

Exercice 1 Soit la suite (T_n) définie par $T_0 = -19$ et pour tout entier n , $T_{n+1} = 0,94T_n + 1,5$.

Le programme ci-contre, écrit en langage Python, doit renvoyer après son exécution la plus petite valeur de l'entier n pour laquelle $T_n \geq 10$. Compléter les lignes incomplètes afin que le programme renvoie la valeur attendue.

Remarque : T est la variable où sont stockées les valeurs successives des termes T_n et n est la variable où sont stockées les valeurs successives de l'entier n .

```
def seuil() :
    n = 0                (n = 0)
    T = -19              (Valeur de T0)
    while T < 10 :       (Tant que l'on n'a pas Tn ≥ 10 donc tant que Tn < 10)
        T = 0,94*T+1,5   (On calcule le terme suivant. Exemple : T1)
        n = n + 1        (On calcule l'indice du terme que l'on vient de calculer. Exemple : n = 1)
    return n             (On affiche la valeur de l'indice du terme qui a stoppé la boucle donc la première valeur de n
                        telle que Tn ≥ 10)
```

Exercice 2 Au 1^{er} janvier 2020, une centrale solaire possédait 10560 panneaux solaires.

On modélise l'évolution du nombre de panneaux solaires par la suite (U_n) définie par $U_0 = 10560$ et, pour tout entier naturel n , $U_{n+1} = 0,98U_n + 250$, où U_n est le nombre de panneaux solaires au 1^{er} janvier de l'année 2020 + n .

On souhaite savoir au bout de combien d'années le nombre de panneaux solaires sera strictement supérieur à 12000.

Compléter les lignes incomplètes afin que le programme renvoie la valeur attendue.

Remarque : u est la variable où sont stockées les valeurs successives des termes U_n et n est la variable où sont stockées les valeurs successives de l'entier n .

```
def seuil() :
    u = 10560            (Valeur de U0)
    n = 0                (Valeur de l'indice n = 0)
    while u <= 12000 :    (Tant que l'on n'a pas Un > 12000 donc tant que Un ≤ 12000)
        u = 0,98*u + 250 (On calcule le terme suivant. Exemple : U1)
        n = n+1          (On calcule l'indice du terme que l'on vient de calculer. Exemple : n = 1)
    return n             (On affiche la valeur de l'indice du terme qui a stoppé la boucle donc la première valeur de n
                        telle que Un > 12000)
```

Exercice 3 Un biologiste s'intéresse à l'évolution de la population d'une espèce animale sur une île du Pacifique.

Au début de l'année 2020, cette population comptait 600 individus.

On considère que l'espèce sera menacée d'extinction sur cette île si sa population devient inférieure ou égale à 20 individus.

Le biologiste modélise le nombre d'individus par la suite (U_n) définie par : $U_0 = 0,6$ et $U_{n+1} = 0,75U_n(1 - 0,15U_n)$

où pour tout entier naturel n , U_n désigne le nombre d'individus, en milliers, au début de l'année 2020 + n .

On admet que la suite (U_n) est décroissante et tend vers 0.

Le biologiste a programmé en langage Python la fonction **menace()** ci-contre. Le programme retourne la valeur 11.

Interpréter ce résultat dans le contexte de l'exercice.

Remarque : u et n sont les variables où sont stockées respectivement les valeurs successives des termes U_n et de l'entier n .

```
def menace() :  
    u = 0,6                ( $U_0 = 0,6$ )  
    n = 0                  ( $n = 0$ )  
    while u > 0,02 :      (Tant que  $U_n > 0,2$ , on exécute la boucle)  
        u = 0,75*u*(1-0,15*u)    (On calcule le terme suivant Exemple :  $U_1$ )  
        n = n+1                (On calcule l'indice du terme que l'on vient de calculer. Exemple :  $n = 1$ )  
    return n                (Dès que  $u \leq 0,02$  la boucle s'arrête donc dès que  $U_n \leq 0,02$ .)
```

Le programme retourne la valeur 11.

La valeur de n qui est retournée est l'indice du premier terme de la suite (U_n) qui est inférieure ou égale à 0,02.

Ce qui veut dire que U_{11} est le premier terme qui est inférieure ou égale à 0,02.

Réponse attendue : C'est en 2031 (car $2020 + 11$) que l'espèce sera menacée d'extinction.

Exercice 4 On considère les deux suites suivantes :

- la suite (U_n) définie pour tout entier n par : $U_n = \frac{8n-4}{n+1}$
- la suite (V_n) définie par $V_0 = 0$ et $V_{n+1} = 0,5V_n + 3,5$ pour tout entier n .

On considère l'affirmation suivante : « pour tout entier n , $U_n < V_n$ ».

Camille pense que cette affirmation est vraie alors que Dominique pense le contraire.

Pour les départager, on réalise le programme ci-contre écrit en langage Python.

Le programme renvoie la valeur 11. Qui de Camille ou Dominique a raison ?

Justifier votre réponse.

Remarque : u , v et n sont les variables où sont stockées respectivement les valeurs successives des termes U_n , des termes V_n et de l'entier n .

```
def algo( ) :  
    n = 0                  ( $n = 0$ )  
    u = -4                 ( $U_0 = -4$ )  
    v = 0                  ( $V_0 = 0$ )  
    while u < v :          (On exécute la boucle tant que  $U_n < V_n$  donc la boucle s'arrête dès que  $U_n \geq V_n$ )  
        n = n+1            (On calcule l'indice suivant . Exemple :  $n = 1$ )  
        u = (8*n-4)/(n+1)  (On calcule le terme suivant de la suite  $(U_n)$ . Exemple :  $U_1$ )  
        v = 0,5*v+3,5      (On calcule le terme suivant de la suite  $(V_n)$ . Exemple :  $V_1$ )  
    return(n)
```

Réponse attendue : Le programme retourne la valeur 11 donc cela signifie que $U_{11} \geq V_{11}$ et donc c'est Dominique qui a raison.