

LES TABLEAUX

Objectif:

- ① Définir ce qu'est un tableau
- ② Connaître les différentes manières de déclaration et d'usage des tableaux

Pré requis:

Les Chapitres précédents de ce cours de langage C

PLAN

I/ DEFINITION D'UN TABLEAU

II/ INITIALISATION DES TABLEAUX

III/ MANIPULATION DES TABLEAUX

IV/ PASSAGE DE TABLEAUX A UNE FONCTION

V/ LES TABLEAUX MULTIDIMENSIONNELS

VI/ LES CHAINES DE CARACTERES

1. La fonction strlen
2. La fonction strcmpi
3. La fonction strcpy

Chapitre 7

LES TABLEAUX

I/ DEFINITION D'UN TABLEAU

Les tableaux sont définis de la même manière que les variables ordinaires suivie d'une spécification de taille.

Un tableau est une suite d'éléments contiguës en mémoire. On peut trouver des tableaux unidimensionnels dites vecteurs ou des tableaux bidimensionnels dites matrices ou des tableaux multidimensionnels.

La syntaxe générale d'une définition d'un tableau unidimensionnel est la suivante:

```
Type_de_tableau tableau[Taille];
```

Exemple:

```
int X[100];      Un tableau de 100 entiers.  
char texte[80]; Un tableau de 80 caractères.  
float n[12];    Un tableau de 12 réels.
```

Exemple:

```
/* Lecture du texte en minuscule et affichage du texte correspondant en majuscule */  
# include <Stdio.h>  
# define dim 80  
main()
```

```
{ char lettre[dim];
    int compteur;
    /* Lecture d'une ligne */
    for (compteur=0;compteur<dim;++compteur)
        lettre[compteur]=getchar();
    /* affichage de la ligne du texte convertie */
    for (compteur=0;compteur<dim;++compteur)
        putchar(upper(lettre[compteur]));
}
```

II/ INITIALISATION DES TABLEAUX

Soient les déclarations suivantes:

```
int chiffres[10]={1,2,3,4,5,6,7,8,9,0}
```

```
float n[6]={0,0.25,.0,-0.5,0,0}
```

```
char couleur[4]={'B','L','E','U'}
```

Ces affectations produisent l'affectation des valeurs suivantes aux éléments de ces tableaux:

chiffre[0]=1	X[0]=0	couleur[0]='B'
chiffre[1]=2	X[1]=0.25	couleur[1]='L'
chiffre[2]=3	X[2]=0	couleur[2]='E'
chiffre[3]=4	X[3]=-0.5	couleur[3]='U'
chiffre[4]=5	X[4]=0	
chiffre[5]=6	X[5]=0	
chiffre[6]=7		
chiffre[7]=8		
chiffre[8]=9		
chiffre[9]=0		

Remarques:

✗ L'indice d'un tableau de n éléments est un nombre variant de 0 à (n-1).

✗ Tout élément d'un tableau dont la valeur n'est pas explicitement initialisée prend automatiquement la valeur 0.

Exemple: `int chiffre[10]={3,3,3} /* chiffre[3] ..chiffre[9] =0 */`

✗ Pour les chaînes de caractères le dernier élément doit être '\0' (caractère nul).

III/ MANIPULATION DES TABLEAUX

Il n'existe pas en C d'opérations simples sur l'ensemble des tableaux. Ainsi, si a et b sont deux tableaux comparables (de même type) les opérations d'affectation et de comparaison doivent se faire élément par élément.

Exemple: Calcul de l'écart à la moyenne d'une série de nombres.

Soit à calculer la moyenne d'une série de n valeurs réels. On cherche de plus à connaître pour chaque valeur de la série donnée, la différence par rapport à la moyenne de la série au moyen de la formule $d = X_i - \text{moy}$.

Pour résoudre ce problème il va falloir enregistrer chaque valeur dans un tableau unidimensionnel de type float.

```
# include <stdio.h>
main()
{
    int n, compteur;
    float moy, d, somme = 0;
    float valeur[100];
    printf(" \n Nombre de valeur à traiter =");
    scanf(" %d",&n);
    /* Lecture des valeurs et calcul de leurs sommes */
    for (compteur=0, compteur<n, ++compteur)
    {
        printf("valeur[%d]=% ", compteur+1);
        scanf("%f ", &valeur[compteur]);
        somme += valeur[compteur];
    }
    /* Calcul et affichage de la moyenne */
    moy = somme / n;
    for (compteur=0, compteur<n, ++compteur)
    {
        d = valeur[compteur] - moy;
        printf(" valeur[%d] =%f      d=%f ", compteur+1 , valeur[compteur], d);
    }
}
```

IV/ PASSAGE DE TABLEAUX A UNE FONCTION

Le passage d'un tableau à une fonction se fait en mentionnant son nom seulement sans crochet ni indice. L'argument formel correspondant est écrit de la même manière mais sa déclaration doit indiquer qu'il s'agit d'un tableau. Lorsqu'on déclare un tableau unidimensionnel comme argument formel son nom doit être suivi d'une paire de crochets vides. La taille du tableau n'a pas à être spécifiée dans la déclaration d'argument formel.

Exemple:

```
# include <stdio.h>
main()
{
    int compteur, T[5];
    void modifie(int a[ ]);
    printf(" dans main avant appel de modifie ");
    for ( compteur = 0 , compteur <= 5 , ++compteur )
    {
        T[compteur] = compteur + 1;
        Printf(" T[%d] = %d \n" , compteur , T[compteur]);
    }
    modifie(T);
    for ( compteur = 0 , compteur <= 5 , ++compteur )
    printf(" dans main après appel de modifie ");
}
/*-----*/

void modifie( int a[ ])
{
    int i;
    printf(" Dans modifie , après modification des valeurs ");
    for ( i = 0 ; i <= 5 ; ++i)
    {
        a[compteur] = -9;
        printf(" a[%d] = %d \n " , i , a[compteur]);
    }
    return;
}
```

Remarque:

Ces résultats illustrent donc que la modification des valeurs de T dans "modifie" portent également sur ces éléments dans la fonction main.

Exercice:

Ecrire un programme C qui permet de tirer une liste de N entiers dans l'ordre croissant. Ce programme doit utiliser une fonction tri qui à pour argument le nombre N des éléments à trier et un tableau T qui contient les éléments à trier.

Solution:

```
# include <stdio.h>
# define dim 100
main()
{
int i , a , T[dim];
void tri ( int p , int a[ ]);
printf(" \n nombre de valeur à saisir ");
scanf(" %d " , &n);
printf(" \n ");
    /* Lecture de la série de valeurs */
    for (i = 0 ; i < n ; ++i )
    {
        printf(" i = %d      x = " , i+1 );
        scanf(" %d " ,&T[i] );
    }
    /* tri du tableau */
    tri(n,T);
    /* affichage des valeurs triées */
    printf(" \n\n valeurs triées par ordre croissant ");
    for (i = 0 ; i < n ; ++i )
        printf(" i = %d      T = %d \n" , i+1 ,T[i] );
}
```

```
/*-----*/
/*          fonction de tri          */
/*-----*/
void tri( int p , int a[ ] );
{
    int i , indice , temp ;
    for (indice = 0 ; indice < n-1 ; ++indice)
        /* détermination du minimum des éléments restants */
        for( i = indice + 1 ; i < n ; ++i)
            if ( a[i] < a[indice] )
                {
                    /* permutation de deux éléments */
                    temp = a[indice];
                    a[indice] = a[i];
                    a[i] = temp;
                }
    return
}
```

V/ LES TABLEAUX MULTIDIMENSIONNELS

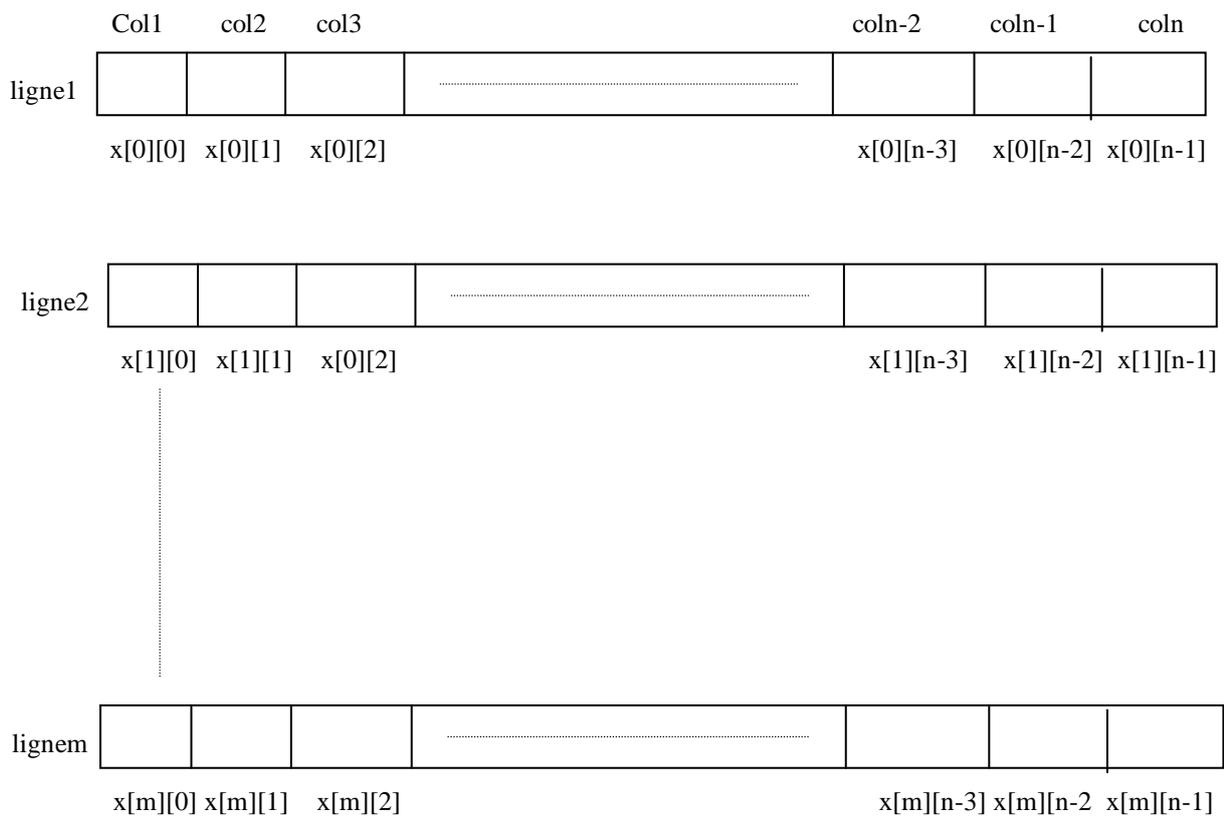
Les tableaux multidimensionnels se définissent de façon analogue aux tableaux unidimensionnels mais il faut une paire de crochets pour chaque indice.

Un tableau à deux dimensions nécessite donc deux paires de crochets, un tableau à trois dimensions en nécessite trois paires et ainsi de suite.

La syntaxe générale de la définition d'un tableau à n dimensions est la suivante:

Type_de_donnée tableau[exp1][exp2]...[expn]

Un tableau de deux dimensions de $m*n$ éléments peut être représenté sous la forme d'un tableau de valeurs à deux entrées composé de m lignes et de n colonnes.

Exemple:

x est un tableau bidimensionnel de m lignes et de n colonnes.

Exercice: Addition de deux tableaux bidimensionnels (A et B)

Il va falloir additionner les deux tableaux élément par élément et mettre le résultat dans un troisième tableau (C).

$$C[i][j]=A[i][j] + B[i][j].$$

Le programme a rédiger peut être découpé en plusieurs fonctions chargées respectivement de saisir les valeurs d'un tableau, calculer la somme des éléments et affiche un tableau.

solution:

```
# include <stdio.h>
# define maxl 20
# define maxc 30
/* calcul de la somme des éléments de deux tableaux */
main()
```

```
{ int nlignes , ncols ;
int A[maxl][maxc], B[maxl][maxc], C[maxl][maxc] ; /* définition de tableaux */
/* prototypes de fonctions */
void saisie_T(int a[ ][maxc], int m , int n) ;
void addition_T(int a[ ][maxc], int b[ ][maxc], int c[ ][maxc], int m, int n) ;
void affich_T(int a[ ][maxc], int m, int n) ;
printf(" Nombre de lignes= ") ;
scanf("%d",&nlignes) ;
printf(" nombre de colonnes = ") ;
scanf("%d",&ncols) ;
printf("\n\n premier tableau") ;
saisie_T(A, nlignes, ncols) ;
printf("\n\n second tableau ") ;
saisie_T(B, nlignes, ncols) ;
printf(" \n\n la somme est = ") ;
addition_T(A, B, C, nlignes, ncols) ;
affiche_T(C, nlignes, ncols) ;
}
/*-----*/

void saisie_T(int a[ ][maxc], int n , int m)
/* Saisie d'un tableau d'entiers */
{
int ligne, col ;
for(ligne=0 ; ligne<m ; ++ligne)
    {
        printf(" saisie de ligne N° :%2d \n", ligne+1) ;
        for(col=0 ; col<n ; ++col)
            scanf("%d" ,&a[ligne][col]) ;
    }
return ;
}
/*-----*/

void addition_T(int a[ ][maxc], int b[ ][maxc], int c[ ][maxc], int m, int n)
```

```
/* Addition des éléments de deux tableaux d'entiers */
{
int ligne,col ;
for(ligne=0 ; ligne<m ; ++ligne)
    for(col=0 ; col<n ; ++col)
        c[ligne][col]=a[ligne][col] + b[ligne][col] ;
return ;
}
/*-----*/

void affich_T(int a[ ][maxc], int m, int n)
/* Affichage du tableau résultat */
{
int ligne,col ;
for(ligne=0 ; ligne<m ; ++ligne)
    {
    for(col=0 ; col<n ; ++col)
        printf(" %4d" , a[ligne][col]) ;
        printf("\n ") ;
    }
return ;
}
```

Remarque :Le nom du tableau est suivi de deux paires de crochets, la première est vide car le nombre de lignes n'a pas à être impérativement renseigné. Par contre, la seconde doit contenir une valeur qui précise le nombre maximum de colonnes que peut contenir le tableau.

VI/ LES CHAINES DE CARACTERES

Une chaîne de caractères peut être traitée comme un tableau de type caractères, dans lequel chaque élément renferme un caractère de la chaîne. Cette chaîne doit se terminer par

'\0'. Le langage C propose des bibliothèques contenant des fonctions spécialisées dans la manipulation des chaînes. La manipulation des chaînes de caractères concerne surtout la comparaison, le copiage et la concaténation des chaînes.

1- La fonction strlen :

Cette fonction retourne la longueur d'une chaîne de caractères.

Prototype :

```
int strlen(char chaîne[ ] ) ;
```

Exemple :

```
int i ;  
char chaîne[10]={ 'T', 'U', 'N', 'I', 'S', 'I', 'E' } ;  
i=strlen(chaîne) ;  
printf(" %s", chaîne ) ;
```

Ce programme affiche la valeur 7.

2- La fonction strcmpi :

Prototype :

```
int strcmpi(char chaîne1[ ], char chaîne2[ ] ) ;
```

Cette fonction retourne une valeur négative si chaîne1 < chaîne2, une valeur nulle si les deux chaînes sont identiques et une valeur positive si chaîne1 > chaîne2.

3- La fonction strcpy :

Prototype :

```
Void strcpy(chaîne1[ ], chaîne2[ ] ) ;
```

Cette fonction copie la valeur de chaîne1 dans chaîne2.

Exercice: Ecrire un programme C permettant de trier plusieurs chaînes (tableaux bidimensionnels) finissant par la chaîne " FIN " suivant l'ordre croissant.

ISET DE SFAX
II N2**TP N°4**
PROGRAMMATION CA.U 1997/1998
Cours de Mr: TAYARI Lassaad*Les tableaux***Objectifs:**

- ① Se familiariser avec la manipulation des tableaux
- ② Pratiquer l'utilisation des fonctions.

Durée: 2 séances de trois heures chacune**EX N°1**

Soit T un tableau de 10 réels non nuls, établir un programme C permettant de diviser ce tableau par son élément d'indice K et de l'afficher.

EX N°2:

Ecrire un programme C qui lit les éléments d'un tableau T1 et forme un deuxième tableau T2 selon le principe suivant:

- * Les éléments pairs de T1 sont rangés dans T2 dans le même ordre (de gauche à droite).
- * Les éléments impairs de T1 sont rangés dans T2 dans l'ordre inverse (de droite à gauche).

Exemple

T1	14	7	6	21	18	77	2	8	101	88
T2	14	6	18	2	8	88	101	77	21	7

EX N°3:

Soient U et V deux vecteurs de l'espace vectoriel \mathcal{R}^n .

U (X1,X2,X3,.....,Xn).

V (Y1,Y2,Y3,.....,Yn).

Ecrire un programme qui calcule et affiche leur produit scalaire P et leurs normes NU et NV.

$P = X1*Y1+X2*Y2+.....+Xn*Yn.$

$NU = \sqrt{X_1^2 + X_2^2 + X_3^2+X_n^2} .$

$NV = \sqrt{Y_1^2 + Y_2^2 + Y_3^2+Y_n^2} .$

EX N°4:

Ecrire un programme C qui détermine la présence d'un entier A dans un tableau d'entiers, si A existe alors il faut afficher tous les positions et le nombre d'apparition sinon afficher le message '*INTROUVABLE*'.

EX N°5:

Etablir un programme C permettant de lire deux tableau d'éléments de type caractère, de former un troisième qui contient les maximums de deux éléments de même indice .

EX N°6:

Soit T un tableau de caractères, établir un programme C qui remplace les virgules(,) par des points virgules(;), et les points(.) par des points d'interrogation (?).

EX N°7:

Etablir un programme C permettant de sommer les valeurs de la diagonale d'une matrice carrée et de la remettre à zéro.

EX N°8: (*Traduction de texte en Louchebem*)

Le louchebem est "le langage de bouchers", à savoir un jargon consistant à rajouter en fin d'un mot sa première lettre suivie des lettres "em" et le remplacer par la lettre "l". Ainsi "voiture" par exemple devient "loiturevem", "vache" devient "lachevem", "bonne" devient "lonnebem", "bouche" devient "louchebem", ... etc.

Développez un programme C permettant de lire une ligne de texte ordinaire n'atteignant pas 80 caractères et dont les mots sont séparés par des blancs et de la traduire en louchbem..

Bon Travail