

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

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

# Déclaration du type TAB
Tab=Tableau[int];

def saisie(n:int, t:Tab):
    i:int;
    for i in range (0,n,1):
        t[i]=int(input("Entrer une valeur : "));

def Element(p:int, t:Tab)-> int:
    return t[p];

# programme principal
nb:int;
p:int;
t:Tab=zeros(100, int);
print("Entrez le nombre de valeurs du tableau : ");
nb = int(input());
saisie(nb,t);
print("Entrez la position de l'élément à afficher : ");
p=int(input());
print(Element(p,t));
```

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

En python

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

Procédure Saisie (Entree: n : Entier, Sortie: t : Tab)
VAR i : Entier;
Début
    pour i de 0 a n-1 pas 1 faire
        Lire (t[i]);
    Fin pour
Fin

Fonction Element (Entree p : Entier, t : Tab) : Entier
Début
    Retourner(t[p]);
Fin

Programme Test
VAR nb, p : Entier;
t : Tab;
Début
    Ecrire("Entrez le nombre de valeurs du tableau : ");
    Lire(nb);
    Saisie(nb, t) ;
    Ecrire("Entrez la position de l'élément à afficher : ");
    Lire(p) ;
    Ecrire(Element(p, t)) ;
Fin
```

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

**Exercice 1 : Nombres premiers (Sans utilisation de tableaux)**

1. Ecrire une fonction booléenne qui retourne si un nombre passé en paramètre est premier.
2. Dans un programme principal, tester avec le nombre 11 et le nombre 21.
3. Compléter le programme principal, pour demander à l'utilisateur un nombre et afficher s'il est premier ou non et répéter les instructions jusqu'à ce que l'utilisateur ne veuille plus tester de nombres premiers.
4. Compléter le programme principal pour afficher tous les nombres premiers inférieur à 100.

**Exercice 2 : Tableau d'entiers**

1. Ecrire un sous-programme permettant d'afficher les n valeurs d'un tableau passé en paramètre.
2. Ecrire un sous-programme permettant de remplir les n premières cases d'un tableau de manière aléatoire. (cf TP1 pour générer un nombre aléatoirement)
3. Dans un programme principal, créer 2 tableaux différents (et les remplir à l'aide du sous-programme de la question 2) et les afficher (à l'aide du sous-programme de la question 1).
4. Ecrire un sous-programme qui détermine si un tableau donné est trié par ordre croissant.
5. Ecrire un sous-programme qui retourne la moyenne des valeurs du tableau.
6. Ecrire un sous-programme qui affiche l'occurrence d'un nombre donné en paramètre.
7. Compléter le programme principal pour tester les questions 3, 4 et 5.
8. Ecrire un sous-programme permettant de retourner la plus petite des valeurs d'un tableau. Le tester dans le programme principal.
9. Ecrire une autre sous-programme pour retourner l'indice de la case qui contient la plus petite valeur.
10. Ecrire une troisième version pour retourner l'indice de la case qui contient la plus petite valeur à partir d'une case donnée.
11. Ecrire un sous-programme qui permute le contenu de 2 cases définies par leur indice.
12. En vous servant des questions précédentes, écrire un sous-programme qui trie le tableau par ordre croissant. Vous commenterez le programme afin d'expliquer votre démarche. (Pour commenter une ligne, il suffit d'écrire # devant la ligne)
13. Dans le programme principal, trier et afficher les 2 tableaux de la question 2.