

# Cours: Introduction à l'algorithmique



**Pr. Hajar IGUER,**  
[hajar.iguer@uic.ac.ma](mailto:hajar.iguer@uic.ac.ma)



# **LES TABLEAUX**



# Tableaux

- L'utilité d'un tableau est d'éviter de déclarer plusieurs variables du même type.
- Un tableau nous permettra de rassembler toutes les valeurs en une seule variable et que chaque valeur sera désignée par un numéro qu'on appelle indice.



# Tableaux

- Un tableau se définit en indiquant son nom, le type des éléments stockés dans le tableau, ainsi que leur nombre, écrit entre crochets.
- Ce nombre est désigné la **taille maximale du tableau**.
- **Tableau nom\_tableau [taille\_max] : type\_variable**



# TABLEAUX

- Le premier élément d'un tableau nommé `tab` sera noté `tab(0)`
- Le  $i^{\text{ème}}+1$  élément : `tab(i)`
- Généralement on travaille avec des tableaux à une dimension mais il existe aussi des tableaux à deux dimensions qu'on notera
- **Tableau `nom_tableau [tmaxi] [tmaxj]` : `type_variable`**



# Tableaux et Boucles

- Les boucles sont extrêmement utiles pour les algorithmes associés aux tableaux.
- En effet, de nombreux algorithmes relatifs au tableau nécessitent de parcourir les éléments du tableau dans un certain ordre.
- Le traitement de chacun des éléments étant souvent le même, seule la valeur de l'indice est amenée à changer.
- Une boucle est donc parfaitement adaptée à ce genre de traitements.



# Tableaux

- L'accès à un élément du tableau se fait au moyen de l'indice. Par exemple,
  - `notes[i]` donne la valeur de l'élément  $i$  ou  $i-1$  du tableau `notes`
- Selon les langages, le premier indice du tableau est soit 0, soit 1. Le plus souvent c'est 0 (c'est ce qu'on va adopter en pseudo-code). Dans ce cas, `notes[i]` désigne l'élément  $i+1$  du tableau `notes`



# Tableaux

- Il est possible de déclarer un tableau sans préciser au départ sa dimension. Cette précision est faite ultérieurement.
- Un grand avantage des tableaux est qu'on peut traiter les données qui y sont stockées de façon simple en utilisant des boucles





# Exemple

- Pour le calcul du nombre d'étudiants ayant une note supérieure à 10 avec les tableaux, on peut écrire :

Variables i ,nbre : entier;

**tableau** notes[30] : réel;

**Début**

nbre  $\leftarrow$  0

**Pour** i de 0 à 29

**Si** (notes[i] >10) alors

        nbre  $\leftarrow$  nbre+1;

**FinSi**

**FinPour**

écrire ("le nombre de notes supérieures à 10 est : ", nbre);

**Fin**



# Tableaux : saisie et affichage

- Saisie des valeurs des éléments d'un tableau

**Pour**  $i$  de 0 à  $n-1$

écrire ("Saisie de l'élément ",  $i + 1$ );

lire ( $T[i]$  );

**FinPour**

- Affichage des valeurs des éléments d'un tableau

**Pour**  $i$  de 0 à  $n-1$

écrire ("T[, $i$ , "] =",  $T[i]$ );

**FinPour**



# Exercices

- Ecrire un algorithme qui permet de lire 100 notes et de déterminer le nombre de cellules qui sont supérieures à la moyenne.
- Soit un tableau  $T(n)$ , écrire un algorithme qui calcule VMP (valeur moyenne des éléments positifs), VMN (valeur moyenne des éléments négatifs) et NEN (nombre d'éléments nuls).



# Exercices

- Soit  $T$  un tableau de  $N$  réels. Ecrire le programme qui permet de calculer le nombre des occurrences d'un nombre  $X$ .
- Soit  $T$  un tableau de  $N$  entiers. Ecrire l'algorithme qui détermine simultanément la valeur et la position du plus grand élément et la valeur et la position du plus petit élément du tableau.



# Exercices

- Ecrire un algorithme qui insère un élément donné  $X$  à la position  $K$  d'un tableau.
- Donner un algorithme qui supprime un élément donné d'un tableau d'entiers.
- Ecrire un algorithme qui renverse un tableau ( s'il est croissant, il deviendra décroissant).

