

**Exercice 0. Prise en main de Python**

Le programme suivant illustre la façon de déclarer en Python les tableaux.

En python :

```
# importation des fonctionnalités
from numpy import *
from typing import Iterable
Tableau=Iterable

# Déclaration du type TAB
TAB = Tableau[int]

# Déclaration des variables
t1:TAB=zeros(100,int);
i:int;
n:int;

n=int(input("Entrer le nombre de valeurs à saisir : "));
for i in range(0,n,1):
    t1[i]=int(input("Entrer un entier :"));
```

En python, vous devez obligatoirement appeler la fonction `zeros(taille_max, type_des_valeurs)` à chaque fois que vous créez un tableau.

Il correspond à une version de l'algorithme suivant en pseudo-langage.

```
CONST MAX = 100 ;
TYPE Tab = tableau [MAX] de Entier ;

Programme RemplirTableau
VAR t1 : TAB ;
    n, i : Entier ;
Début
    Ecrire("Entrer le nombre de valeurs à saisir :");
    Lire(n) ;
    Pour i de 0 à n-1 pas 1 faire
        Ecrire("Entrer un entier : ");
        Lire (t1[i]) ;
    Fin Pour
Fin
```

**NB : Dans la suite des exercices, vous devez obligatoirement définir les tableaux tels qu'ils sont définis ci-dessus (et ne pas utiliser de listes). Et vous n'avez pas le droit d'utiliser les fonctions prédéfinies sur les tableaux (min...)**

### Exercice 1 : Tableau d'entiers

1. Ecrire un programme qui demande à l'utilisateur le nombre d'entiers à saisir, puis demande chaque valeur en les stockant dans un tableau.
2. Compléter le programme pour afficher la plus petite des valeurs du tableau.
3. Compléter le programme pour afficher l'occurrence d'une valeur saisie par l'utilisateur (nombre de fois où cette valeur apparaît dans le tableau). Attention vous devrez donner la possibilité à l'utilisateur de recommencer le point n°3 jusqu'à ce qu'il refuse de continuer.

### Exercice 2 : Gestion des notes

1. Écrire un programme qui va demander à l'utilisateur le nombre de notes à saisir, puis faire la saisie des notes dans un tableau pouvant contenir au maximum 400 notes. Enfin, le programme affichera les notes saisies pour vérification.
2. Compléter le programme pour calculer et afficher ensuite la moyenne du groupe.
3. Compléter le programme pour afficher ensuite le nombre d'étudiants ayant eu 10 au DS (en utilisant une boucle tant que).

### Exercice 3 : Tableau de caractères

1. Écrire un programme qui saisit des caractères et les place dans un tableau pouvant contenir au maximum 100 caractères. L'utilisateur utilisera le '.' pour terminer sa suite de caractères.

NB : En python vous déclarerez un tableau de chaîne de caractères (str).

2. Ajouter les instructions permettant de compter le nombre de voyelles et le nombre de consonnes de la chaîne de caractères saisie par l'utilisateur puis afficher les résultats.
3. Compléter le programme pour afficher si la chaîne de caractères saisie est un palindrome ou non (par exemple, « radar » est un palindrome).
4. Enfin compléter le programme pour créer un nouveau tableau à partir du précédent en ne gardant que les consonnes et l'afficher.

### Exercice 4 : Crible d'Eratosthène

Le crible d'Eratosthène est un procédé qui permet de trouver tous les nombre premiers inférieurs à un certain entier naturel donné  $M$ . L'algorithme procède par élimination : il s'agit de supprimer d'une table des entiers de 2 à  $N$  tous les multiples d'un entier (autres que lui-même).

- Mettre les nombres entiers de 2 à  $M$  ( $M$  étant saisi par l'utilisateur) dans un tableau à une dimension.
- Rayer (dans le programme, on conviendra que rayer signifie mettre à 0) ensuite dans le tableau tous les multiples du premier nombre non rayé et supérieur au précédent.
- Une fois la méthode finie, afficher les nombres premiers (nombres non rayés du tableau).