Algorithme : somme des termes d'une suite

Exercices corrigés PDF · Première



Le cours en vidéo

et les exercices en version interactive, corrigés pas à pas.

Vidéo et exercices

Exercice 1

66 % de réussite

Corrigé interactif

Soit (V_n) la suite définie par $V_0 = 3$ et, pour tout entier naturel n, $V_{n+1} = -2V_n - 2$.

1) Que renvoie la fonction Python suivante ? Justifier.

```
def terme():
    V = 3
    for i in range(1, 10):
        V = -2*V - 2
    return V
```

1) Compléter la fonction suivante pour qu'elle renvoie la somme des 10 premiers termes $V_0 + V_1 + \cdots + V_9$.

```
def somme():
    V = 3
    S = 3
    for i in range(1, 10):
        V = -2*V - 2
        S = ……
    return ……
```

Exercice 2

84 % de réussite

Corrigé interactif

Soit (U_n) la suite définie par $U_0 = 3$ et, pour tout entier naturel n, $U_{n+1} = \dfrac{-1}{U_n + 7}$.

1) Que renvoie la fonction Python suivante ? Justifier.

```
def terme():
    U = 3
    for i in range(1, 20):
        U = -1/(U + 7)
    return U
```

1) Compléter la fonction suivante pour qu'elle renvoie la somme des 20 premiers termes $U_0 + U_1 + \cdots + U_{19}$.

```
def somme():
    U = 3
    S = 3
    for i in range(1, 20):
        U = -1/(U + 7)
        S = ......
    return ......
```

Exercice 3

87 % de réussite

Corrigé interactif

Soit (W_n) la suite définie par $W_0 = 3$ et, pour tout entier naturel n, $W_{n+1} = -2W_n^2 - 2$.

1) Que renvoie la fonction Python suivante ? Justifier.

```
def terme():
    W = 3
    for i in range(1, 12):
        W = -2*W**2 - 2
    return W
```

1) Compléter la fonction suivante pour qu'elle renvoie la somme des 12 premiers termes $W_0 + W_1 + \cdots + W_{11}$.

```
def somme():
    W = 3
    S = 3
    for i in range(1, 12):
        W = -2*W**2 - 2
        S = ......
    return ......
```

Exercice 1

1) La boucle s'exécute 9 fois (i de 1 à 9) : partant de V_0, elle calcule successivement $V_1, \ldots, V_9$.
La fonction renvoie $\boxed{V_9}$, le 10e terme de la suite.

2) À chaque passage, on ajoute le nouveau terme à S, qui contient déjà V_0 :

```
def somme():
    V = 3
    S = 3
    for i in range(1, 10):
        V = -2*V - 2
        S = S + V
    return S
```

Exercice 2

1) La boucle s'exécute 19 fois (i de 1 à 19) : partant de U_0, elle calcule successivement $U_1, \ldots, U_{19}$.
La fonction renvoie $\boxed{U_{19}}$, le 20e terme de la suite.

2) À chaque passage, on ajoute le nouveau terme à S, qui contient déjà U_0 :

```
def somme():
    U = 3
    S = 3
    for i in range(1, 20):
        U = -1/(U + 7)
        S = S + U
    return S
```

Exercice 3

1) La boucle s'exécute 11 fois (i de 1 à 11) : partant de W_0, elle calcule successivement $W_1, \ldots, W_{11}$.
La fonction renvoie $\boxed{W_{11}}$, le 12e terme de la suite.

2) À chaque passage, on ajoute le nouveau terme à S, qui contient déjà W_0 :

```
def somme():
    W = 3
    S = 3
    for i in range(1, 12):
        W = -2*W**2 - 2
        S = S + W
    return S
```

Algorithme suite, somme des termes : exercices corrigés PDF (Première). Ces exercices d'algorithme sur les suites, niveau première, apprennent à lire une boucle qui calcule un terme puis à écrire une fonction somme. Chaque algorithme de suite est corrigé ligne par ligne, en Python. Cours en vidéo et exercices interactifs corrigés sur galilee.ac.