



COLLEGE
DE PARIS
SUPÉRIEUR

DAKARTECH

ALGORITHMIQUE & PROGRAMMATIONS(C/C++)

Dr Pape Abdoulaye BARRO

Enseignant – Chercheur

Spécialiste en Télémétrie & Systèmes Intelligents

TABLEAUX

RAPPEL ALGORITHMIQUE

TABLEAUX

- ✖ Un tableau est une liste d'éléments ayant le même type, désignés sous le même nom et accessibles par indices.
- ✖ Les tableaux peuvent être d'une, deux ou de plusieurs dimensions.
 - + Pour un tableau à une dimension, la syntaxe est la suivante:

nomTableau: tableau[taille] de type

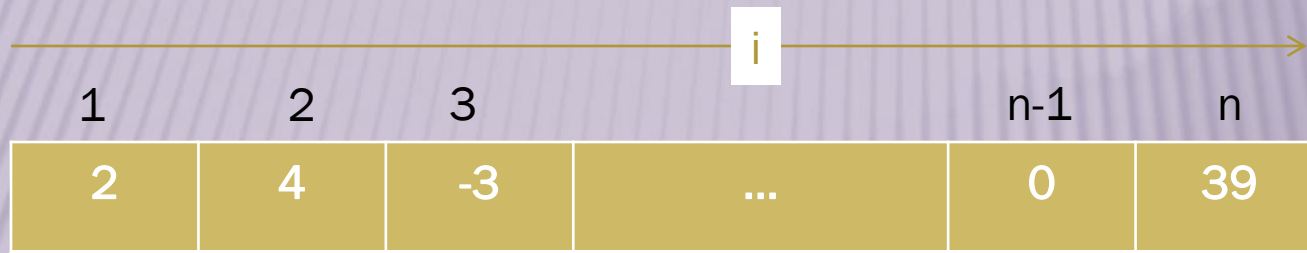
- ✖ **Exemple:**

- ★ notes: tableau[10] de réels
- ★ texte: tableau[255] de caractères

RAPPEL ALGORITHMIQUE

TABLEAUX

- ✖ Les éléments d'un tableau sont accessibles pas indice (commençant par 1) que cela soit en lecture ou en écriture.



+ Lecture

- ✖ Lire(`nomTableau[i]`)

+ Ecriture

- ✖ Ecrire(`nomTableau[i]`)

+ Affectation

- ✖ `nomTableau[i] ← Valeur`

nomTableau

RAPPEL ALGORITHMIQUE

TABLEAUX

✖ Exemple:

Écrire un algorithme permettant de saisir 10 entiers et de les stocker dans un tableau nommé tab, puis les afficher.

✖ Solution:

Algorithme tableau_dix_elements

Variable tab[10]: réels

i : entier

Début

Ecrire("Remplir le tableau")

Pour i allant de 1 à 10 faire

Lire(tab[i])

FinPour

 { **Affichage du tableau** }

Pour i allant de 1 à 10 faire

Ecrire(tab[i])

FinPour

Fin

RAPPEL ALGORITHMIQUE

TABLEAUX

+ Pour un tableau à deux dimensions, la syntaxe est la suivante:

nomTableau: tableau[ligne][colonne] de type

× **Exemple:**

- ★ notes: tableau[10][20] de réels
- ★ texte: tableau[10][255] de caractères
- ★ matrice: tableau[3][4] de réels

× **Remarque :** Nous pouvons utiliser autant de dimensions que souhaitées

RAPPEL ALGORITHMIQUE

TABLEAUX

- ✗ Les éléments d'un tableau à deux dimensions sont accessibles par indice ligne (**commençant par 1**) et colonne (**commençant par 1 aussi**) que cela soit en lecture ou en écriture.

	1	2	3	...	j	...	m-1	m
1	2	4	-3	...			0	39
2	-20	23	17	...			100	-15
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	32	54	92	...			-98	30

+ Lecture

✗ `Ecrire(nomTableau[i][j])`

+ Ecriture

✗ `Lire(nomTableau[i][j])`

+ Affectation

✗ `nomTableau[i][j] ← Valeur`

nomTableau

RAPPEL ALGORITHMIQUE

TABLEAUX

✖ Exemple:

Ecrire un algorithme qui permet de remplir une matrice M de 10 lignes et 20 colonnes. On affiche le contenu de M par la suite.

✖ Solution:

Algorithme Factorielle

Variable M[10][20]: entier

i , j , val : entiers

Début

val \leftarrow 1

{Remplissage du tableau}

Pour i allant de 1 à 10 faire

Pour j allant de 1 à 20 faire

 M[i][j] \leftarrow val

 val \leftarrow val+1

FinPour

FinPour

{Affichage du tableau}

Pour i allant de 1 à 10 faire

Pour j allant de 1 à 20 faire

 Ecrire(M[i][j])

FinPour

FinPour

Fin

RAPPEL ALGORITHMIQUE

CAS PRATIQUES N° 5

✖ Application 21:

Écrire un algorithme permettant de saisir 10 notes et qui affiche la moyenne de ces notes.

✖ Application 22:

Écrire un algorithme permettant de saisir 10 entiers et qui affiche le maximum de ces entiers.

✖ Application 23:

Écrire un algorithme permettant de saisir 10 entiers dans un tableau, et de calculer le nombre d'occurrences d'un élément N dans ce tableau. Où N saisi par l'utilisateur.

✖ Application 24:

Ecrire un algorithme qui calcule la somme des éléments d'une matrice.

✖ Application 25:

Ecrire un algorithme qui calcule la somme des lignes d'une matrice.

Affaires à suivre

