CHAPITRE REM - 3

SÉANCE 3 : ALGOS CLASSIQUES SUR LES TABLEAUX

Exercice 1 : Tableau statique en Python

1. Que réalise le programme 1 ?
2. Que signifie n en terme de valeurs stockées ? en terme de valeur à insérer ?
3. Que réalisent les programmes 2 et 3 où on ne remplace que la partie entre début ... fin ?

```
# Importation de nouvelles fonctionnalités
from random import *              # importe toutes les fonctions de random
from numpy import *               # importe toutes les fonctions de numpy
from typing import Iterable       # importe Iterable de typing

MAX = 100
Tab = Iterable[int]               # TYPE Tab = Tableau[MAX]  d'Entier

# Programme REM3
# VAR : déclaration
t : Tab = zeros(MAX, int) ;      # déclaration & initialisation à 0
i : int                          # indice
n : int                          # nombre de cases vraiment utilisées : n <= MAX

# début
n = 0 ;
t[n] = randint(0, 100) ; n = n + 1
t[n] = randint(0, 100) ; n = n + 1
t[n] = randint(0, 100) ; n = n + 1

for i in range(0, n):                        # pour i de 0 à (n-1) faire
    print(t[i])
# fin pour
# fin
```

```
# début
for i in range(0, 10):
    t[i] = randint(0,100)
    print(t[i])
n = i + 1
print()
print(n)
```

```
# début
for i in range(0, 10):
    t[i] = randint(0,100)
    print(t[i], end=' ')
n = i + 1
print()
print(n)
```

Exercice 2 : Algorithmes classiques

Compléter le programme 3 pour réaliser les actions proposées.

1. Lire les questions suivantes et réaliser l'en-tête du programme (la partie VAR notamment).
2. Continuer pour demander un nombre entre 0-100 (en précondition), stocker et afficher le nombre de valeurs strictement supérieures à cette valeur.
3. Continuer : stocker et afficher le nombre de valeurs inférieures ou égales à cette valeur.
4. Continuer : vérifier que la somme des deux nombres précédents donne bien le nomnbre de cases. Affichez 'Erreur' en cas de problème.
5. Continuer : stocker et afficher la somme des valeurs dans le tableau.
6. Continuer : stocker et afficher la moyenne.
7. Continuer : stocker et afficher le nombre de cases strictement supérieures à la moyenne.
8. Continuer : stocker et afficher la valeur max.
9. Continuer : stocker et afficher la valeur min.

Exercice 3 : Variante avec indice

On travaille avec des variantes des programmes précédents liés aux indices, pas juste aux valeurs.

Créer un programme qui stocke 12 valeurs aléatoires entre 10 et 19 dans un tableau statique de 200 place et qui ...

1. Stocke et affiche l'indice de la première valeur maximale trouvée.
2. Stocke la valeur maximale puis affiche les indices sur lequels se trouvent cette valeur.
3. Stocke dans un second tableau de 200 places les indices où se trouvent les valeurs maximales. On stockera dans une variable m le nombre d'indices qui correspondent à cette valeur maximale.

Exercice 4 : Comptage de lettres

En ascii, A correspond à 65 et Z à 90.

La fonction Python chr(nombre) renvoie le caractère correspondant à nombre (précondition : entier n dans l'intervalle 0 à x10ffff).

La fonction Python ord(caractere) renvoie le nombre correspondant au caractère.

```
>>> n = ord('B')
>>> n
66

>>> lettre = chr(67)
>>> lettre
'C'
```

1. Créer un type TL et un tableau lettres pour 4000 caractères (str en Python).
2. Créer un type TN et un tableau memoire de 91 cases d'entiers.
3. Créer un programme qui génère 10 nombres entre 65 et 90 et stocke dans un tableau de 4000 cases les 10 lettres qui correspondent.
4. Compléter le programme en comptant dans un tableau de type TN le nombre de A, de B, de C ...
5. Modifier le programme pour parvenir à stocker cela à l'aide d'un type tableau[26] d'entiers.