Exercice 1 :

Écris un algorithme qui demande à un enseignant un nombre d’élèves, qui lui propose alors de saisir des notes sur 20, qui les stocke dans un tableau.

À la fin de la saisie, le tableau des notes sera affiché ainsi que la meilleure et la moins bonne note.

Exercice 2 :

Définis par compréhension les tableaux suivants :

1. Le tableau contenant tous les nombres entiers multiples de 3 compris entre 0 et 30.
2. Le tableau contenant les éléments suivants [1,4,9,16,25,36,49]
3. On a le tableau t suivant t = [1,2,5,7,8,15,21]

On veut définir le tableau contenant les éléments impairs de t.

1. Un tableau de 12 cases dont tous les nombres sont des entiers aléatoires compris entre 1 et 10.

Exercice 3 :

Écris chacune des fonctions demandées et teste-les à l’aide d’assertions. Effectue à chaque fois tous les tests pertinents pour vérifier que la fonction est bien écrite.

1. Écris une fonction appartient(tab,*p*) qui renvoie un booléen et permet de déterminer si l’élément *p* est dans le tableau tab.
2. Écris une fonction occurrence(tab,*p*) qui renvoie la première occurrence (numéro de la case dans laquelle l’élément apparaît pour la première fois) de *p* dans le tableau tab si *p* est un élément du tableau et 1 sinon.
3. Écris une fonction apparition(tab, *p*) qui renvoie le nombre d’apparitions de *p* dans le tableau tab

Exercice 4 :

1. Écris une fonction qui calcule la somme des éléments d’un tableau et teste-la à l’aide d’assertions.
2. Écris une fonction qui calcule la moyenne des éléments d’un tableau et teste-la à l’aide d’assertions.
3. Écris une fonction qui permet de déterminer nombre d’éléments supérieurs à 10 dans un tableau et teste-la à l’aide d’assertions
4. Écris une fonction qui permet de déterminer l’élément maximal du tableau (n’utilise pas la fonction prédéfinie dans Python) et teste-la à l’aide d’assertions.

Exercice 5 :

Écris un algorithme qui remplit un tableau à deux dimensions [1..10][1..10] avec les multiples de 3 ligne par ligne en commençant par 3.

Exercice 6 :

On se donne un tableau de nombres entiers. Écris une procédure qui range les éléments de ce tableau dans un autre tableau en rangeant les nombres pairs à gauche et les nombres impairs à droite.

Exercice 7 :

Écris une fonction 𝑒𝑠𝑡𝑇𝑟𝑖𝑒 qui permet de savoir si les éléments contenus dans un tableau sont triés ou non. Cette fonction prend en paramètre un tableau et retourne un booléen.

On testera cette fonction à l’aide d’assertions.

Exercice 8 : Algorithmes de tri

Il est souvent nécessaire de trier les données d’un tableau afin de pouvoir ensuite les exploiter.

* 1. Réfléchis à un algorithme qui permet de trier les éléments d’un tableau et écris-le en langage naturel.
  2. Va sur le lien suivant et analyse les différents algorithmes de tri.

https://interstices.info/les-algorithmes-de-tri/?hlText=algorithmes+de+tris

* 1. Choisis un algorithme parmi ceux vu précédemment et programme-le en Python.