

## I) Codage d'algorithmes

### A) Exemple du cours

Voici ce que donne le premier exemple du cours, codé en Python.

```
k=2
n=5
res=1
while n>0 :
    res=res*k
    n=n-1
print('Le résultat est '+str(res))
```

#### Exercice 1 (introduction)

Le recopier et le faire tourner sur votre machine. Modifier les valeurs initiales, observer l'explorateur de variables. L'exécuter en mode « debug » (avec les boutons bleus ou **F6**).

### B) Exponentiation rapide

#### Exercice 2 (une approche de la complexité)

On donne l'algorithme ci-dessous.

```
k ← 2
n ← 5
res ← 1
Tant que n > 0 faire
    Si n est divisible par 2 Alors
        k ← k × k
        n ← n/2
    Sinon
        res ← res × k
        n ← n - 1
Afficher res
```

Le coder en Python. Comprendre ce qu'il fait pas à pas.

Comparer son efficacité avec l'algorithme de l'exercice 1 (de manière empirique).

### C) Dichotomie

#### Exercice 3

Soit  $f$  la fonction définie sur  $\mathbb{R}_+^*$  par  $f(x) = 1 + \frac{\ln x}{x}$ .

On donne l'algorithme ci-dessous.

Variables :	$a, b$ et $m$ sont des nombres réels.				
Initialisation :	Affecter à $a$ la valeur 0,1. Affecter à $b$ la valeur 1.				
Traitement :	Tant que $b - a > 0,01$ <table style="margin-left: 20px; border-left: 1px solid black; border-right: 1px solid black; border-collapse: collapse;"> <tr> <td style="padding: 0 10px;">Affecter à <math>m</math> la valeur <math>\frac{1}{2}(a + b)</math>.</td> </tr> <tr> <td style="padding: 0 10px;">Si <math>f(m) &lt; 0</math> alors Affecter à <math>a</math> la valeur <math>m</math>.</td> </tr> <tr> <td style="padding: 0 10px;">Sinon Affecter à <math>b</math> la valeur <math>m</math>.</td> </tr> <tr> <td style="padding: 0 10px;">Fin de Si.</td> </tr> </table>	Affecter à $m$ la valeur $\frac{1}{2}(a + b)$ .	Si $f(m) < 0$ alors Affecter à $a$ la valeur $m$ .	Sinon Affecter à $b$ la valeur $m$ .	Fin de Si.
Affecter à $m$ la valeur $\frac{1}{2}(a + b)$ .					
Si $f(m) < 0$ alors Affecter à $a$ la valeur $m$ .					
Sinon Affecter à $b$ la valeur $m$ .					
Fin de Si.					
	Fin de Tant que.				
Sortie :	Afficher $a$ . Afficher $b$ .				

Dans l'éditeur, commencer votre programme par `from math import log` et utiliser `log` pour calculer le `ln`.

Coder cet algorithme en Python.

Cet algorithme s'appelle la *dichotomie*. Il est à connaître. La fonction doit être continue et monotone sur l'intervalle concerné pour qu'on soit sûr que la boucle se termine et donne le résultat voulu.

## II) Construire des algorithmes

### A) Détecter un carré

#### Exercice 4

On se donne un nombre entier  $n$ . Détecter si  $n$  est un carré à l'aide d'une boucle `while`.

### B) Plus avancé

#### Exercice 5 (décomposition en base $N$ )

- 1) Écrire un algorithme qui décompose en base 2 un nombre entier positif, puis affiche le résultat. On pourra stocker le résultat dans une chaîne de caractères.
- 2) Généraliser l'algorithme à une base  $N \in \llbracket 2, 10 \rrbracket$ .
- 3) Pour  $N > 10$ , il nous faut  $N$  symboles distincts, donc les chiffres  $0, \dots, 9$  ne sont plus suffisants. Généralement, on rajoute des lettres. Pour  $N = 16$ , par exemple, les symboles sont les suivants :

Valeur de $a_i$ en décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Symbole codant $a_i$ en base 16	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f

Reprendre le cours pour comprendre les notations :  $n = \sum_{i=0}^k a_i N^i = a_0 + a_1 N + \dots + a_k N^k$ .

Construire un algorithme décomposant un entier en base 16.

#### Exercice 6

Il y a 5 façons différentes d'obtenir un total de 8 en additionnant le résultat de deux dés indexés de 1 à 6 (en comptant différemment les couples (3, 5) et (5, 3)).

- 1) De combien de façon différentes peut-on obtenir un total de 12 avec trois dés ?
- 2) Même question avec  $d$  dés pour un total de  $n$ .