

# Les Matrices

INTRODUCTION AU LANGAGE C



## C

### Les matrices

## Définition :

Une matrice est un tableau de nombres.  
Les nombres sont appelés les éléments de la matrice.

Une matrice à **m** lignes et **n** colonnes est appelée une matrice  $m \times n$ .

Exemple  $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

A est une matrice  $2 \times 3$ .

Les éléments de A sont 1, 2, 3, 4, 5 et 6.



# C

## Les matrices

### Declaration & initialization:

En C, une matrice est un tableau à deux dimensions.  
Voici comment déclarer et initialiser une matrice :

```
#include <stdio.h>
int main() {
    int matrice[3][4];
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 4; j++) {
            matrice[i][j] = i * 4;
        }
    }

    return 0;
}
```



# C

## Les matrices

### Exemple concret :

```
#include <stdio.h>
```

```
#define LIGNES 3
```

```
#define COLONNES 4
```

```
int main() {  
    int matrice[LIGNES][COLONNES];  
    printf("remplir la matrice %d x %d:\n", LIGNES, COLONNES);  
    for(int i = 0; i < LIGNES; i++) {  
        for(int j = 0; j < COLONNES; j++){  
            printf("Element [%d][%d]: ", i, j);  
            scanf("%d", &matrice[i][j]);  
        }  
    }
```



# C

## Les matrices

### Exemple concret :

```
printf("\nVoici votre matrice:\n");  
    for(int i = 0; i < LIGNES; i++) {  
        for(int j = 0; j < COLONNES; j++) {  
            printf("%d\t", matrice[i][j]);  
            printf("\n");  
        }  
    }  
  
    return 0;  
}
```



# C

## Les matrices

### Explication du code :

- Nous définissons la taille de la matrice avec #define:
  - ✓ **LIGNES** pour le nombre de lignes
  - ✓ **COLONNES** pour le nombre de colonnes
- ❖ Pour remplir la matrice:
  - On utilise deux boucles for imbriquées
  - La première boucle (i) parcourt les **lignes**
  - La deuxième boucle (j) parcourt les **colonnes**
  - **scanf** permet à l'utilisateur d'entrer chaque élément



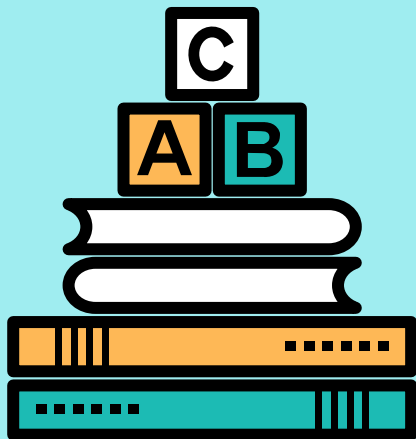
# C

## Les matrices

### Explication du code :

#### Pour afficher la matrice:

- On utilise également deux boucles for imbriquées.
- `printf("%d\t", matrice[i][j])` affiche chaque élément suivi d'une tabulation.
- `printf("\n")` crée une nouvelle ligne après chaque ligne de la matrice.



# CHERCHER

CHERCHER DANS UNE MATRICE





**C**

## Les matrices

### EXEMPLE :

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    int matrice[100][100];

    int lignes, colonnes;
    int valeurRecherchee;
    bool valeurTrouvee = false;
    int ligneTrouvee, colonneTrouvee;
```



# C

## Les matrices

### EXEMPLE :

```
printf("Entrez le nombre de lignes : ");
scanf("%d", &lignes);
printf("Entrez le nombre de colonnes : ");
scanf("%d", &colonnes);

printf("Entrez les éléments de la matrice :\n");
for (int i = 0; i < lignes; i++) {
    for (int j = 0; j < colonnes; j++) {
        printf("Element [%d][%d] : ", i, j);
        scanf("%d", &matrice[i][j]);
    }
}
```



# C

## Les matrices

### EXEMPLE :

```
printf("Entrez la valeur à rechercher : ");
scanf("%d", &valeurRecherchee);

for (int i = 0; i < lignes; i++) {
    for (int j = 0; j < colonnes; j++) {
        if (matrice[i][j] == valeurRecherchee) {
            valeurTrouvee = true;
            ligneTrouvee = i;
            colonneTrouvee = j;
            break;
        }
    }
    if (valeurTrouvee) { break; }
}
```



# C

## Les matrices

### EXEMPLE :

```
if (valeurTrouvee) {  
    printf("trouvée à [%d][%d]\n", ligneTrouvee,  
    colonneTrouvee); }  
else {  
    printf("Valeur non trouvée \n");  
}  
return 0;  
}
```