### TP 3

# Boucle FOR – suites récurrentes

## I. Suites récurrentes d'ordre 1

## a. Application du cours

Exercice 1 Écrire une fonction terme(n) qui renvoie la valeur de  $u_n$ , où :

1. 
$$u_1 = 6$$
 et  $\forall n \ge 2$ ,  $u_n = (u_{n-1} + n)^2$ 

2. 
$$u_0 = 3$$
 et  $\forall n \in \mathbb{N}, u_{n+1} = \frac{u_n + 3n}{2n + 1}$ 

#### b. Conjectures

**Exercice 2** Soit la suite u définie par  $u_0 = 4$ , et  $\forall n \in \mathbb{N}$ ,  $u_{n+1} = \frac{u_n}{1 + u_n^2}$ .

- 1. Écrire une fonction suite(n) qui calcule et renvoie la valeur de  $u_n$ .
- 2. Utiliser cette fonction suite afin d'émettre une conjecture sur la limite de la suite  $(u_n)$ .
- 3. Écrire une fonction monotonie (n) qui teste la monotonie de la suite u, c'est-à-dire :
  - renvoie False si il existe  $k \in [0, n]$  tel que  $u_{k+1} u_k$  change de signe,
  - renvoie True sinon.
- 4. Comment utiliser cette fonction monotonie pour émettre une conjecture sur la monotonie de la suite  $(u_n)$ ?
- 5. Si la suite est supposée monotone, comment utiliser la fonction suite pour préciser la monotonie ?

**Exercice 3** On considère la suite u définie par  $u_0 = -1$ , et  $\forall n \in \mathbb{N}$ ,  $u_{n+1} = u_n - \frac{1 + u_n + 2e^{u_n}}{1 + 2e^{u_n}}$ .

Écrire une fonction suite(n) qui calcule et renvoie la valeur de  $u_n$ , pour  $n \ge 1$ .

A l'aide de cette fonction, établir une démarche informatique permettant d'émettre des conjectures sur la limite de la suite  $(u_n)$  et sa monotonie.

on pourra s'inspirer de l'exercice précédent.

## b. Sommes

**Exercice 4** On considère la suite u définie par :  $u_0 = 1$  et pour tout entier naturel n,  $u_{n+1} = 3\sqrt{u_n} + 1$ 

- 1. Écrire une fonction suite(n) qui calcule et renvoie la valeur de  $u_n$
- 2. On note  $S_n = \sum_{k=0}^n u_k$ 
  - (a) Écrire une fonction somme\_1(n) qui calcule et renvoie la valeur de  $S_n$ , en utilisant la fonction suite
  - (b) Écrire une fonction somme\_2(n) qui calcule et renvoie la valeur de  $S_n$ , sans utiliser la fonction suite
- 3. En Python, on peut chronométrer le temps que met un calcul avec la fonction time de la bibliothèque time de la façon suivante

```
import time
T0=time.time() stocke dans T0 l'heure de début de calcul
# écrire le calcul à effectuer
T1=time.time() stocke dans T0 l'heure de fin de calcul
print(T1-T0) affiche la durée du calcul
```

Proposer un script qui permet de constater que somme\_2 est plus rapide que somme\_1 pour n = 100 ou n = 1000.

# II. Suites récurrentes d'ordre 2

## a. Application du cours

Exercice 5 Écrire une fonction terme(n) qui renvoie la valeur de  $u_n$ , où :

1. 
$$u_0 = 1$$
,  $u_1 = 2$  et  $\forall n \in \mathbb{N}$ ,  $u_{n+2} = 2u_{n+1} - 3u_n$ 

2. 
$$u_0 = 2$$
,  $u_1 = 3$  et  $\forall n \ge 2$ ,  $u_{n+2} = \sqrt{u_n + u_{n+1}^2}$ 

## b. Conjectures

**Exercice 6** On considère la suite de Fibonacci  $(F_n)_{n\in\mathbb{N}}$  définie par:

$$F_0 = 0$$
  $F_1 = 1$  et  $\forall n \ge 2$ ,  $F_n = F_{n-1} + F_{n-2}$ .

Écrire une fonction qui calcule et affiche  $F_n$ , pour tout entier n. En déduire une conjecture sur la limite éventuelle de cette suite.