Introduction à la Programmation en Python

1 Analyse et Simulation de programmes

Exercice 1. *Arithmétique* COURS

Voici deux algorithmes $\mathcal{E}$ et $\mathcal{P}$.

Algorithme $\mathcal{E}$

ENTRÉES: a et b deux entiers

tant que $a \neq b$ **faire**
  si $a \geq b$ **alors**
    $a \leftarrow a - b$
  sinon
    $b \leftarrow b - a$
  fin si
fin tant que
renvoyer a

Algorithme $\mathcal{P}$

ENTRÉES: n un entier

$p \leftarrow$ **VRAI**
$d \leftarrow 2$
tant que $d^2 \leq n$ **faire**
  si $n \mod d = 0$ **alors**
    $p \leftarrow$ **FAUX**
  fin si
  $d \leftarrow d + 1$
fin pour
renvoyer p

- Essayer de deviner leurs spécification de sortie.
- Implémentez les algorithmes en Python.
- Simulez les algorithmes pas à pas.
- Vérifiez et éventuellement corrigez la spécification de sortie.

2 Premiers programmes

Exercice 2. *Méthode de Halley pour le calcul de* $\sqrt{2}$

Implémentez en Python la fonction `halley` qui calcule le $n^{\text{ème}}$ terme de la suite récurrente suivante :

$$H_0 = 1\,,\ H_{n+1} = H_n \cdot \frac{(H_n^2 + 6)}{(3\,H_n^2 + 2)}$$

Exercice 3. *Logarithme entier*

On rappelle que pour tout $n \geq 1$ et $b \geq 2$, le logarithme entier en base b de n est l'unique entier l tel que

$$b^l \leq n < b^{l+1}$$

Implémentez en Python la fonction ayant les spécifications suivantes :

Algorithme `logarithme_entier`

ENTRÉES: $n \geq 1$ et $b \geq 2$ deux entiers

SORTIES/EFFETS: un entier. Le logarithme entier en base b de n

Exercice 4. *Suite de Fibonacci* COURS

Implémentez en Python la fonction `fibonacci` qui calcule le $n^{\text{ème}}$ terme de la suite récurrente suivante :

$$F_0 = 0\,,\ F_1 = 1\,,\ F_{n+2} = F_{n+1} + F_n$$

Exercice 5. *Syracuse* COURS

Les mathématiques ne sont pas encore prêtes pour de tels problèmes – Paul Erdõs

Une suite de Syracuse est définie par la relation de récurrence suivante :

$$S_{n+1} = \begin{cases} \dfrac{S_n}{2} & \text{si } S_n \text{ est pair,} \\ 3S_n + 1 & \text{si } S_n \text{ est impair.} \end{cases}$$

La conjecture de Syracuse stipule que, quelque soit $S_0 \in \mathbb{N}^*$, la suite de Syracuse débutant par S_0 atteint 1. On notera qu'une fois la valeur 1 atteinte, la suite boucle sur les valeurs $1, 4, 2, 1, \ldots$.

Implémentez une fonction `syracuse` qui prend en entrée un entier strictement positif s_0 et qui renvoie le premier rang auquel la suite atteint la valeur 1.

Est-ce que la fonction `syracuse` termine sur toutes les entrées ?

Exercice 6. *Nombre parfait*

Un nombre parfait est un nombre entier positif égal à la somme de ses diviseurs stricts (lui-même exclu).

Implémentez une fonction `parfait` qui prend en entrée un entier et renvoie vrai s'il est parfait.

3 Pour s'entraîner

Exercice 7. *Suites adjacentes*

Implémentez en Python la fonction `adjacentes` qui prend en argument un flottant ϵ et qui calcule le premier rang n termes des suites adjacentes suivantes, à partir duquel $|a_n - b_n| < \epsilon$:

$$b_{n+1} = \frac{a_n + b_n}{2}\,,\ a_{n+1} = \frac{2}{b_{n+1}}\,,\ a_0 = 1, b_0 = 2$$

Exercice 8. *Méthode de Héron*

La suite x^a définie si dessous converge vers $\sqrt{a}$.

$$\forall n \in \mathbb{N} \quad x_{n+1}^a = \frac{x_n^a + \frac{a}{x_n^a}}{2}\ ,\ x_0^a = a$$

Implémentez une fonction `heron` qui prend en argument deux entiers a et n positifs et renvoie x_n^a.

Trouvez le rang à partir duquel x_n^2 approche $\sqrt{2}$ à 10^{-10} prêt (≈ 1.4142135623).

Exercice 9. *Nombres Chanceux d'Euler*

Un nombre chanceux d'Euler est un nombre $n > 1$ tel que pour tout $0 \leq i \leq n-2$, $i^2 + i + n$ est premier.

Implémentez une fonction `chanceux_euler` qui prend en entrée un entier n et renvoie vrai ssi n est chanceux.

4 Pour aller plus loin

Exercice 10. *Fonction 91 de McCarthy*

La fonction 91 de McCarthy est définie pour $n \in \mathbb{N}$ par la formule suivante :

$$f(n) = \begin{cases} n - 10 & \text{si } n > 100 \\ f(f(n+11)) & \text{sinon.} \end{cases}$$

Implémentez une fonction `mccarthy` qui prend en entrée un nombre entiers n positifs et renvoie $f(n)$. Observer le retour de la fonction pour les valeurs plus petites ou égales à 101.