



SAGIM 2024/2025

Langage C

Les tableaux

Résumé avec des exercices d'explication +

Les tableaux en C

Un **tableau** est une structure de données qui permet de stocker une collection d'éléments de même type.

Les **éléments** sont **stockés** dans des emplacements mémoire contigus, ce qui permet **un accès rapide** à n'importe quel élément via un indice.

Les tableaux en C

Exemple :

tab1	1	2	3	4	5
	[0]	[1]	[2]	[3]	[4]

tab2	1	2	3	4	5	60	70	80	90	100
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

Déclaration d'un tableau

En C, un tableau se déclare de la manière suivante :

`type nom_du_tableau[taille];`

type : le type des éléments du tableau (int, float, char, etc.).

nom_du_tableau : le nom du tableau.

taille : le nombre d'éléments que le tableau peut contenir.

Exemple :

`int nombres[5]; // Déclare un tableau de 5 entiers`

Initialisation d'un tableau

Un tableau peut être initialisé lors de sa déclaration :

```
int nombres[5] = {1, 2, 3, 4, 5};
```

Si vous ne spécifiez pas la taille du tableau, le compilateur la déduira du nombre d'éléments fournis :

```
int nombres[] = {1, 2, 3, 4, 5}; // Tableau de taille 5
```

Accès aux éléments d'un tableau

Les éléments **d'un tableau** sont accessibles via leur **indice**. Les indices commencent à **0** et vont jusqu'à **taille - 1**.

Exemple :

```
int nombres[5] = {1, 2, 3, 4, 5};
```

```
int premier = nombres[0]; // Accède au premier élément
```

```
int dernier = nombres[4]; // Accède au dernier élément
```

Modification

Vous pouvez modifier les éléments d'un tableau en utilisant leur **indice** :

Exemple :

```
int nombres[5] = {1, 2, 3, 4, 5};
```

```
nombres[2] = 10; // Modifie le troisième élément
```

Lecture d'un tableau

La lecture d'un tableau consiste à parcourir les éléments du tableau pour les afficher ou les utiliser dans des calculs.

Exemple :

```
#include <stdio.h>
int main() {
    int nombres[5] = {10, 20, 30, 40, 50};
    printf("Éléments du tableau : ");
    for (int i = 0; i < 5; i++) {
        printf("%d ", nombres[i]); printf("\n");
    }
    return 0; }
```


Remplir un tableau

pour remplir **un tableau** de 5 éléments en utilisant **une boucle**, vous pouvez procéder comme suit.

Voici un exemple où nous allons **remplir un tableau** de nombres entiers,

Remplir un tableau

```
#include <stdio.h>

int main() {
    int tableau[5];
    printf("Entrez 5 nombres entiers :\n");

    for (int i = 0; i < 5; i++) {
        printf("Entrez la valeur pour tableau[%d] :", i);
        scanf("%d", &tableau[i]);    }

    return 0;
}
```

Afficher un tableau

pour afficher **les éléments** d'un **tableau** en utilisant **une boucle**, vous pouvez procéder comme suit.

Voici un exemple où nous allons **afficher** les éléments d'un tableau,

Afficher un tableau

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

```
int main() {
```

```
    int nombres[5] = {1, 2, 3, 4, 5};
```

```
    printf("\nContenu du tableau :\n");
```

```
    for (int i = 0; i < 5; i++) {
```

```
        printf(" nombres[%d] = %d\n", i, nombres[i]);    }
```

```
    return 0;
```

```
}
```



Exercise 1

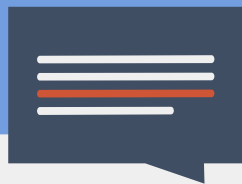
Exercice 1

Énoncé :

Écrivez un programme qui **lit** un tableau de **5** entiers saisis par l'utilisateur, puis **affiche** ces éléments.

Exercice 1

```
#include <stdio.h>
int main() {
    int nombres[5];    int taille = 5;
    printf("Entrez 5 entiers : ");
    for (int i = 0; i < taille; i++) {
        scanf("%d", &nombres[i]); }
    printf("Éléments du tableau : ");
    for (int i = 0; i < taille; i++) {
        printf("%d ", nombres[i]); }
    printf("\n");
    return 0;}
```



Chercher un
élément dans un
tableau

Algorithme de recherche

L'approche la plus courante pour **rechercher** un élément dans **un tableau** est de parcourir chaque **élément** du tableau et de le **comparer** à la valeur recherchée.

Parcourir le tableau : Utilisez une boucle **for** pour **itérer** sur chaque élément du tableau.

Comparer chaque élément : À chaque **itération**, comparez l'élément actuel du **tableau** à la valeur recherchée.

Algorithme de recherche

Retourner la position : Si une correspondance est trouvée, retournez la position (**indice**) de l'élément correspondant.

Retourner "non trouvé" : Si la boucle se termine sans trouver la valeur recherchée, retournez une **indication** que la valeur n'est pas présente dans le tableau.

Algorithme de recherche

```
#include <stdio.h>
#include <stdbool.h>
int main() {
    int taille;
    printf("Entrez la taille du tableau : ");
    scanf("%d", &taille);
    int tableau[taille];
    printf("Entrez les éléments du tableau :\n");
    for (int i = 0; i < taille; i++) {
        printf("Element %d : ", i);
        scanf("%d", &tableau[i]);
    }
```

Algorithme de recherche

```
int valeurRecherchee;  
bool valeurTrouvee = false;  
int positionTrouvee;  
printf("Entrez la valeur à rechercher : ");  
scanf("%d", &valeurRecherchee);  
for (int i = 0; i < taille; i++) {  
    if (tableau[i] == valeurRecherchee) {  
        valeurTrouvee = true;  
        positionTrouvee = i;  
        break;  
    }  
}
```

Algorithme de recherche

```
if (valeurTrouvee) {  
    printf("trouvée à la position %d\n", positionTrouvee);  
}  
  
else {  
    printf("Valeur non trouvée\n");  
}  
  
return 0;  
}
```