

# Chapitre 1 : Les Tableaux à Une Dimension en Langage C

## Introduction

Un tableau est une structure de données qui permet de stocker plusieurs valeurs du même type sous un seul nom de variable. Ces valeurs sont accessibles à l'aide d'un indice (index).

### Exemple d'utilisation :

- ✓ Stocker les notes des étudiants
- ✓ Enregistrer les températures mensuelles
- ✓ Gérer des listes d'éléments (ex. : numéros de téléphone, noms, scores de jeu)

## 1. Déclaration et Allocation d'un Tableau

### 1.1 Déclaration d'un Tableau

Un tableau est déclaré avec un type de données, un nom et une taille.

 type nom\_du\_tableau[taille];

### Exemples :

- int nombres[5]; // Tableau de 5 entiers
- float notes[10]; // Tableau de 10 nombres flottants
- char lettres[26]; // Tableau de 26 caractères

### 1.2 Initialisation d'un Tableau

Un tableau peut être initialisé dès sa déclaration.

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

Si on ne précise pas tous les éléments, les autres sont mis à 0 par défaut.

int tab[5] = {1, 2}; // tab = {1, 2, 0, 0, 0}

On peut aussi initialiser un tableau vide avec zéro :

```
int tab[5] = {0}; // Tous les éléments valent 0
```

## 2. Accès aux Éléments d'un Tableau

Chaque élément est accessible via son indice (qui commence à 0).

- int tab[5] = {10, 20, 30, 40, 50};
- printf("%d\n", tab[0]); // Affiche 10
- printf("%d\n", tab[2]); // Affiche 30

### ⚠ Important :

- tab[0] est le premier élément.
- tab[4] est le dernier élément (indice = taille - 1).
- tab[5] (hors limites) entraîne une erreur de segmentation.

## 3. Parcourir un Tableau

### 3.1 Parcours avec une Boucle for

On peut utiliser une boucle for pour parcourir et afficher un tableau :

```
#include <stdio.h>

int main() {
    int tab[5] = {10, 20, 30, 40, 50};
    int i;
    for (i = 0; i < 5; i++) {
        printf("tab[%d] = %d\n", i, tab[i]);
    }
    return 0;
}
```

## 3.2 Lecture des Valeurs avec scanf

On peut demander à l'utilisateur d'entrer les valeurs du tableau :

```
#include <stdio.h>

int main() {
    int tab[5];
    int i;
    for (i = 0; i < 5; i++) {
        printf("Entrez une valeur pour tab[%d] : ", i);
        scanf("%d", &tab[i]);
    }
    for (i = 0; i < 5; i++) {
        printf("%d ", tab[i]);
    }
    return 0;
}
```

## 4. Opérations sur les Tableaux

### 4.1 Recherche du Maximum et du Minimum

```
#include <stdio.h>

int main() {
    int N, i, min, max;
    printf("Entrez la taille du tableau : ");
    scanf("%d", &N);
    int tab[N];
    for (i = 0; i < N; i++) {
        printf("Entrez l'élément %d : ", i);
        scanf("%d", &tab[i]);
    }
```

```

}

min = max = tab[0];

for (i = 1; i < N; i++) {
    if (tab[i] < min) min = tab[i];
    if (tab[i] > max) max = tab[i];
}

printf("Min : %d, Max : %d\n", min, max);

return 0;
}

```

## 4.2 Vérifier si un Mot est un Palindrome

Un palindrome est un mot qui se lit de la même manière dans les deux sens (ex : OTTO, KAYAK).

```

#include <stdio.h>
#include <string.h>

int main() {
    char mot[100];
    int i, longueur, est_palindrome = 1;
    printf("Entrez un mot : ");
    scanf("%s", mot);
    longueur = strlen(mot);
    for (i = 0; i < longueur / 2; i++) {
        if (mot[i] != mot[longueur - i - 1]) {
            est_palindrome = 0;
            break;
        }
    }
    if (est_palindrome)
        printf("%s est un palindrome.\n", mot);
}

```

```

    else
        printf("%s n'est pas un palindrome.\n", mot);

    return 0;
}

```

### 4.3 Trier un Tableau (Tri par Sélection)

```

#include <stdio.h>

void triSelection(int tab[], int n) {
    int i, j, min, temp;
    for (i = 0; i < n - 1; i++) {
        min = i;
        for (j = i + 1; j < n; j++) {
            if (tab[j] < tab[min]) {
                min = j;
            }
        }
        temp = tab[i];
        tab[i] = tab[min];
        tab[min] = temp;
    }
}

int main() {
    int N, i;
    printf("Entrez la taille du tableau : ");
    scanf("%d", &N);
    int tab[N];
    for (i = 0; i < N; i++) {
        printf("Entrez un nombre : ");
        scanf("%d", &tab[i]);
    }
}

```

```
triSelection(tab, N);

printf("Tableau trié : ");
for (i = 0; i < N; i++) {
    printf("%d ", tab[i]);
}

return 0;
}
```

## 5. Applications Pratiques

- Les tableaux sont utilisés dans plusieurs domaines :
  - ✓ Gestion des notes des étudiants
  - ✓ Statistiques et calculs météorologiques
  - ✓ Analyse de données et algorithmes de tri
  - ✓ Traitement des chaînes de caractères (ex. : détection de palindromes)

## 6. Conclusion

Un tableau permet de stocker plusieurs valeurs sous un même nom.

Les indices commencent à 0.

Les boucles (for, while) permettent de remplir, parcourir et manipuler un tableau.

On peut effectuer des opérations avancées comme tri, recherche, suppression de doublons, etc.

Avec cette maîtrise des tableaux, vous pourrez optimiser vos algorithmes et traiter efficacement des données en C !