

Exercice 1 Soit (u_n) la suite définie par
$$\begin{cases} u_1 &= 1 \\ u_0 &= 0 \\ u_{n+1} &= nu_n + u_{n-1} \end{cases}.$$

Ecrire une fonction en Python qui prend comme paramètre une valeur N à l'utilisateur et calcule U_N .

Exercice 2 Soit $(u_n, v_n, w_n) \in (\mathbb{R}^{\mathbb{N}})^3$ un triplet de suite définie par

$$\begin{cases} u_0 = 1, v_0 = 2, w_0 = 3 \\ u_{n+1} = u_n + v_n + w_n \\ v_{n+1} = 2u_n + 2v_n + w_n \\ w_{n+1} = u_n + 2v_n + 3w_n \end{cases}$$

Ecrire une fonction qui renvoie le plus petit indice n tel que $u_n > 10^5$.

On n'utilisera aucune variable temporaire, ni d'affectation multiple.

Correction de l'exercice 1 Il est important de bien écrire sous la forme d'une suite ou d'un système récurrent d'ordre 1 pour ne pas se tromper dans la programmation.

La relation de récurrence est d'ordre 2, il faudra donc deux variables u et v pour le calcul de la suite.

On pose (v_n) la suite (u_{n+1}) et on a :

$$\begin{cases} u_0 = 0 \\ v_0 = 0 \\ u_{n+1} = v_n \\ v_{n+1} = u_{n+2} = (n+1)u_{n+1} + u_n = (n+1)v_n + u_n \end{cases}$$

```
def suite(N):
    v=1
    u=0
    n=0
    while n<N-1:
        vtemp=v
        v=(n+1)*v+u
        u=vtemp
        n=n+1
    return v
```

On a pu prendre comme condition de fin $n < N - 1$, car comme $v_n = u_{n+1}$, $v_{N-1} = u_N$

Correction de l'exercice 2 Soit $(u_n, v_n, w_n) \in \mathbb{R}^{\mathbb{N}^3}$ un triplet de suite définie par

$$\begin{cases} u_0 = 1, v_0 = 2, w_0 = 3 \\ u_{n+1} = u_n + v_n + w_n \\ v_{n+1} = 2u_n + v_n + 2w_n \\ w_{n+1} = u_n + 2v_n + 3w_n \end{cases} \Leftrightarrow \begin{cases} u_0 = 1, v_0 = 2, w_0 = 3 \\ u_{n+1} = u_n + v_n + w_n \\ -u_{n+1} + w_{n+1} = 2w_n + v_n \\ -2u_{n+1} + v_{n+1} = -v_n \end{cases} \Leftrightarrow \begin{cases} u_0 = 1, v_0 = 2, w_0 = 3 \\ u_{n+1} = u_n + v_n + w_n \\ w_{n+1} = u_{n+1} + 2w_n + v_n \\ v_{n+1} = 2u_{n+1} - v_n \end{cases}$$

Il y a une subtilité dans ce pivot de Gauss que vous n'avez pas forcément compris lors de la première séance. L'important dans ce pivot est le choix de l'ordre (U,V,W) qui est à la fois sur les inconnues des variables et sur le second membre. Ainsi échanger deux ligne correspond aussi à changer l'ordre des inconnues (échanger deux colonnes).

On obtient la fonction suivante :

```
def suite():  
    u=1  
    v=2  
    w=3  
    n=0  
    while u<=1e5:  
        u=u+w+v  
        w=u+2w+v  
        v=u-v  
        n=n+1  
    return n
```