

Exercice 1.

Soit le tableau `mesNotes` = [10, 14, 8, 16, 15] représentant les notes obtenues par un élève au cours du trimestre.

1. Trier le tableau par ordre croissant.

```
...  
print(mesNotes) → [8, 10, 14, 15, 16]
```

2. Trier le tableau par ordre décroissant.

```
...  
print(mesNotes) → [16, 15, 14, 10, 8]
```

3. Écrire la fonction `maMoy` qui reçoit un tableau de notes en paramètre et qui renvoie la moyenne des notes.

```
maMoy(mesNotes) → 12.6
```

Exercice 2.

Soit le tableau `mesNotesCoef` = [[10,2], [14,1], [8,2], [16,3], [15,2]] représentant les notes obtenues par un élève au cours du trimestre avec un coefficient pour chacune.

Dans l'exemple, la note 10 a pour coefficient 2, 14 a pour coefficient 1...

Écrire la fonction `maMoyC` qui reçoit un tableau de notes en paramètre et qui renvoie la moyenne des notes.

```
maMoyC(mesNotesCoef) → 12.8
```

Exercice 3.

Écrire un programme qui génère par compréhension un tableau contenant tous les nombres pairs compris entre 0 et 100.

Une seule ligne de code est nécessaire pour générer ce tableau.

Exercice 4.

1. Écrire un programme qui génère par compréhension un tableau contenant 100 nombres aléatoires compris entre 0 et 100.
2. Créer un deuxième tableau par compréhension qui ne prélève dans le tableau précédent que les multiples de 3.

Exercice 5.

Soit la liste de mots :

```
monTab = ['Tim Berners-Lee', 'George Boole', 'John von Neumann', 'Alan Turing']
```

Écrire une fonction `nbCar` qui génère par compréhension un tableau contenant le nombre de caractères dans le nom complet de ces quatre informaticiens.

```
nbCar(monTab) → [15, 12, 16, 11]
```

Exercice 6.

Écrire un programme qui :

1. Crée par compréhension un tableau de 20 tableaux de trois entiers aléatoires compris entre 0 et 9.
2. Le trie en fonction du chiffre du milieu de chaque triplet.

Exercice 7.

```
monTuple = (10, 14, 12, 14)
```

1. Dissocier monTuple dans 4 variables a, b, c et d.

```
print(a) → 10  
print(b) → 14  
print(c) → 12  
print(d) → 14
```

2. Créer un tuple monTuple2 contenant les valeurs de monTuple dans l'ordre inverse.

```
print(monTuple2) → (14, 12, 14, 10)
```

3. Écrire la fonction MOY qui renvoie la moyenne des valeurs de monTuple.

```
MOY(monTuple) → 12.5
```

4. Écrire la fonction MinMax qui renvoie un tuple constitué de la valeur minimale et de la valeur maximale de monTuple.

```
MinMax(monTuple) → (10,14)
```

5. Écrire la fonction nb qui renvoie le nombre d'occurrences d'une valeur dans un tuple.

```
nb(monTuple,14) → 2  
nb(monTuple,10) → 1  
nb(monTuple,11) → 0
```

Exercice 8.

1. Générer un tuple nommé leTup constitué de l'unique valeur 5.

2. Ajouter à leTup la liste ['rouge', 'vert', 'bleu'].

```
print(leTup) → (5, ['rouge', 'vert', 'bleu'])
```

3. Extraire la couleur 'rouge' de leTup.

```
print(...) → rouge
```







Exercice 9.

1. Écrire un programme qui génère un tuple constitué de 20 entiers aléatoires compris entre 0 et 10.

2. Compter ensuite dans le tuple ainsi constitué le nombre de fois où le chiffre 5 apparaît.

Exercice 10.

Données

chr(9856) =  ; chr(9857) =  ; chr(9858) =  ; chr(9859) =  ; chr(9860) =  ; chr(9861) = .

1. Générer un tuple constitué de 100 tuples de trois entiers aléatoires compris entre 1 et 6.

2. Afficher les 10 tirages en représentant les dés.

3. Compter le nombre de fois où le tirage « 421 » est sorti (quel que soit l'ordre).

Exercice 11.

Dans la liste ci-dessous, chaque couleur est accompagnée de son codage RVB.

```
Coul = [('blanc',[255,255,255]), ('saumon',[250,128,114]), ('vert',[0,128,0]),  
('violet',[128,0,128]), ('turquoise',[64,224,208])]
```

Trier cette liste par valeur du rouge croissante.

```
→ Coul = [('vert',[0,128,0]), ('turquoise',[64,224,208]), ('violet',[128,0,128]),  
('saumon',[250,128,114]), ('blanc',[255,255,255])]
```