

# Informatique : devoir n°3 (non surveillé)

Vous pouvez rendre le DM sur papier ou m'envoyer vos programmes par mail. Dans ce cas tous les programmes doivent être écrits dans un unique fichier nommé `DM3_votrenom.py`. Les noms des variables doivent être bien choisis. On peut utiliser `#` pour mettre des commentaires. Les DM d'informatique n'étant pas notés, l'utilisation de ChatGPT ne vous ramènera strictement rien.

## Exercice 1

- 1) Écrire une fonction `valeur_absolue` qui, recevant un nombre (entier ou flottant), renvoie sa valeur absolue. On n'utilisera pas la fonction `abs`.
- 2) Écrire une fonction `plus_proche` qui, recevant trois nombres, renvoie celui des deux derniers qui est le plus proche du premier. Par exemple, `plus_proche(10, 13, 8)` doit renvoyer 8.

## Exercice 2 - Moyenne, variance, médiane

Soient  $x_1, \dots, x_n$  des réels.

- 1) La moyenne  $m$  et la variance  $V$  de la famille  $(x_1, \dots, x_n)$  sont définies par :

$$m = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{et} \quad V = \frac{1}{n} \sum_{i=1}^n (x_i - m)^2.$$

Écrire deux fonctions `moyenne` et `variance` qui, recevant une liste de nombres, renvoient respectivement sa moyenne et sa variance.

```
>>> moyenne([12, 7, 11, 15])
11.25
>>> variance([12, 7, 11, 15])
8.1875
```

- 2) Si la famille  $(x_1, \dots, x_n)$  est croissante, sa médiane est  $x_{p+1}$  si  $n = 2p + 1$  et  $(x_p + x_{p+1})/2$  si  $n = 2p$  (où  $p \in \mathbb{N}$ ). Par exemple, la médiane de la famille  $(1, 2, 3, 4, 5)$  est 3 et celle de la famille  $(1, 2, 3, 4, 5, 6)$  est  $(3 + 4)/2 = 3.5$ . Si la famille  $(x_1, \dots, x_n)$  n'est pas croissante, on commence par la mettre dans l'ordre.

Écrire une fonction `mediane` qui, recevant une liste de nombres, renvoie sa médiane. On peut utiliser `sorted` pour trier la liste.

```
>>> mediane([12, 7, 11, 15, 18])
12
>>> mediane([12, 7, 11, 15])
11.5
```

## Exercice 3 - La suite de Conway

- 1) Écrire une fonction `suivant` qui, recevant une liste d'entiers, renvoie la liste obtenue en lisant “à voix haute”. Par exemple, si `L = [1, 2, 3, 3, 1]`, on lit “un 1, un 2, deux 3, un 1”, donc `suivant(L)` est `[1, 1, 2, 3, 3, 1]`.

```
>>> suivant([1, 2, 3, 3, 1])
[1, 1, 2, 3, 3, 1]
```

Il est conseillé de tester la fonction sur de nombreux exemples différents pour vérifier qu'elle renvoie le résultat attendu.

- 2) La suite de Conway est la suite de premier terme  $C_1 = 1$  et dont chaque terme s'obtient en lisant à voix haute le terme précédent. On a ainsi  $C_1 = 1$ ,  $C_2 = 11$  (un 1),  $C_3 = 21$  (deux 1),  $C_4 = 1211$  (un 2, un 1), etc.

- a) Écrire une fonction `conway` qui, recevant un entier naturel  $n$ , renvoie la liste des chiffres de  $C_n$ .

```
>>> conway(5)
[1, 1, 1, 2, 2, 1]
>>> conway(6)
[3, 1, 2, 2, 1, 1]
```

- b) Écrire une fonction `conway_p_chiffres` recevant un entier naturel  $p$  et renvoyant le premier indice  $n$  pour lequel  $C_n$  est formé de plus de  $p$  chiffres.

```
>>> conway_p_chiffres(1000000)
51
```