

Ecrire un terme d'indice donné pour une suite définie par récurrence.

Exemple 1 :

Soit (u_n) la suite définie sur $\mathbb{N}$ par $\begin{cases} u_0 = 3 \\ u_{n+1} = 2u_n - 1 \end{cases}$.

- Déterminer u_1, u_2 et u_3 .
- Compléter le programme Python ci-contre pour qu'il permette de calculer u_n .
- Vérifier vos réponses à la question a) et utiliser ce programme pour calculer u_{100}

```
1 def u(n):
2     u=
3     for i in range ( ):
4         u=
5     return (u)
```

Exemple 2 :

On doit étudier une suite (v_n) définie sur $\mathbb{N}$.

Pour cela, on utilise le programme Python ci-contre pour qu'il permette de calculer les termes v_n de cette suite.

- Quelle est la valeur de v_0 ?
- Quelle est la relation de récurrence entre v_{n+1} et v_n ?
.....
- Utiliser ce programme pour calculer v_{100}
.....

```
1 def v(n):
2     v=4
3     for i in range (n):
4         v=v+1/v
5     return (v)
```

Exemple 3 :

Soit (w_n) la suite définie sur $\mathbb{N}$ par $\begin{cases} w_0 = 3 \\ w_{n+1} = \sqrt{(w_n)^2 + 1} \end{cases}$.

A l'aide d'un programme Python, calculer w_{100}

Programme Python

Pour les plus rapides :

Une autre façon de programmer l'exemple 1 pourrait être celle-ci :

En utilisant le même modèle, proposer une autre façon de programmer les exemples 2 et 3.

```
1 def u(n):
2     if n == 0:
3         return 3
4     else:
5         return 2*u(n-1)-1
```

Trouver des seuils

Exercice 4 :

En 2018, Carole verse 3 000 € sur son compte épargne. Chaque année, la somme disponible augmente de 3 %.

On souhaite écrire un algorithme qui permet d'afficher l'année à partir de laquelle Carole disposera d'au moins 3500 €.

- Augmenter une quantité de 3% revient à multiplier cette quantité par
.....
- Compléter les parties manquantes de l'algorithme.

```
S ← 3000
A ← 2018
Tant que .....
    S ← .....
    A ← A + 1
Fin Tant que
```

c) Compléter le tableau suivant :

	Initialisation								
S	3000								
A	2018								
Condition									

d) Programmer cet algorithme en langage Python en utilisant une fonction sans argument et qui renvoie l'année cherchée.

Exercice 5 :

Soit (u_n) la suite définie par $u_0 = 15$ et la relation $u_{n+1} = 3u_n + 7$ pour tout entier naturel n .

Compléter la fonction Python ci-contre afin qu'elle retourne le plus petit entier p tel que $u_p > 1000$.

Quel est le résultat renvoyé par cette fonction ?

```

1 def seuil():
2     u=15
3     p=0
4     while :
5         p=p+1
6         u=
7     return()

```

Exercice 6 :

La désintégration de l'atome de radium produit un gaz appelé radon. Ce gaz se désintègre au cours du temps en perdant 16,5 % de sa masse chaque jour.

On considère qu'au début de l'expérience, on dispose d'un gramme de radon.

On désigne par u_n la masse de radon au bout du $n^{\text{ième}}$ jour. On a donc $u_0 = 1$.

a) Expliquer pourquoi $u_{n+1} = 0,835u_n$.

b) Quel semble être le comportement de la suite lorsque n devient de plus en plus grand ?

c) On considère que la quantité de radon est négligeable lorsque celle-ci est inférieure à 10^{-6} .

Voici un programme Python qui retourne l'indice du premier terme de la suite inférieure à 10^{-6} .

Compléter les parties manquantes du programme.

d) Recopier ce programme et l'exécuter. Interpréter ce résultat.

```

1 def radon():
2     u=1
3     n=0
4     while :
5         n=
6         u=
7     return(n)

```

Calculer la somme des termes consécutifs d'une suite

Exercice 7 :

Soit (v_n) la suite définie par $v_0 = 4$ et $v_{n+1} = v_n + \frac{1}{v_n}$.

a) Compléter l'algorithme ci-contre pour que la variable S contienne la valeur de la somme $v_0 + v_1 + \dots + v_n$ en fin d'algorithme.

b) Ecrire une fonction Python, d'argument n , qui retourne S.

```

V ← 4
S ← 4
Pour i variant de 1 à ...
    | V ← .....
    | S ← .....
Fin Pour

```

Exercice 8 :

Un coureur cycliste, Ugo a programmé un entraînement hebdomadaire afin de se préparer à une course qui aura lieu dans quelques mois. Son objectif est de parcourir une distance totale de 1 500 km pendant la période d'entraînement de 20 semaines (**de la semaine 0 à la semaine 19**).

Ugo commence son entraînement en parcourant 40 km la semaine 0 et prévoit d'augmenter cette distance de 5 km par semaine.

On note u_n la distance, en kilomètres, parcourue par Ugo la semaine n .

On a ainsi $u_0 = 40$.

- a) Donner la nature de la suite (u_n) en précisant son premier terme et sa raison, ainsi que la relation de récurrence.

.....
.....
.....

- b) Compléter les pointillés dans l'algorithme ci-contre de façon à ce qu'il affiche en sortie la distance parcourue par Ugo durant la semaine n d'entraînement.

$u \leftarrow \dots$ Pour i allant de 1 à n $u \leftarrow \dots$ Fin Pour Afficher u
--

- c) Compléter le programme ci-dessous pour qu'il affiche la distance totale parcourue par Ugo à la semaine n .

```
1 def distance(n):  
2     u=40  
3     S=  
4     for i in range():  
5         u=  
6         S=  
7     return(S)
```

- d) Ugo va-t-il atteindre son objectif de 1 500 km à la semaine 19 ?

Pour les plus rapides : écrire une fonction "semaine" qui prend en argument une distance d donnée et qui renvoie le nombre de semaines d'entraînement nécessaires pour atteindre cette distance d .

Tester cette fonction en trouvant le nombre de semaines nécessaires pour atteindre une distance totale de 3000 km.

Programme Python