

Leçon 4 : Les structures itératives



Objectifs

Ce cours devra vous permettre

- **Déclarer** correctement un tableau en langage C
- **Manipuler** correctement un tableau en langage C

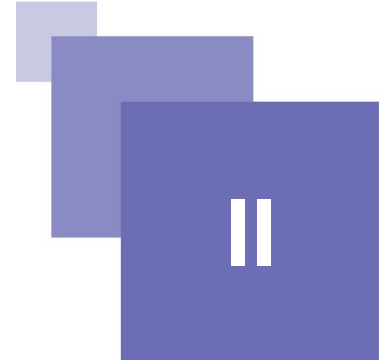
Pré requis



Avant de suivre cette leçon vous devez :

- Avoir pour acquis la connaissance de l'algorithme ;
- Avoir survie la leçon 1 : Concepts généraux du Langage C
- Avoir survie la leçon 2 : Les structures conditionnelles
- Avoir survie la leçon 3 : Les structures itératives

INTRODUCTION



La particularité d'un tableau est de pouvoir stocker de manière temporaire, plusieurs valeurs afin de les manipuler.

Comme tous les langages de programmation, le langage c permet l'utilisation des tableaux. Les valeurs d'un tableau en C sont de même type et stockées en mémoire à des adresses contiguës. La position de chaque valeur dans un tableau est déterminée à partir d'un indice. Le premier indice d'un tableau en C a pour valeur zéro.



2- Déclaration

Les **indices** du tableau partent de **0 (zéro) à dim-1** cela signifie que pour **un tableau de 10 éléments** nous aurons des indices qui vont **de 0 à 9**.



```
int tab[5] ;
```

2	-12	1	9	-5
1		4		

tab[4] a pour valeur -5

Exercice :

EXERCICE N° 1

IV

[Solution n°1 p 21]

EXO 1

1- Soit le tableau défini ci-dessous

2	5	4	5	2	5
---	---	---	---	---	---

```
for(i=0;i<=3;i++)  
{  
    val=i;  
}  
printf("%d",tab[val]);
```

Que retour val à la fin du programme ?

☐ a. 4

☐ b. 5

☐ c. 2

EXO 2

2- `printf("%d",tab[4][3]);`

Dans une matrice nommée *tab*, que retourne cette instruction ?

☐ a. La valeur de la quatrième ligne et troisième colonne

☐ b. La valeur de la quatrième colonne et troisième colonne

☐ c. La valeur de la cinquième ligne et quatrième colonne

EXO 3

Exercice : EXERCICE N° 1

3- Soit le vecteur *tab* suivant :

2	5	6	4	3	6
---	---	---	---	---	---

$val = (tab[2] \% 2 == 0) ? tab[3] * 2 : tab[3] * 3;$

Que retourne *val* ?

☐ a. 8

☐ b. 12

☐ c. 12

II- LES TABLEAUX A DEUX DIMENSIONS



V



Définition : 1- Définition

Encore appelé **matrice**, le **tableau à deux dimensions** est constitué de **plusieurs lignes** et **plusieurs colonnes**.

2- Déclaration

Type `nom_tableau[dim1] [dim2];`

- **dim1** représente le nombre de lignes
- **dim2** représente le nombre de colonnes
- Le premier **indice** de **ligne** a pour **valeur 0** et le dernier a pour **valeur dim1-1**
- Le premier **indice** de **colonne** a pour **valeur 0** et le dernier a pour **valeur dim2-1**.

Pour parcourir une matrice, l'on parcourt chaque ligne de la matrice et étant sur la ligne, l'on parcourt ses différentes colonnes.



Exemple : 3- Exemple

Soient 6 éléments contenus dans un tableau `tab` de type réel représenté ci-dessous et comprenant 2 lignes et 3 colonnes.

2.2	4.0	5.5
0.6	7.2	8.2

NB : L'indice commence toujours par 0 en langage donc on aura :

La valeur **2.2** est situé dans la **ligne** ayant pour **indice 0** et la **colonne** ayant pour **indice 0** : `tab[0][0]`.

La valeur **7.2** est situé dans la **ligne** ayant pour **indice 1** et la **colonne** ayant pour **indice 1** : `tab[1][1]`.

Exercice :

EXERCICE N° 2

VI

[Solution n°2 p 21]

EXO 1

1- Soit la matrice tab suivante :

2	7	0
1	8	2
2	5	9

```
v1=6;
for(i=0;i<3;i++)
{
  for(j=0;j<=2;j++)
  {
    If(tab[i][j]==i)
    { v1++; }
    else
    { v1--; }
  }
}
printf("%d",v1);
Que retourne v1 ?
```

- ☐ a. 5
- ☐ b. 3
- ☐ c. 0

EXO 2

Exercice : EXERCICE N° 2

2- Soit la matrice *tab* suivante :

2	6
1	2
5	5

```
for(i=0;i<2;i++)  
{  
  v1=0;  
  for(j=0;j<=2;j++)  
  {  
    If(v1<tab[j][i])  
    { v1=tab[j][i]; }  
  }  
  tab2[i]=v1;  
}
```

Quelles sont les valeurs respectives contenues dans le tableau *tab2* à la fin du programme ?

- | | |
|-----------------------|--------------|
| <input type="radio"/> | a. 6 2 5 |
| <input type="radio"/> | b. 6 5 |
| <input type="radio"/> | c. 5 6 |

IV- APPLICATION

VII

1- Énonce

Ecrire un programme en langage C permettant de renseigner 10 nombres pairs strictement supérieurs à zéro dans un tableau et de l'afficher.

2- Correction

```
#include<stdio.h> // elle permet d'utiliser les fonctions printf() et scanf()
#include<conio.h> // elle permet de manipuler la console windows (par exemple
effacer l'écran, changer la couleur de police de console etc.)
main()
{
//Déclaration des variables
int i,tab[10];
printf("***** SAISIE DU TABLEAU
*****\n\n");
//Parcours des colonnes du tableau
for(i=0;i<10;i++) // permet de parcourir les colonnes du tableau en
partant de 0 à 9
{
do // cette boucle permettra de répéter la même saisie jusqu'à ce la condition
mentionnée dans l'énoncé soit respecter
{
printf("Entrez un nombre pair strictement supérieur à zéro!\n"); // affiche
le message entre les doubles griffes et retourne à la ligne suivant grâce à \n
scanf("%d",&tab[i]); // récupère la valeur saisie dans un tableau
}
while(tab[i]<=0 || tab[i]%2!=0); vérifie si la valeur saisie est inférieur ou le
reste de la division par 2 est différent de zéro
}
printf("\n\n ***** AFFICHAGE DU TABLEAU
*****\n\n");
for(i=0;i<10;i++) // permet d'afficher le dix éléments du tableau en commençant
pas l'indice 0.
{
printf("%d |",tab[i]);
}
}
```

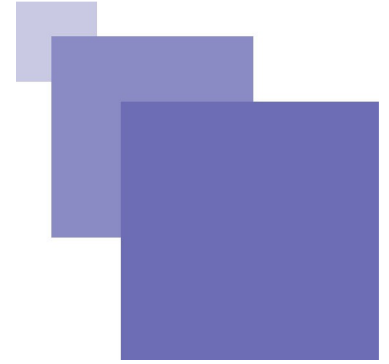
1- Énoncé

Ecrire un programme en langage C permettant de renseigner les 12 premiers nombres multiples de 3 dans une matrice de 3 lignes et 4 colonnes.

2- Correction

```
#include<stdio.h> // elle permet d'utiliser les fonctions printf() et scanf()
#include<conio.h> // elle permet de manipuler la console windows (par exemple
effacer l'écran, changer la couleur de police de console etc.)
main()
{
    int i,k,j,tab[3][4];
    printf("***** RENSEIGNEMENT DU TABLEAU
    *****\n\n");
    k=1;
    // parcours des lignes de la matrice
    for(i=0;i<3;i++)
    {
        //parcours et renseignement des colonnes de chaque ligne de la matrice
        for(j=0;j<4;j++)
        {
            tab[i][j]=3*k;
            k++;
        }
    }
    printf("\n\n ***** AFFICHAGE DU TABLEAU
    *****\n\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
            //Affichage du contenu de chaque ligne suivi d'une tabulation
            printf("%d \t ",tab[i][j]);
        }
        //retour-chariot avant l'affichage du contenu de la ligne suivante
        printf("\n");
    }
}
```

Solution des exercices



> Solution n°1 (exercice p. 13)

EXO 1

- ☐ a. 4
- ☒ b. 5
- ☐ c. 2

EXO 2

- ☐ a. La valeur de la quatrième ligne et troisième colonne
- ☐ b. La valeur de la quatrième colonne et troisième colonne
- ☒ c. La valeur de la cinquième ligne et quatrième colonne

EXO 3

- ☒ a. 8
- ☐ b. 12
- ☐ c. 12

> Solution n°2 (exercice p. 17)

EXO 1

- ☐ a. 5
- ☒ b. 3
- ☐ c. 0

EXO 2

Solution des exercices



a. 6 | 2 | 5



b. 6 | 5



c. 5 | 6