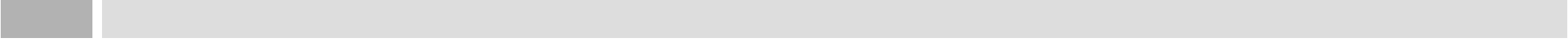




FACULTE DES SCIENCES DE TUNIS
DEPARTEMENT INFORMATIQUE

Atelier programmation I

Enseignant: Hedia Kochkar



Chapitre 3

Les tableaux et les chaînes de caractères

Chapitre 3



1- Les tableaux

1.1- Les tableaux à une dimension

1.2- Les tableaux à deux dimensions

2- Les chaines de caractères

2.1- Les chaines de caractères simples

2.2- Les fonctions sur les chaines de caractères

2.3- Les tableaux de chaines de caractères

1 - Les Tableaux

1.1. Les tableaux à une dimension

- Un tableau est un ensemble fini d'éléments de même type, stockés en mémoire à des adresses contiguës
- Déclaration `type nom_tableau[nombre_elements]`
- *nombre_elements*: est une constante entière positive
- Exemple: `float Moy[20] ;`
- L'indice du premier élément est toujours 0.
- L'indice du dernier élément est alors *nombre_element-1*

1 - Les Tableaux

1.1. Les tableaux à une dimension

- Accès aux éléments :

nom_tableau [indice]

Exemples :

//affecter la moyenne 17 au 2ème élément

Moy[1] = 17 ;

//Saisie de la moyenne du 5ème élément

scanf("%f", &Moy[4]);

//Affichage de la moyenne du 5ème élément

Printf("%f", Moy[4]);

1 - Les Tableaux

1.1. Les tableaux à une dimension

- Le nom du tableau est une adresse constante du premier élément

```
Tab=&tab[0]
```

- Un tableau ne peut pas figurer à gauche d'un opérateur d'affectation

```
Tab1=tab2;  
Faux
```


□ Pour copier un tableau dans un autre:

```
void main()
{
    const int N=10;
    int tab1[N], tab2[N];
    int i;
    ...
    for (i = 0; i < N; i++)
        tab1[i] = tab2[i];
}
```

1 - Les Tableaux

1.1. Les tableaux à une dimension

- On peut initialiser un tableau lors de sa déclaration par une liste de constantes de la façon suivante:

type nom_tableau[] = {constante_1,constante_2,...,constante_N};

- Exemple:

```
main()
{
int tab[] = {1, 2, 3, 4};

}
```

1 - Les Tableaux

1.1. Les tableaux à une dimension

- Écrire un programme C qui permet de :
 - Lire et d'afficher un tableau T de 10 éléments.
 - Calculer la somme des éléments de T.
 - Chercher le maximum et le minimum de T.

1 - Les Tableaux

1.1. Les tableaux à une dimension

```
#include<stdio.h>
void main()
{
    int T[10];
    int i, Som, Max, Min;

    //Saisie de T
    for(i=0;i<10;i++)
    {
        printf ("Donner l'élément d'indice %d ", i);
        scanf ("%d", &T[i]);
    }
```

1 - Les Tableaux

1.1. Les tableaux à une dimension

```
// Affichage de T
for(i=0;i<10;i++)
    printf ("\nT[%d]=%d", i, T[i]);

// Somme de T
Som =0;
for(i=0;i<10;i++)
    Som=Som+T[i];
printf("\nLa somme de T = %d", Som);
```



Max et Min de T

Max = T[0];

Min = T[0];

for(i=1;i<10;i++)

 if(T[i]>Max)

 Max = T[i];

 else

 if(T[i]<Min)

 Min = T[i];

printf (“\nLe max de T = %d et min de T = %d“, Max, Min);

}

1 - Les Tableaux

1.2. Les tableaux à deux dimensions

□ Déclaration :

Type Nom [Dim1][Dim2];

Avec

- Type : Type des éléments du tableau.
- Nom : Nom du tableau
- Dim1 : Nombre de lignes
- Dim2 : Nombre de colonnes

□ Exemple :

```
float M[3][4];
```

M	0				3
0					
2					

1 - Les Tableaux

1.2. Les tableaux à deux dimensions

Accès aux éléments : `Nom [Ind_Ligne][ind_Colonne]`

Exemples :

//Affecter 22 à l'élément de la ligne 2 et de la colonne 3

`M[1][2] = 22`

//Saisie de l'élément de la ligne 3 et de la colonne 1

`Scanf("%f", &M[2][0]);`

//Affichage de l'élément de la ligne 3 et de la colonne 1

`printf("%f", M[2][0]);`

1 - Les Tableaux

1.2. Les tableaux à deux dimensions

□ Exercice :

Écrire un programme C qui permet de :

- Lire et d'afficher une Matrice M de 7 lignes et 5 colonnes.
- Calculer la somme de des éléments de M .

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

```
- void main()
```

```
{
```

```
int M[7][5];
```

```
int i, j, Som, Max, Min;
```

1 - Les Tableaux

1.2. Les tableaux à deux dimensions

```
#include<stdio.h>
void main()
{
int M[7][5];
int i, j, Som, Max, Min;

//Saisie de M
for (i=0;i<7; i++)
    for (j=0;j<5; j++)
    {
        printf ("Donner un élément ");
        scanf("%d", &M[i][j]);
    }
```



```
// Affichage de M
```

```
for (i=0;i<7; i++)
```

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

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

```
// Somme de M
```

```
Som =0;
```

```
for (i=0;i<7; i++)
```

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

```
        Som =Som + M[i][j];
```

```
printf ("La somme de M = %d", Som);
```

1 - Les Tableaux

1.2. Les tableaux à deux dimensions

```
// Max et Min de M
Max = M[0][0];
Min = M[0][0];

for (i=0;i<7; i++)
    for (j=0;j<5; j++)
        if (M[i][j]>Max)
            Max = M[i][j];
        else
            if(M[i][j]< Min)
                Min = M[i][j];
printf("Le max de M = %d et le min de M = %d", Max,
Min);
}
```

2- Les chaines de caractères

2.1. Les chaines de caractères simples

Définition

Une chaîne est un tableau de caractères dont le dernier élément significatif est le caractère '\0'.

Déclaration :

Syntaxe :

char Nom[Dim];

Avec

- Nom : Nom de la chaîne
- Dim : Dimension de la chaîne

Exemple :

char CH[20];

2- Les chaines de caractères

2.1. Les chaines de caractères simples

- Remarques :

Le compilateur ajoute toujours le caractère NUL ('\0') à la fin de la chaîne en mémoire.

2- Les chaines de caractères

2.2. Les fonctions sur les chaines de caractères

- **Bibliothèque `<stdio.h>`**
- **scanf** : `scanf("%s" , chaine)` : Lecture de chaine de caractères. La fonction `scanf` ne permet pas la saisie d'une chaîne comportant des espaces: les caractères saisis à partir de l'espace ne sont pas pris en compte mais rangés dans le tampon.
- **gets** : `gets( <Chaîne> )`: Lecture de chaine de caractères contenant des espaces.
- **printf** : `printf("%s" , chaine)`: Affichage de chaines de caractères
- **puts** : L'écriture de chaînes de caractères `puts( <Chaîne> )`

2- Les chaines de caractères

2.2. Les fonctions sur les chaines de caractères

- **Bibliothèque <stdio.h>**
- **sscanf** : La source de lecture n'est plus l'entrée standard **stdin** mais une chaine de caractère et sa syntaxe est :
`sscanf(chaine_source,"%type",&var_dest);`
- **sprintf**: `sprintf(chaine_dest,« %type",variables);`: Ecriture de données dans une chaine de caractères.

2- Les chaines de caractères

2.2. Les fonctions sur les chaines de caractères

- **Bibliothèque `<string.h>`**
- **`strlen(s)`** : fournit la longueur de la chaîne sans compter le `'\0'`
- **`strcpy(s, t)`** : copie t vers s, Remarque : le `'\0'` est aussi copié
- **`strncpy(s,t,n)`** : copie les n premiers caractères de t vers s
- **`strcat(s, t)`** : ajoute t à la fin de s (à partir du `'\0'` de s qui sera décalé)
- **`strncat(s,t,n)`** : concatène les n premiers caractères de t à la fin de s
- **`strcmp(s, t)`** : compare s et t lexicographiquement :
 - négatif si s précède t
 - zéro si s est égal à t
 - positif si s suit t
- **`strncmp(s, t, n)`** compare les n premiers caractères de s et t

2- Les chaines de caractères

2.2. Les fonctions sur les chaines de caractères

- **Bibliothèque `<stdlib.h>`**
- Conversion de chaines en nombres et inversement
 - Chaîne --> Nombre
- **`atoi(s)`** : retourne la valeur numérique représentée par `<s>` comme int
- **`atol(s)`** : retourne la valeur numérique représentée par `<s>` comme long
- **`atof(s)`** : retourne la valeur numérique représentée par `<s>` comme double

2- Les chaines de caractères

2.2. Les fonctions sur les chaines de caractères

□ **Bibliothèque <stdlib.h>**

▪ Nombre --> Chaîne

□ **itoa (n_int, s, b)**

□ **ltoa (n_long, s, b)**

□ **ultoa (n_uns_long, s, b)**

Avec :

- n_int: est un nombre du type int
- n_long : est un nombre du type long
- n_uns_long : est un nombre du type unsigned long
- s : est une chaîne de caractères
- b : est la base pour la conversion (2 ... 36)

2- Les chaines de caractères

2.3. Les tableaux de chaines de caractères

- Définition

Un tableau de chaînes de caractères est un tableau de tableaux de caractères.

- Déclaration :

Syntaxe :

`char Nom[Dim1][Dim2]`

- Avec

- Nom : Nom de la chaîne

- Dim1 : Nombre de mots

- dim2 : Dimension des mots

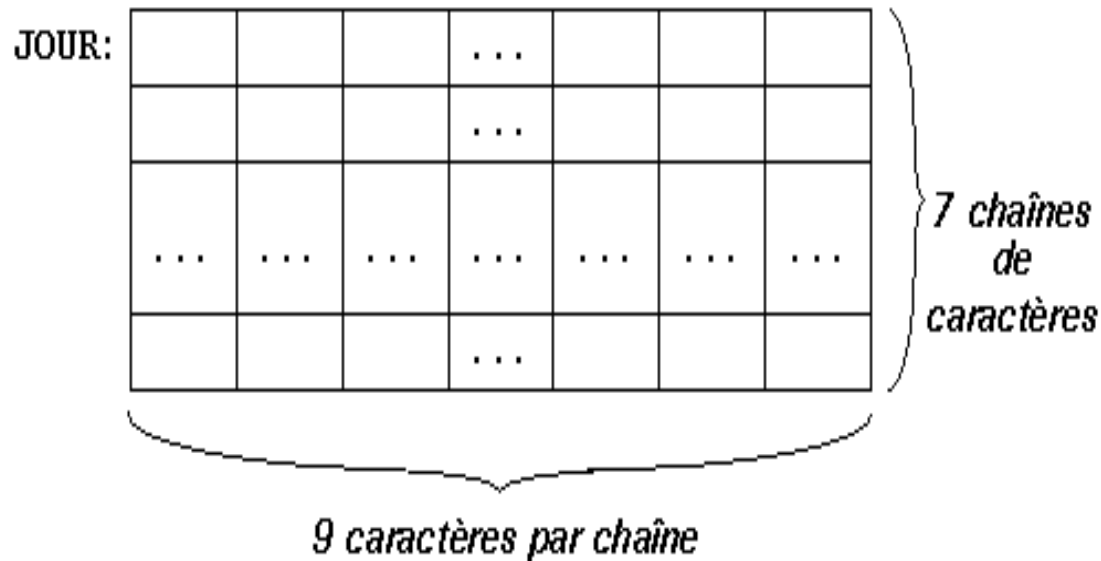
2- Les chaines de caractères

2.3. Les tableaux de chaines de caractères

- Exemple

- La déclaration **char JOUR[7][9];**

réserve l'espace en mémoire pour 7 mots contenant 9 caractères (dont 8 caractères significatifs).



2- Les chaines de caractères

2.3. Les tableaux de chaines de caractères

- Initialisation:
- `char JOUR[7][9] = {"lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi", "dimanche"};`

JOUR:

'l'	'u'	'n'	'd'	'i'	'\0'			
'm'	'a'	'r'	'd'	'i'	'\0'			
'm'	'e'	'r'	'c'	'r'	'e'	'd'	'i'	'\0'
...	...	...	...	...	...	...	...	...
'd'	'i'	'm'	'a'	'n'	'c'	'h'	'e'	'\0'

2- Les chaines de caractères

2.3. Les tableaux de chaines de caractères

```
int l = 2;  
printf("Aujourd'hui, c'est %s !\n", JOUR[l]);
```

□ affichera la phrase:

Aujourd'hui, c'est mercredi !

2- Les chaines de caractères

2.3. Les tableaux de chaines de caractères

□ Affectation

Des expressions comme `JOUR[I]` représentent *l'adresse* du premier élément d'une chaîne de caractères. On ne peut pas donc 'modifier' une telle adresse par une affectation directe !

~~`JOUR[4] = "Friday";`~~

L'attribution d'une chaîne de caractères à une composante d'un tableau de chaînes se fait à l'aide de la fonction **strcpy**:

`strcpy(JOUR[4], "Friday");`