

Ex 1 : On donne les 2 programmes ci-dessous :

```
# CODE PYTHON
def suite(N):
    '''In : indice N, entier naturel
    Out : ? '''
    assert N >= 0
    u = 500
    n = 0
    while n <= N:
        n = n + 1
        u = 0.9 * u + 1
    return n, u
```

Pseudo Code

Fonction suite(N)

assert $N \geq 0$

$u \leftarrow 500$

$n \leftarrow 0$

Tant que $n \leq N$ Faire

$n \leftarrow n + 1$

$u \leftarrow 0.9 \times u + 1$

Fin Tant que

Renvoyer n, u

- 1) Définir la suite récurrente (u_n) utilisée dans ces 2 programmes
- 2) Dresser le tableau d'avancement dans le cas $N = 5$
- 3) On remarque que le programme 1 ne renvoie pas le terme de rang 5 mais le suivant, modifier-le pour corriger ce problème.
- 4) Conjecturer la limite quand n tend vers $+\infty$ de la suite (u_n) à l'aide de vos algorithmes en calculant les termes d'indice 10, 20, 50, 100 par exemple.

Ex 2 : Un couple fait un placement au taux annuel de 2% dont les intérêts sont capitalisés tous les ans. Le couple a placé le montant de 1 000 euros à l'ouverture le 1er janvier 2010 puis, tous les ans à chaque 1er janvier, verse 2 400 euros. On note u_n le capital présent sur le compte le 1er janvier 2010 + n après le versement annuel. On donne les 2 programmes ci-dessous :

```
# CODE PYTHON
def u(N):
    '''In : indice N, entier naturel
    Out : u(N)'''
    assert N >= 0
    u, n = 1000, 0
    while n < N:
        n = n + 1
        u = ...
    return u
```

Pseudo Code

fonction u(N)

assert $N \geq 0$

$u, n \leftarrow 1000, 0$

Tant que $n < N$ Faire

$n \leftarrow n + 1$

$u \leftarrow \dots$

Fin Tant que

Renvoyer u

- 1) Définir la suite récurrente (u_n) utilisée dans ces 2 programmes
- 2) Expliquer le rôle de ces 2 programmes
- 3) L'objectif du couple est de constituer un capital de 18 000 euros. On cherche à écrire un algorithme qui renvoie l'indice et la valeur du premier terme de la suite qui dépasse un 18 000. Compléter et tester le programme en déterminant l'année durant laquelle le couple aura atteint son objectif.

Code Python

```
def recherche(seuil):
    '''renvoie le terme et l'indice de la suite qui dépasse le
    seuil.'''
    assert seuil >= 1000
    u = 1000
    n = 0
    while ..... : # à compléter
        n = n + 1
        u = ..... # à compléter
    return (n, u)
```

Pseudo Code

Fonction recherche(seuil)

assert $seuil \geq 0$

$u \leftarrow 1000$

$n \leftarrow 0$

Tant que Faire

$n \leftarrow n + 1$

$u \leftarrow \dots$

Fin Tant que

Renvoyer (n, u)

Ex 3 : On considère la suite (v_n) définie par $v_{n+1} = \frac{v_n - 2}{3v_n - 4}$

On donne les 2 programmes ci-dessous :

Code Python

```
def suite2(N, v0):
    assert N >= 0
    v = v0
    for i in range(....):
        ...
    return v
```

Pseudo Code

Fonction suite2(N, v0)

assert $N \geq 0$

$v \leftarrow v_0$

Pour i variant de ... à ... Faire

...

Fin Pour

renvoyer v

- 1) Compléter ces 2 programmes permettant de calculer le terme d'indice N de la suite (v_n) de premier terme v_0
- 2) Calculer la limite de cette suite (v_n) et écrire un programme "de seuil" permettant d'approcher la valeur de cette limite

Ex 4 : Soient les suites (u_n) et (v_n) définies par $u_{n+1} = 0,8u_n + 45$; $u_0 = 150$ et $v_n = u_n - 225$

- 1) Calculer les 5 premiers termes des suites (u_n) et (v_n)
- 2) Écrire un programme permettant de renvoyer le terme U_N et V_N pour un certain rang N donné
- 3) Conjecturer alors la limite des suites (u_n) et (v_n)

```
def f(n):
    '''In : indice n, entier naturel
    Out : (n, u_n, v_n)'''
    assert n >= 0
    ...
    ...
    return (n, u, v)
```

- 4) Montrer que pour tout n , $u_n = 225 - 75 \times 0,8^n$
- 5) Vérifier la limite ℓ de la suite (u_n)