



COLLEGE  
DE PARIS  
SUPÉRIEUR  

---

DAKARTECH

## PROGRAMMATIONS C

Dr Pape Abdoulaye BARRO

Enseignant – Chercheur

Spécialiste en Télémétrie & Systèmes Intelligents

# TABLEAUX STATIQUES

---

# PROGRAMMATIONS C

## 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:

**Type nomTableau[taille];**

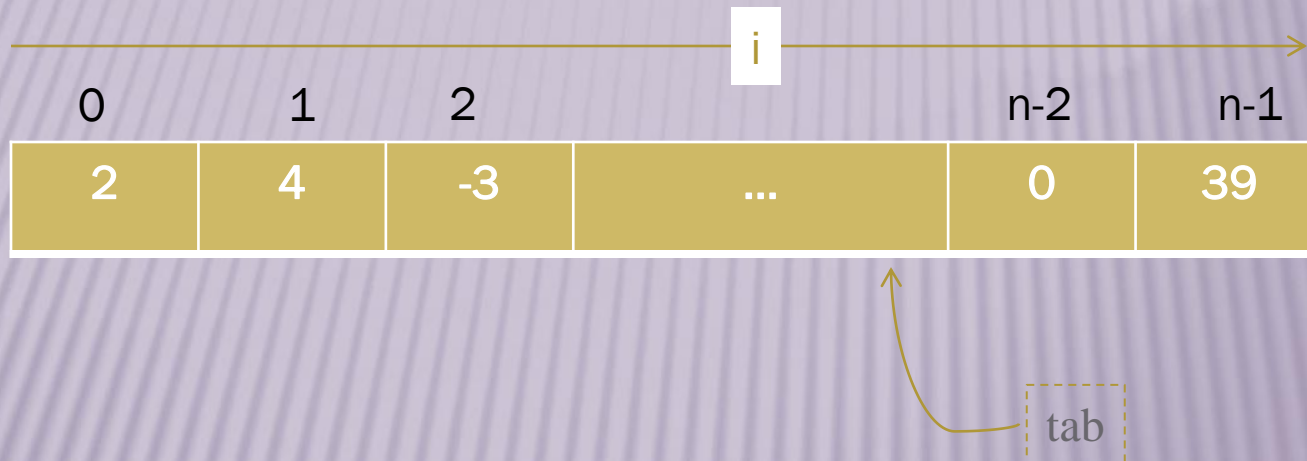
- ✖ **Exemple:**

- ✖ float notes[10] ;
- ✖ char texte[255] ;

# PROGRAMMATIONS C

## TABLEAUX

- ✗ Les éléments d'un tableau sont accessibles pas indice (commençant par 0) que cela soit en lecture ou en écriture.



### + Pour accéder au contenu d'une case:

✗ `printf("format", nomTableau[i]);` exemple: `printf("%d", tab[0]);` // donne 2

### + Pour écrire dans une case avec une valeur saisie au clavier:

✗ `scanf("format", &nomTableau[i]);` exemple: `scanf("%d", &tab[1]);`

### + Par affectation:

✗ `nomTableau[i] = Valeur;` exemple: `tab[n-1] = 27;` // la case n-1 vaut maintenant 27

# PROGRAMMATIONS C

## TABLEAUX

### ✖ Exemple:

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

### ✖ Solution:

```
#include<stdio.h>
int main(){
    float tab[10];
    int i;
    printf("Remplir le tableau")
    for(i=1; i<10; i++){
        scanf("%f", &tab[i]);
    }

    // Affichage du tableau
    for(i=1; i<10; i++){
        printf("%f, \t", tab[i]);
    }
    return 0;
}
```



# PROGRAMMATIONS C

## TABLEAUX

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

**Type nomTableau[ligne][colonne];**

× **Exemple:**

- ★ float notes[10][20] ;
- ★ char texte[10][255] ;
- ★ float matrice[3][4] ;

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

# PROGRAMMATIONS C

## TABLEAUX

- ✖ Les éléments d'un tableau à deux dimensions sont accessibles par indice ligne (**commençant par 0**) et colonne (**commençant par 0 aussi**) que cela soit en lecture ou en écriture.

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

tab

### + Pour accéder au contenu d'une case:

- ✖ `printf("format", nomTableau[i][j]);` exemple: `printf("%d", tab[0][0]);` // donne 2

### + Pour écrire dans une case avec une valeur saisie au clavier:

- ✖ `scanf("format", &nomTableau[i][j]);` exemple: `scanf("%d", &tab[1][2]);`

### + Par affectation:

- ✖ `nomTableau[i][j] = Valeur;` exemple: `tab[n-1][m-2] = 27;` // `tab[n-1][m-2]` vaut maintenant 27

# PROGRAMMATIONS C

## TABLEAUX

### ✖ Exemple:

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

### ✖ Solution:

```
#include<stdio.h>
int main(){
    int M[10][20] ;
    int i , j , val = 1;

    // Remplissage du tableau
    for(i=0; i<10; i++){
        for(j=0; j<20; j++){
            M[i][j] = val;
            val = val+1;
        }
    }

    // Affichage du tableau
    for(i=0; i<10; i++){
        for(j=0; j<20; j++){
            printf("%d \t", M[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```



# PROGRAMMATIONS C

## CAS PRATIQUES N° 5

### ✖ Application 21:

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

### ✖ Application 22:

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

### ✖ Application 23:

Écrire un programme 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 programme qui calcule la somme des éléments d'une matrice.

### ✖ Application 25:

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

Affaires à suivre

