Suites et programmation

EXERCICE 1

Suites récurrentes

1) Soit la suite définie par $u_0 = 0,2$ et pour $n \in \mathbb{N}$, $u_{n+1} = 0,5u_n + 0,3$
   a) Calculer u_1, u_2 et u_3
   b) Écrire un programme en langage naturel permettant de calculer u_n, n étant donné.
   c) Traduire ce programme en Python avec la fonction $u(n)$. Tester cette fonction avec $n = 3$, $n = 10$, $n = 50$.
   Que peut-on conjecturer sur la convergence de la suite (u_n) ?
   d) Écrire un programme en Python permettant de déterminer le plus petit $n \in \mathbb{N}$ tel que $|u_n - 0,6| < 10^{-6}$

2) Soit la suite (u_n) définie par : $\begin{cases} u_0 = 0 \ , \ u_1 = 1 \\ u_{n+2} = 5u_n + 3u_n \end{cases}$
   a) Calculer u_2, u_3 et u_4.
   b) Écrire un programme en langage naturel permettant de calculer u_n, n étant donné.
   c) Traduire ce programme en Python avec la fonction $u(n)$. Tester cette fonction avec $n = 4$, $n = 8$, $n = 15$.

EXERCICE 2

Suites définie par une somme

1) Soit la suite (u_n) définie par $u_0 = 1$ et pour $n \in \mathbb{N}^*$, $u_n = 1 \times 2 \times 3 \times \cdots \times n$
   a) Calculer u_1, u_2, u_3 et u_4.
   b) Écrire un programme en langage naturel permettant de calculer u_n, n étant donné.
   c) Traduire ce programme en Python avec la fonction $u(n)$. Tester cette fonction avec $n = 4$, $n = 8$, $n = 12$.

2) Soit la suite (u_n) définie pour $n \in \mathbb{N}^*$ par, $u_n = 1 + \dfrac{1}{2} + \dfrac{1}{3} + \cdots + \dfrac{1}{n}$.
   a) Calculer en fraction irréductible : u_1, u_2 et u_3.
   b) Écrire un programme en langage naturel permettant de calculer u_n, n étant donné.
   c) Traduire ce programme en Python avec la fonction $u(n)$. Tester cette fonction avec $n = 3$, $n = 10$, $n = 50$ et $n = 100$.
   d) Que peut-on conjecturer sur la convergence de la suite (u_n).

EXERCICE 3

Suite de Syracuse

Soit la suite (u_n) définie par u_0 et pour $n \in \mathbb{N}$, $\begin{cases} u_{n+1} = \dfrac{1}{2}u_n & \text{si } u_n \text{ est pair} \\ u_{n+1} = 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$

On s'intéresse au nombre de termes nécessaire pour que $u_n = 1$.

1) a) On prend $u_0 = 6$. Déterminer les premiers termes jusqu'à trouver un terme égal à 1.

   b) On prend $u_0 = 7$. Déterminer les premiers termes jusqu'à trouver un terme égal à 1.

2) Écrire un programme en langage naturel permettant de donner la liste et le nombre de termes nécessaires pour obtenir un terme égal à 1.

3) a) Traduire ce programme en Python avec la fonction $u(n)$ qui renvoie la liste, le nombre de termes nécessaires pour obtenir un terme égal à 1 et le nombre le plus grand de la liste.

   On donne les instructions python suivantes :

   - $a\%n$: reste de la division de a par n.
   - L.append(a) : on rajoute l'élément a à la fin de la liste L
   - max(L) : détermine le nombre le plus grand de la liste L

   b) Remplir le tableau suivant :

n	Nombre de termes	Terme maximum
23		
24		
41		
53		
57		

   Que peut-on en conclure ?

Pour la petite histoire : Dès 1928, Lothar Collatz s'intéressait aux itérations dans les nombres entiers. Il inventa alors le problème $3x + 1$, et le présentait souvent ensuite dans ses séminaires.
En 1952, lors d'une visite à Hambourg, Collatz expliqua son problème à Helmut Hasse. Ce dernier le diffusa en Amérique à l'université de Syracuse : la suite de Collatz prit alors le nom de "*suite de Syracuse*". Entre temps, le mathématicien polonais Stanislas Ulam le répand dans le Laboratoire national de Los Alamos. Dans les années 1960, le problème est repris par le mathématicien Shizuo Kakutani qui le diffuse dans les universités Yale et Chicago.
Cette conjecture mobilisa tant les mathématiciens durant les années 1960, en pleine guerre froide, qu'une plaisanterie courut selon laquelle ce problème faisait partie d'un complot soviétique visant à ralentir la recherche américaine.