TP n°17

Suites récurrentes

Exercice 1

Soit $(u_n)_{n\in\mathbb{N}}$ la suite définie par la donnée de $u_0 \in \mathbb{R}$ et la relation de récurrence :

$$u_{n+1} = 2u_n - 1$$

valable pour tout $n \in \mathbb{N}$.

1. En utilisant la relation de récurrence précédente, construire une fonction nommée `terme_suite_ag` d'arguments :
   - `first_term` , une expression de type `float` ;
   - `n` , une expression de type `int` (représentant un entier naturel) ;

   qui retourne :
   - le terme d'indice `n` de la suite $(u_n)_{n\in\mathbb{N}}$ dont le premier terme est u_0 :=`first_term`.

   Voici quelques exemples d'appels de la fonction `terme_suite_ag`.

```
>>> terme_suite_ag(0,2)
-3
>>> terme_suite_ag(1/2,10)
-511.0
>>> terme_suite_ag(2,50)
1125899906842625
```

2. Constuire une variante de la fonction `terme_suite_ag` , nommée `terme_suite_ag_bis` , en utilisant, cette fois-ci, le résultat du cours de mathématiques sur l'expression explicite d'un terme de suite arithmético-géométrique.
3. Laquelle des deux fonctions `terme_suite_ag` , `terme_suite_ag_bis` est la meilleure ? On expliquera ce que l'on entend par « meilleure » dans la réponse proposée.

Exercice 2

Soit $(u_n)_{n\in\mathbb{N}}$ la suite définie par la donnée de $u_0 \in \mathbb{R}$, $u_1 \in \mathbb{R}$ et la relation de récurrence :

$$u_{n+2} = 2u_{n+1} - u_n$$

valable pour tout $n \in \mathbb{N}$.

1. En utilisant la relation de récurrence précédente, construire une fonction nommée `terme_suite_rlo2` d'arguments :
   - `first_term` , une expression de type `float` ;
   - `second_term` , une expression de type `float` ;
   - `n` , une expression de type `int` (représentant un entier naturel) ;

   qui retourne :
   - le terme d'indice `n` de la suite $(u_n)_{n\in\mathbb{N}}$ dont le premier terme est u_0 :=`first_term` et le second est u_1 :=`second_term`.

   Voici quelques exemples d'appels de la fonction `terme_suite_rlo2`.

```
>>> terme_suite_rlo2(1,2,100)
101
>>> terme_suite_rlo2(2,-5,1000)
-6998
>>> terme_suite_rlo2(10,59,10000)
490010
```

2. Constuire une variante de la fonction `terme_suite_rlo2` , nommée `terme_suite_rlo2_bis` , en utilisant, cette fois-ci, le résultat du cours de mathématiques sur l'expression explicite d'un terme de suite récurrente linéaire d'ordre 2.

Exercice 3
Une suite de Syracuse est définie par la donnée de $u_0 \in \mathbb{N}^*$ et la relation de récurrence :

$$u_{n+1} = \left| \begin{array}{ll} \dfrac{u_n}{2} & \text{si } u_n \text{ est pair} \\ & \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{array} \right.$$

valable pour tout $n \in \mathbb{N}$.

1. Construire une fonction nommée `suite_Syracuse` d'arguments :
   - `first_term` , une expression de type `int` (représentant un entier naturel non nul) ;
   - `n` , une expression de type `int` (représentant un entier naturel non nul) ;

   qui retourne :
   - la liste des `n` premiers termes de la suite $(u_n)_{n\in\mathbb{N}}$ dont le premier terme est u_0 :=`first_term`.

   Voici quelques exemples d'appels de la fonction `suite_Syracuse`.

   ```
   >>> suite_Syracuse(7,10)
   [7, 22, 11, 34, 17, 52, 26, 13, 40, 20]
   >>> suite_Syracuse(11,10)
   [11, 34, 17, 52, 26, 13, 40, 20, 10, 5]
   >>> suite_Syracuse(131,20)
   [131, 394, 197, 592, 296, 148, 74, 37, 112, 56, 28, 14, 7, 22, 11, 34, 17, 52, 26, 13]
   ```

2. À l'aide de la fonction `suite_Syracuse`, écrire un programme qui affiche la liste des 50 premiers termes des suites de Syracuse dont le premier terme u_0 appartient à $[\![1,20]\!]$.
3. Énoncer alors une conjecture sur les suites de Syracuse.
4. Vérifier la conjecture précédente pour les suites de Syracuse dont le premier terme u_0 appartient à $[\![1,100000]\!]$.