

exige la capacité d'identifier la mémoire allouée, les relations entre les données et les processus de données. Or, les structures de données facilitent ces opérations.

Par ailleurs, non seulement il est important d'utiliser des structures de données, mais il est également indispensable de choisir la structure adaptée à chaque tâche. Choisir une structure de données universelle pourrait entraîner un ralentissement des temps de traitement ou une absence de réponse du code.

Plusieurs facteurs sont à prendre en compte lors du choix d'une structure de données :

- le type d'information qui sera stocké,
- l'emplacement de stockage des données existantes,
- le mode de stockage des données,
- la quantité de mémoire à réserver aux données.

## ETUDE D'UNE STRUCTURE DE DONNEES : LES TABLEAUX

Un tableau est une suite de variables de même type , situées dans un espace contigu en mémoire.

### DECLARATION

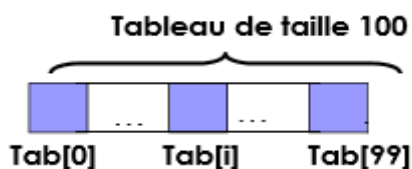
Un tableau en C se déclare à l'aide de 3 informations :

1. Le type des éléments du tableau
2. Le nom du tableau
3. La taille du tableau (le nombre d'éléments)

La syntaxe de déclaration d'un tableau est : **TYPE Nom\_du\_tableau[taille]**

Par exemple, pour déclarer la variable **Tab** comme étant un tableau de 100 entiers, on écrira : **int Tab[100]** allant de **Tab[0]** à **Tab[99]**.

La taille d'un tableau doit être une expression constante (ça ne peut pas être une variable du programme). Les indices vont obligatoirement de 0 à **taille-1**.



**Attention aux problèmes de débordement : l'élément Tab [100] n'existe pas**

## DECLARATION ET INITIALISATION DU TABLEAU

- ✚ Initialisation à la déclaration

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

- ✚ Initialisation dans le programme

Par exemple avec une boucle

```
{...
```

```
Int tab[5];
```

```
Int i;
```

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

```
scanf("%d",&tab[i]);
```

```
}
```

## TABLEAUX ET FONCTION/ PROCEDURE

On est souvent amené à parcourir les éléments d'un tableau pour faire des opérations (Calcul de Moyenne, tri, recherche,...) . Les fonctions sont indispensables pour ne pas avoir à réécrire un nouveau code pour chaque tableau.

- ✚ Un tableau peut donc être passé comme argument à une fonction(ou une procédure)
- ✚ On transmet à la fonction (la procédure) l'emplacement mémoire du début du tableau
- ✚ On parlera de l'adresse du tableau qui est égale à l'adresse du 1er élément du tableau: il s'agit du nom du tableau.

Exemple 1 : dire ce que fait chacun des sous programmes suivants

```
void initialise_tab(int tab[ ], int n)
```

```
{ int i;
```

```
for (i=0;i<n;i++)
```

```
scanf("%d",&tab[i]);
```

```
return;
```

```
}
```

```
int affiche_tab(int tab[ ], int n)
```

```
{ int i;
```

```
for (i=0;i<n;i++)
```

```
printf("%d",tab[i]);
```

```
return 0;
```

```
}
```

## Exemple2 exercice

Ecrire en langage C, un petit programme qui utilise les sous programmes de l'exemple 1 pour créer, initialiser et afficher un tableau de 4 chaînes de caractères.

## UTILISATION D'UN TABLEAU

On souhaite utiliser un tableau pour stocker les notes d'informatique de 10 élèves après une évaluation. on vous demande de :

1. Déclarer un tableau pouvant contenir ces notes.
2. Initialiser un tableau avec les notes obtenues par ces élèves.
3. Ecrire en C le code qui affiche le tableau obtenu.
4. écrire en C le code qui calcule la somme des notes obtenues par ces 10 élèves.

## SOLUTION GUIDEE

La variable i sera utilisée comme compteur, et la variable somme contiendra la somme des notes obtenues par ces élèves

1. Il s'agit de déclarer un tableau de 10 réels : `Float Tab[10] ;`

2. Utilisons la boucle for

```
for (i=0;i<10;i++){  
scanf("%f",&Tab[i]);  
}
```

3. Utilisons la boucle while

```
i=0;  
While(i<10){  
printf("%f",&Tab[i]);  
i++)  
}
```

4. Utilisons la boucle do... while()

```
somme=0;  
do {  
Somme+=Tab[i];  
i++;  
While(i<=9);
```



## RECHERCHE D'UNE VALEUR DANS UN TABLEAU

Ecrivez un programme qui :

1. initialise un tableau de 10 valeurs entières
2. Recherche la valeur minimale et son indice dans le tableau
3. Recherche la valeur maximale et son indice dans le tableau
4. Affiche ces valeurs extrêmes et leur indice respectif.
5. Execute ce programme avec le tableau suivant

|   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|----|
| 5 | 4 | 6 | 2 | 0 | 8 | 1 | 3 | 7 | 11 |
|---|---|---|---|---|---|---|---|---|----|

### SOLUTION GUIDEE

```
include <stdio.h>
#include <stdlib.h>
void affichage (int tab [10]) {
    int i ;
    for (i = 0 ; i < 10 ; i = i + 1){
        printf ("%d ", tab [i]) ;
    }
    printf ("\n") ;
}
int main () {
    int i,min,max,indicemin ;indicemax ;
    int tab [10] ;
    /* On demande a l'utilisateur de remplir les 10 cases du tableau */
    for (i = 0 ; i < 10 ; i = i + 1)
    {
        printf ("Quelle valeur pour la case %d ?\n", i) ;
        scanf ("%d", &tab [i]) ;
    }
    printf ("Tableau avant\n") ;
    affichage (tab) ;
    /* recherche du minimum dans le tableau*/
    min = tab [0] ; indicemin = 0 ;          /* le minimum est initialisé au premier élément*/
    for (i = 1 ; i < 10 ; i ++ )
    {
        if (tab [i] < min) {
            min = tab [i] ;
            indicemin = i ;
        }
    }
    printf ("Minimum : tab [%d] = %d\n", indicemin, min) ;
    /* recherche du maximum */
```



```

max = tab [0] ;                               /* le minimum est initialisé au premier élément*/
indicemax = 0 ;
for (i = 1 ; i < 10 ; i ++)
{
    if (tab [i] > max) {
        max = tab [i] ;
        indicemax = i ;
    }
}
printf ("Maximum : tab [%d] = %d\n", indicemax, max) ;
return 0 ;
}

```

Trace d'exécution du programme

|   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|----|
| 5 | 4 | 6 | 2 | 1 | 8 | 3 | 0 | 7 | 11 |
|---|---|---|---|---|---|---|---|---|----|

| I | Cond              | indicemin | min | Valeur de Tab[i] | Cond              | indicemax | Max |
|---|-------------------|-----------|-----|------------------|-------------------|-----------|-----|
| 0 | XXXX              | 0         | 5   | 5                | XXXX              | 0         | 5   |
| 1 | Tab[1] <5<br>Vrai | 1         | 4   | 4                | Tab[1] >5<br>Faux | 0         | 5   |
| 2 | Tab[2] <4<br>faux | 1         | 4   | 6                | Tab[2] >5<br>Vrai | 2         | 6   |
| 3 | Tab[3] <4<br>Vrai | 3         | 2   | 2                | Tab[3] >6<br>Faux | 2         | 6   |
| 4 | Tab[4] <2<br>Vrai | 4         | 1   | 1                | Tab[4] >6<br>Faux | 2         | 6   |
| 5 | Tab[5] <1<br>faux | 4         | 1   | 8                | Tab[5] >6<br>Vrai | 5         | 8   |
| 6 | Tab[6] <5<br>faux | 4         | 1   | 3                | Tab[6] >5<br>Faux | 5         | 8   |
| 7 | Tab[7] <1<br>Vrai | 7         | 0   | 0                | Tab[7] >1<br>Faux | 5         | 8   |
| 8 | Tab[8] <5<br>Faux | 7         | 0   | 7                | Tab[8] >5<br>Faux | 5         | 8   |
| 9 | Tab[9] <5<br>Faux | 7         | 0   | 11               | Tab[9] >5<br>Vrai | 10        | 11  |



## ACTIVITE D'INTEGRATION

Ecrire en langage C, une fonction nbOccurrences qui prend 3 paramètres : Un tableau d'entiers, Sa taille et une valeur entière quelconque  $x$  ; puis calcule et renvoie le nombre de fois où cette valeur est présente dans le tableau. Et donnez le résultat de l'exécution de votre programme pour le tableau ci-dessous. Avec  $x=11$ .

|    |    |    |   |    |    |
|----|----|----|---|----|----|
| 11 | 23 | 11 | 4 | 11 | 34 |
|----|----|----|---|----|----|

## REINVESTISSEMENT

### EXERCICE1

1. initialise et affiche un tableau de 10 valeurs entières
2. Modifiez le programme afin l'affichage du tableau soit réalisée par une procédure
3. Ecrivez une procedure qui double chacune des valeurs saisies dans le tableau

### EXERCICE 2

Ecrivez un programme qui :

1. initialise un tableau de 10 valeurs réelles
2. affiche l'amplitude du tableau (écart entre le min et le max)
3. affiche la moyenne de ses valeurs.

