

Le but de ce TP est de réviser la notion de récursivité vue en première année puis de l'illustrer sur des questions traitées par une stratégie de programmation dynamique.

Exercice 1 (sommés d'entiers)

Écrire la fonction récursive `s_entiers(n)` (pour $n \in \mathbb{N}$) qui renvoie la somme des entiers de 1 à n .

Exercice 2 (suites numériques)

1. Définir en python la suite v donnée par :

$$v_0 = \sqrt{2} \quad \text{et} \quad \forall n \in \mathbb{N}, v_{n+1} = \sqrt{v_n + 2}.$$

2. Définir en python la suite w donnée par :

$$w_0 = 0 \quad \text{et} \quad \forall n \in \mathbb{N}, \begin{cases} w_{2n} = w_n \\ w_{2n+1} = 1 - w_n. \end{cases}$$

Pourquoi la suite w est-elle bien définie?

Exercice 3 (un système de suites numériques)

Soit a et b deux réels. On définit les suites u et v de la façon suivante :

$$u_0 = a, v_0 = b \quad \text{et} \quad \forall n \in \mathbb{N}, \begin{cases} u_{n+1} = \frac{1}{2}(u_n + v_n) \\ v_{n+1} = \sqrt{u_n v_n} \end{cases}$$

1. Écrire un programme définissant ces deux suites de façon récursive.
2. Quel défaut présente la programmation récursive de ces deux suites?
3. Écrire un programme définissant ces deux suites de façon itérative.
4. Comparer la rapidité de ces deux programmes (par exemple en important le module `time` et en utilisant `time.time()`).

Exercice 4 (exponentiation)

Écrire la fonction récursive `exp_rec(x, n)` qui renvoie x^n en utilisant :

$$x^0 = 1$$

$$\forall x \in \mathbb{N}^*, x^n = x \cdot x^{n-1}$$

On mettra en place une série de tests avec l'instruction `assert`.

