.

NOM	.....
PRENOM	.....

Exercices

Exercices section 7.

Dans cette section, il est demandé de programmer des itérations de calcul de termes de suites à l'aide de boucles `for`. L'usage des bibliothèques `math` ou `numpy` est alors requis.

Suites récurrentes

Le but du TP est de voir comment, progressivement, passer du calcul d'un terme particulier à l'établissement d'un script général de traitement de nombreuses suites numériques. Ilm est recommandé de traiter chaque exercice complètement dans le même script (fichier).

1. *Cas des suites arithmético-géométriques*

   On rappelle que, pour une suite arithmético-géométrique donnée $(u_n)_{n\in\mathbb{N}}$ vérifiant $u_{n+1} = qu_n + r$ pour tout $n \in \mathbb{N}$ avec $q \neq 1$:

   $$\forall n \in \mathbb{N} \quad u_n = \left(u_0 - \frac{r}{1-q}\right) q^n + \frac{r}{1-q}$$

   on gardera à l'esprit la possibilité de poser $\alpha = \dfrac{r}{1-q}$.

   (a) Calculer le $12^{\text{ième}}$ terme de la suite arithmético-géométrique $(u_n)_{n\in\mathbb{N}}$ de premier terme $u_0 = 5$ et vérifiant $u_{n+1} = 2u_n - 3$ à l'aide d'une boucle `for` en Python (utiliser l'éditeur de script !) et en exploitant la formule de récurrence :

   ```
   from math import*
   u=..........................
   for ........................
           u = ....................
   print(.....................)
   ```

   Comparer le résultat avec celui obtenu au moyen de commandes :

   ```
   >>>
   >>>
   >>>
   .
   ```

   Des remarques ?

   ```
   ...
   ...
   ```

   (b) Créer une fonction qui prend n en entrée et revoie le terme u_n de la suite précédente :

```
def suitu(n):
...
...
...
...
...
...
```

Créer alors la liste des termes de u_{20} à u_{25} au moyen de votre programme.

```
import numpy as np
Lu=np.array([0,0,0,0,0,0])
for ..........................
        Lu[...]=..................
print("Lu=",Lu)
```

Puis fournissez vos résultats :

```
...
...
...
```

(c) Le but de cette question est de comparer les méthodes dites *itératives* et *explicites*

Commençons par la méthode itérative en programmant une fonction dont les paramètres d'entrée seront alors n, q, r et a et qui renvoie le terme u_n de la suite géométrique définie par :

$$u_0 = a \quad \text{et} \quad \forall n \in \mathbb{N} \ \ u_{n+1} = qu_n + r$$

```
def ITarigeo(a,q,r,n):
        u=a
        for ......................
                ......................
        return(..................)
```

On poursuit avec l'usage direct de la formule explicite :

```
def EXarigeo(a,q,r,n):
        u=a
        for ......................
                ......................
        return(..................)
```

Tester compartivement vos deux procédures pour générer la liste des 15 premiers termes de la suite définie par :

$$u_0 = -1 \quad \text{et} \quad u_{n+1} = 1,32u_n + 1$$

```
...
...
...
...
...
...
```

(un tableau synthétique est fortement recommandé)

Des remarques ?

```
...
...
```

2. *Avec une situation de récurrence un plus générale :* On se donne une suite récurrente $(v_n)_{n\in\mathbb{N}}$ définie par $v_0 = a \in \mathbb{R}$ et $v_{n+1} = f(v_n)$ où :

$$f(x) = \frac{2x-1}{1+x^2}$$

(a) On donne $a = 1$ dans cette question. Proposer un calcul de v_1 directement à la console :

```
>>>
...
```

Et pour v_2 ?

```
>>>
...
```

(b) On garde $a = 1$. Compléter le script pour que la fonction `vfa` calcule le terme v_n lorsque n est donné en entrée :

```
def vfa(n):
        v=1
        for ..........................
...
...
...
...
...
```

Et on n'oublie pas de faire quelques essais :

```
...
...
...
```

(c) Généraliser cette approche en écrivant un script Python qui prend a et n en entrées et renvoie le terme v_n de cette suite.

```
def suiteV(a,n):
...
...
...
...
...
...
```

Utiliser ensuite ce programme pour générer une liste des 10 premiers termes de la suite $(v_n)_{n\in\mathbb{N}}$ pour : $a = 0$ puis $a = 2$ et enfin $a = -3$.

```
...
...
...
```

(seuls les résultats sont attendus)

(d) Pour chacun des trois cas précédents, conjecturer le sens de variations de la suite $(v_n)_{n\in\mathbb{N}}$.

```
...
...
...
```

(e) *Et si on devait changer la fonction f ?* Dans le but de généraliser l'approche à toute suite $v_{n+1} = f(v_n)$, nous pourrions définir f dans un bloc à part...

Proposer un script en deux blocs permettant de réaliser cette tâche dans le cas de la fonction $f(x) = \dfrac{2x-1}{1+x^2}$ fournie :

```
def frec(x):
        y=.............................
        return(y)
```

```
def VsuitF(a,n):
...
...
...
...
...
...
```

3. Finalement, comment feriez-vous pour programmer de façon assez générale les calculs des termes d'une suite définie par récurrence au moyen d'une relation de type $u_{n+1} = f(u_n)$ et dont le terme initial est $u_0 = a$ (un réel) ?

On n'attend pas de code particulier pour cette question

```
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...
```