

**Entraînement 1**

Écrire une fonction `suite(n, f, a)` qui prend en paramètres un entier `n`, un flottant `a` et une fonction `f` et renvoie le terme de rang `n` de la suite définie par :

$$\begin{cases} u_0 = a \\ u_{n+1} = f(n, u_n) \end{cases}$$

*Solution :*

```

1 def suite(n, f, a):
2     """renvoie le terme de rang n de la suite récurrente définie par
3     u_0 = a et u_{n+1} = f(n, u_n) """
4
5     u = a
6     for k in range(n):
7         u = f(k, u)
8     return u

```

Remarque : On peut justifier la correction du code en montrant par récurrence sur `k` la propriété : « la variable `u` contient le terme de rang `k` de la suite avant l'exécution du corps de la boucle lorsque le compteur de boucle vaut `k` ».

**Entraînement 2**

Écrire une fonction qui prend en paramètre un entier `n` et renvoie le terme de rang `un` de la suite.

$$(u_n) : \begin{cases} u_0 = 1, u_1 = 2, u_2 = 3 \\ u_{n+3} = u_n u_{n+1} + n^2 u_{n+2} \end{cases}$$

*Solution :*

```

1 def suite_E2(n):
2     (a, b, c) = (1, 2, 3)
3     for k in range(n):
4         (a, b, c) = (b, c, a*b+k**2*c)
5     return a

```

**Entraînement 3**

Écrire une fonction qui prend en paramètres un entier `n` et deux flottants `x` et `y` et calcule les termes `an` et `bn` des suites définies par :

$$\begin{cases} a_0 = x, b_0 = y, \\ a_{n+1} = \frac{a_n + b_n}{2} \\ b_{n+1} = \sqrt{a_n b_n} \end{cases}$$

*Solution :*

```

1 def suite_E3(n, x, y):
2     (a, b) = (x, y)
3     for k in range(n):
4         (a, b) = ((a + b)/2, sqrt(a * b))
5     return a, b

```