



1. Calcul des premiers termes d'une suite :

Une suite arithmétique est une suite dans laquelle chaque terme est obtenu à l'aide du précédent auquel nous ajoutons une constante (la raison).

Première méthode : liste des premiers termes

Pour calculer les premiers termes d'une suite arithmétique, on peut alors définir une fonction comme suit :

```
def suite_arithmetique(terme , raison , indice_final):
    U = [ terme ]          #initialisation de la liste de stockage
    for n in range(indice_final):
        terme += raison    #on pourrait mettre terme = raison + terme
        U.append( terme )  #ajout de terme dans la liste U

    return U
```

Point Python :

[1, 4, 5] est une liste pour Python. On peut y stocker ce que l'on veut : des nombres des chaînes de caractères ou même d'autres listes.

Pour ajouter un élément par exemple 7 à une liste qui s'appelle liste, on fait : liste.append(12).

Attention au programme si le premier terme est u_1 ...

Essayer le programme en tapant : `suite_arithmetique(3,5,20)` dans la console.

À vous de jouer :

Créer une fonction `suite_géo` qui calculera les premiers termes d'une suite géométrique.

2. Afficher le n-ième terme avec une formule par récurrence :

Dans une suite arithmétique, chaque terme est obtenu en ajoutant la raison r au terme précédent.

La forme récurrente de la suite arithmétique est : $u_{n+1} = u_n + r$

```
# Calcul du n-ième terme d'une suite arithmétique par récurrence
def terme_suite_arithmetique_reccurrence(U0, r, n):
    terme = U0
    for i in range(1, n + 1):
        terme = terme + r
    return terme
```

Calculer u_{150} pour une suite arithmétique de raison -7 et de premier terme $u_0 = 6$.

À vous de jouer :

Créer une fonction `terme_suite_geometrique_recurrence` qui calculera le n-ième terme d'une suite géométrique.

3. Somme des termes d'une suite

Calcul de la somme des n premiers termes d'une suite arithmétique par itération

```
def somme_suite_arithmetique(U0, r, n):  
    somme = 0  
    terme = U0  
    for i in range(n):  
        somme = somme + terme  
        terme = terme + r  
    return somme
```

Appliquez le programme pour :

`U0 = 2` # Premier terme

`r = 4` # Raison

`n = 8` # Nombre de termes à sommer

À vous de jouer :

Créer une fonction `somme_suite_geometrique` qui calculera la somme des n premiers termes d'une suite géométrique d'une suite géométrique.

4. Calcul de seuil

Soit la suite définie comme suit :

$$\begin{cases} u_{n+1} = 0,9u_n + 1,3 \\ u_1 = 3 \end{cases}$$

On définit une fonction qui permet de calculer l'indice du terme de la suite qui dépassera la valeur p:

```
def seuil(p) :  
    n=1  
    u=3  
    while u<=p :  
        n=n+1  
        u=0.9*u+1.3  
    return n
```

Déterminer la valeur renvoyée par la saisie de `seuil(8.5)`.

À vous de jouer (comme au bac):

a. En tenant compte des contraintes du milieu naturel dans lequel évoluent les insectes, les biologistes choisissent une nouvelle modélisation. Ils modélisent le nombre d'insectes à l'aide de la suite (v_n) , définie par :

$v_0 = 0,1$ et, pour tout entier naturel n , $v_{n+1} = 1,6v_n - 1,6v_n^2$, où, pour tout entier naturel n , v_n est le nombre d'insectes, exprimé en millions, au bout de n mois.

On donne ci-contre la fonction seuil, écrite en langage Python.

a. Qu'observe-t-on si on saisit seuil(0.4) ?

b. Déterminer la valeur renvoyée par la saisie de seuil(0.35).

Interpréter cette valeur dans le contexte de l'exercice.

```
def seuil(a) :  
    v=0.1  
    n=0  
    while v<a:  
        v=1.6*v-1.6*v*v  
        n=n+1  
    return n
```

b. Pour tout entier naturel $n > 1$, on a :

$$I_n = \frac{1 + e^{-n\pi}}{n^2 + 1}$$

On souhaite obtenir le rang n à partir duquel la suite (I_n) devient inférieure à 0,1. Recopier et compléter la cinquième ligne du script Python ci-dessous avec la commande appropriée.

```
1  from math import *  
2  def seuil() :  
3      n = 0  
4      I = 2  
5      ...  
6          n=n+1  
7          I=(1+exp(-n*pi))/(n*n+1)  
8  return n
```

c. On considère la suite (u_n) définie par : $\begin{cases} u_0 = 0,5 \\ u_{n+1} = 2u_n - u_n^2 \end{cases}$.

Recopier et compléter le script Python ci-dessous afin que celui-ci renvoie le rang n à partir duquel la suite dépasse 0,95.

```
def seuil() :  
  
    n=0  
  
    u=0.5  
  
    while u < 0.95 :  
  
        n=...  
  
        u=...  
  
    return n
```

d. Pour tout entier naturel n , on définit la suite (T_n) par : $\begin{cases} T_0 = -19 \\ T_{n+1} = 0,94T_n + 1,5 \end{cases}$

Le programme suivant, écrit en langage Python, doit renvoyer après son exécution la plus petite valeur de l'entier n pour laquelle $T_n > 10$.

Recopier ce programme sur la copie et compléter les lignes incomplètes afin que le programme renvoie la valeur attendue.

```
def seuil() :  
    n=0  
    T= .....  
    while T .....  
        T= .....  
        n=n+1  
    return
```

Bonus :

a. Compléter le programme suivant pour qu'il donne les solutions d'une équation du second degré de la forme $ax^2 + bx + c = 0$.

```
from math import * #importation de la biblio math pour la racine  
def resolution(a, b, c):  
    delta = ...  
    if delta > 0 :  
        x1 = (-b-sqrt(delta))/(2*a) #sqrt pour racine carrée  
        x2 = ...  
        print("Les solution sont : ",x1, "et ", x2)  
    elif ... == 0 :  
        x0 = ...  
        print(...)  
    else :  
        print(...)
```

b. Faire un programme qui calcul le produit scalaire de 2 vecteurs dont on donne les coordonnées et qui dit si ceux-ci sont orthogonaux.

Bon courage pour le bac de Français et merci pour cette année avec vous....