – Comparer deux algorithmes

Les deux algorithmes suivants portent sur $u_0 = 1000$ et une hausse de 4 %.

```
def A():  
    u = 1000  
    for k in range(10):  
        u = u * 1.04  
    return u  
  
def B():  
    return 1000 * 1.04 ** 10
```

- a) Que renvoient-ils ?
- b) Lequel préférer, et pourquoi ?

Exercice 15 – Repérer une erreur d'algorithme

```
def seuil():  
    u = 5000  
    n = 0  
    while u > 1000:  
        n = n + 1  
    return n
```

Que se passe-t-il ? Corriger.

Exercice 16 – Synthèse : un plan d'investissement

Un fonds de 40 000 € croît de 5 % par an, avec un retrait de 3000 € en fin d'année :
 $u_{n+1} = 1,05 u_n - 3000$.

- a) Calculer u_1 , u_2 et u_3 .
- b) Écrire l'algorithme donnant l'année d'épuisement.
- c) Que vaudrait le point d'équilibre ?