

TP n°7

Fonctions – partie 1

Consignes

- Tous les programmes seront documentés.
- Pour chacune des fonctions construites, on effectuera une batterie de tests de manière à couvrir les différents cas possibles, en particulier les cas « limite ».

Exercice 1 (Fonction définie par morceaux)

Soit f la fonction définie par

$$\left| \begin{array}{l} f : \mathbb{R} \rightarrow \mathbb{R} \\ x \mapsto \begin{cases} 3 & \text{si } x < 1 \\ 2x + 1 & \text{si } x \geq 1. \end{cases} \end{array} \right.$$

1. Écrire une fonction `f`
 - d'argument x de type `float` ;
 - qui retourne l'image de x par f .
2. Écrire une fonction `integrale_f`
 - d'arguments (a, b) un couple d'expressions de type `float` ;
 - qui retourne la valeur de l'intégrale $\int_a^b f(x) dx$.

On utilisera l'interprétation géométrique de l'intégrale (cf. aire sous une courbe) et on prendra en compte l'ordre des bornes.

Exercice 2 (Somme des éléments d'une liste)

Écrire une fonction `somme_liste`

- d'argument L une liste d'expressions de type `int` ;
- qui retourne la somme des entiers figurant dans la liste L .

Par convention, la somme des éléments de la liste vide est 0.

Exercice 3 (Médiane d'une liste de réels)

Écrire une fonction `mediane`

- d'argument L une liste non vide d'expressions de type `float` ;
- qui retourne la médiane de la liste L .

On pourra utiliser la méthode `sort()` du type `list`. La documentation de cette méthode peut être obtenue en saisissant

```
help(list)
```

dans la console¹.

Exercice 4 (Répétitions éventuelles dans une liste)

Écrire une fonction `repetition`

- d'argument L une liste ;
- qui retourne `True` si la liste contient des répétitions et `False` sinon.

1. Les tris de listes seront l'un des thèmes du programme de Spé.