TP 1 : Suites et récurrence

Pour ce TP, nous allons coder des fonctions sous Python. Vous pourrez utiliser l'interpréteur Python de votre choix.

Exercice 1 Soit $(u_n)_{n\in\mathbb{N}}$ la suite arithmétique de premier terme $u_0 = a$ et de raison $a + b$, où a est la somme des chiffres de votre jour de naissance et b est la somme des chiffres de votre mois de naissance. Par exemple, pour moi qui suis né un 6 janvier, $a = 6$ et $b = 1$.
a) Quelle est la relation entre u_{n+1} et u_n ?
b) Donner l'expression générale de u_n en fonction de n. En déduire la valeur de u_{10}.

Exercice 2 Soit $(u_n)_{n\in\mathbb{N}}$ la suite géométrique de premier terme $u_0 = a$ et de raison $a + b$.
a) Quelle est la relation entre u_{n+1} et u_n ?
b) Donner l'expression générale de u_n en fonction de n. En déduire la valeur de u_5.

Définition Dans n'importe quel langage de programmation, une fonction est dite **récursive** si elle s'appelle elle-même.

Par exemple, voici une fonction Python qui calcule $x^n = x \times x^{n-1}$ de façon récursive.

```
def puissance (x,n):
   if n>0:
      return x * puissance (x,n-1)
   else: return 1

print puissance (2,32)
```

Vous pouvez trouver plus d' informations sur :

`http://igm.univ-mlv.fr/~rispal/W3bis/src/python/cours2/CM_5.pdf`

Exercice 3 Soit $(u_n)_{n\in\mathbb{N}}$ la suite arithmético-géométrique telle que

$$\forall n \in \mathbb{N}, \quad u_{n+1} = au_n + (a + b)$$

avec $u_0 = 0$.
a) Programmer une fonction Python récursive qui prend en entrée la valeur de n et renvoie la valeur de u_n.
b) Donner la valeur de u_5, puis de u_{10}.
c) On introduit la suite $(v_n)_{n\in\mathbb{N}}$ avec $v_n = u_n + c$. Trouver la valeur de c telle que la suite (v_n) soit géométrique.
d) En déduire l'expression de u_n en fonction de n et retrouver les valeurs de u_5 et u_{10}.

Exercice 4 Soit $(u_n)_{n\in\mathbb{N}}$ la suite définie par

$$\forall n \in \mathbb{N}, \quad u_{n+2} = au_n + (a + b)u_{n+1}$$

avec $u_0 = 0$ et $u_1 = 1$.
a) Programmer une fonction Python doublement récursive qui prend en entrée la valeur de n et renvoie la valeur de u_n.
b) Donner la valeur de u_5, puis de u_{10}.

Exercice 5 Pour calculer une approximation de la racine carrée d'un nombre positif x, une façon de faire, appelée méthode de Héron d'Alexandrie, est de calculer les termes de la suite définie par

$$u_{n+1} = 1/2(u_n + x/u_n)$$

avec $u_0 = 1$. Cette suite est positive et se rapproche très rapidement de sa limite ℓ.
a) Montrer, en utilisant le théorème du point fixe, que $\ell = \sqrt{x}$.
b) Programmer une fonction Python récursive qui prend en entrée la valeur de x et la valeur de n et renvoie la valeur de u_n. La tester pour des valeurs de $\sqrt{x}$ connues.
c) Comment savoir jusqu'à quelle valeur de n aller pour obtenir une bonne approximation de $\sqrt{x}$? On peut décider de s'arrêter lorsque la variation relative entre u_{n+1} et u_n, donnée par

$$\frac{|u_{n+1} - u_n|}{u_n},$$

est inférieure à un certain seuil. Programmer une fonction Python qui prend en entrée la valeur de x et renvoie la suite (u_n) obtenue jusqu'à ce que la variation relative soit inférieure à 10^{-3}.
d) Soit x le nombre correspondant à votre date de naissance (par exemple 6011979 pour moi qui suis né le 06/01/1979). Tracer la suite (u_n) correspondant à cette valeur de x en utilisant la bibliothèque matplotlib.pyplot. En utilisant la fonction math.sqrt, calculer l'erreur relative commise lorque l'on utilise l'approximation de $\sqrt{x}$ précédente (le dernier u_n calculé).

Exercice 6 Soit $(u_n)_{n\in\mathbb{N}}$ et $(v_n)_{n\in\mathbb{N}}$ deux suites définies par les relations de récurrence suivantes :

$$\begin{cases} u_{n+1} &= 0,7 \times (-1)^a u_n + 0,4 \times (-1)^b v_n \\ v_{n+1} &= 0,3 \times (-1)^{b+1} u_n + 0,6 \times (-1)^{a+1} v_n \end{cases}$$

a) Soit le vecteur $X_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix}$. Montrer que l'on peut écrire $X_{n+1} = AX_n$, où A est une matrice à préciser.
b) Calculer $A^2 = AA$. En déduire l'expression de u_2 et de v_2 lorsque $u_0 = 2$ et $v_0 = 3$.
c) Écrire une fonction Python récursive qui prend en entrée u_0, v_0 et n et renvoie les valeurs de u_n et v_n. Vérifier les résultats de la question b).
d) Modifier la fonction Python pour qu'elle renvoie l'ensemble des valeurs des deux suites jusqu'à l'ordre n : $(u_0, u_1, \cdots, u_n)$ et $(v_0, v_1, \cdots, v_n)$.
e) Représenter sur un même graphique les deux suites jusqu'à $n = 100$.