CPES Scientifique

Informatique

TP 5 (TP 4 bis)

Fonctions et algorithmes avec boucles conditionnelles

Résumé

Ce TP poursuit et complète le TP4. Il peut être commencé avant d'avoir abordé les deux dernières questions du TP4.

1 Suites etc

Question 1.1 (Suites arithmétiques). *Étant donnés u_0 et b, on considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par la relation*

$$u_{n+1} \quad = \quad u_n + b$$

(a) Donner deux méthodes pour calculer le terme u_n en fonction de u_0, b et n.

(b) Pour chacune de ces deux méthodes, écrire une fonction Python qui prend en arguments u_0, b et n et qui renvoie le terme u_n.

(c) Tester vos fonctions sur des suites de votre choix.

Question 1.2. *Étant donnés u_0 et b, on considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par la relation*

$$u_{n+1} \quad = \quad u_n + b + n$$

(a) Écrire une fonction Python qui prend en arguments u_0, b et n et qui renvoie le terme u_n.

(b) Tester votre fonction sur des suites de votre choix.

Question 1.3. *Étant donnés u_0, k, p, on considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par la relation*

$$u_{n+1} \quad = \quad k \cdot u_n - p \cdot n$$

(a) Écrire une fonction Python prennant en arguments u_0, k, p et n et renvoyant la valeur du terme u_n.

(b) Écrire une fonction Python prennant en arguments u_0, k et p et renvoyant le rang du dernier terme strictement positif de la suite $(u_n)_{n\in\mathbb{N}}$ s'il existe.

On pourra tester ces fonctions avec la suite

$$u_{n+1} \quad = \quad \frac{1}{2}u_n - 3n$$

Question 1.4 (Suites arithmético-géométriques). *Étant donnés u_0, a, b, on considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par la relation*

$$u_{n+1} \quad = \quad a \cdot u_n + b$$

On rappelle que lorsque $a \neq 1$, on a

$$u_n \quad = \quad a^n(u_0 - r) + r \qquad \text{avec} \quad r = \frac{b}{1-a} \tag{1}$$

(a) *Écrire une fonction Python prennant en arguments u_0, a, b et n et renvoyant le terme u_n, et procédant itérativement, sans utiliser la relation (1).*

(b) *On souhaite maintenant utiliser la relation (1) (dans le cas $a \neq 1$). Quelle(s) opération(s) arithmétiques peut-on utiliser pour calculer r ?*

(c) *Écrire une fonction Python implémentant la relation (1) et comparer ses résultats avec ceux de la fonction définie en (a). On pourra se baser sur les suites*

$$\begin{array}{rrcll} & u_{n+1} & = & 3 \cdot u_n + 1 & \text{avec} \quad u_0 = 1 \\ \text{et} & u_{n+1} & = & 2^{54} \cdot u_n + 1 & \text{avec} \quad u_0 = 1 \end{array}$$

Question 1.5 (Suite de Fibonacci)**.** *On rappelle que la suite de Fibonacci est la suite*

$$0 \quad 1 \quad 1 \quad \underbrace{2}_{=1+1} \quad \underbrace{3}_{=1+2} \quad \underbrace{5}_{=2+3} \quad \ldots$$

(a) *Donner la formule de récurrence de la suite de Fibonacci.*

(b) *Écrire et tester une fonction Python prennant en argument un entier n et renvoyant le n-ième terme de la suite de Fibonacci.*

(c) *Écrire et tester une fonction Python prennant en argument un entier n et renvoyant le premier terme $\geq n$ de la suite de Fibonacci.*

2 Un peu d'arithmétique

Question 2.1 (PGCD par la méthode d'Euclide)**.** *On rappelle la méthode d'Euclide pour calculer le* **plus grand diviseur commun** (pgcd) *de deux entiers $a, b \in \mathbb{N}$, avec $a \geq b > 0$:*

- *Soit r le reste de la division euclidienne de a par b.*
  - *Si $r = 0$ alors $\text{pgcd}(a, b) = b$,*
  - *sinon $\text{pgcd}(a, b) = \text{pgcd}(b, r)$.*

Écrire une fonction Python qui prend en arguments deux entiers positifs a et b et qui renvoie leur pgcd.

Question 2.2 (Calcul du coefficient de poids fort dans la décomposition base k)**.** *Soit k un entier ≥ 2. La décomposition base k d'un entier positif n est la suite*

$$a_m, a_{m-1}, \ldots, a_0$$

telle que

- $n = \sum_{i=0}^{m} a_i k^i$,
- $0 \leq a_i < k$ *pour tout* i,
- $a_m > 0$ *si* $n > 0$, *et* $a_m = m = 0$ *si* $n = 0$.

(a) *Écrire et tester une fonction Python prennant en arguments n et k et renvoyant $m + 1$ (c'est-à-dire le nombre de termes de la suite $a_m, a_{m-1}, \ldots, a_0$).*

(b) *Écrire et tester une fonction Python prennant en arguments n et k et renvoyant a_m.*