



# Cours programmation :tableaux et chaînes de caractères

## I- Les tableaux:

Un tableau est un ensemble d'éléments rangés en mémoire dans des cases consécutives .  
Un tableau peut être constitué de plusieurs lignes et colonnes. Nous n'utiliserons que les tableaux à une seule ligne.

Numéro de case	0	1	2	3	4	5	6	7
Contenu	A	B	C	D	E	F	G	H

Notez que les cases d'un tableau sont numérotées à partir de 0 en langage C.

## Déclaration:

Un tableau se déclare de la manière suivante :

```
<type> <nom du tableau> [<taille du tableau>];  
  
/* Déclaration d'un tableau de 10 caractères */  
char tab_char [10];  
/* Déclaration d'un tableau de 10 entiers */  
int tab_int [10];
```

## Utilisation:

On accède à une case du tableau en mettant le nom du tableau, suivi d'un crochet ouvrant «[ » puis un numéro de case et un crochet fermant : « ] ».  
Cela donne, par exemple :

```
/* déclarations */  
int tab_int[10]; /* tableau de 10 cases (0 à 9) d'entiers */  
char tab_char[10]; /* tableau de 10 cases (0 à 9) de caractères */  
/* utilisation */  
tab_char[3]='C'; /* Initialisation de la case 3 (la quatrième) de tab_char */  
tab_int[6]=10; /* Initialisation de la case 6 (la septième) de tab_int */  
  
tab_int[7]=tab_int[6] * 2; /* La case 7 (la huitième) contiendra donc 20  
(10*2) */
```

N'oubliez pas que le compilateur ne vérifie pas que vous utilisez le tableau dans ses limites.  
Il vous est donc possible d'écrire à l'extérieur de votre tableau.  
C'est l'un des bugs les plus courants de la programmation en C.

## II- Les chaînes de caractères :



Les chaînes de caractères sont des tableaux de caractères suivis du 0 qui est considéré lui aussi comme un caractère.

Une chaîne s'écrit donc : contenu utile de la chaîne + valeur 0.

« Eric » s'écrit dans un tableau de 5 caractères de la façon suivante :

caractère	'E'	'r'	'i'	'c'	'\0'
code ASCII	69	114	105	99	0
case	0	1	2	3	4

## Déclaration d'une chaîne de caractère:

Une chaîne de caractères se déclare sous la forme d'un tableau de caractères de longueur fixe.

Attention, comme signalé auparavant, si vous dépassez la longueur de tableau, vous écrivez chez le voisin.

Ainsi :

```
char m_chaine [20] ;
```

permettra d'enregistrer des chaînes de 19 caractères maximum (20-1 pour le 0 de fin de chaîne).

Il est possible de déclarer une chaîne de caractères sans en spécifier la longueur de départ de la façon suivante :

```
char chaine [] = "Eric" ;
```

De cette façon, la chaîne fera exactement la longueur nécessaire pour stocker « Eric » et le 0 final soit 4+1=5 octets.

## Affichage d'une chaîne de caractère:

Une chaîne de caractères s'affiche grâce à la commande printf et le format %s.

Ainsi :

```
printf("%s", chaine);
```

## Longueur d'une chaîne de caractère:

La longueur d'une chaîne de caractères s'obtient par la fonction strlen (disponible au travers de la bibliothèque string).

Le 0 de fin de chaîne n'est pas compté dans cette longueur.

```
#include <stdio.h>
#include <string.h>
int main () {
    char ch [] = "toto";
    printf("La longueur de %s est : %d", ch, strlen(ch));
    return 0;
}
```

Affichera : « La longueur de toto est : 4 » à l'écran.



## La fonction gets: saisie d'une chaîne de caractères

La fonction gets permet de saisir une chaîne de caractères validée par la touche Entrée.

```
#include <stdio.h>
int main(void) {
    char line[81];
    /* 81 : taille arbitraire supposée suffisante
       Une ligne écran = 80 caractères + 1 case
       pour le '\0' de fin de chaîne */
    printf( "Saisissez une chaîne de caractère :\n" );
    gets( line );
    /* La frappe de l'utilisateur sera enregistrée dans
       line, on suppose qu'il ne frappera pas plus de
       80 caractères, sinon aïe aïe aïe */
    printf( "\nLa chaîne saisie est : %s\n", line );
    return 0;
}
```

## Quelques fonctions utiles:

### La fonction strcat:

La fonction strcat(<s>, <t>) ajoute la chaîne de caractères <t> à la fin de <s> (on appelle cela une concaténation).

```
#include <stdio.h>
#include <string.h>
int main () {
    char chaine1[20]="Bonjour ";
    char chaine2[20]="Paul";
    strcat(chaine1,chaine2); /* ajoute chaine2 à la fin de chaine1 */
    printf("%s\n",chaine1);
    return 0;
}
```

Le programme suivant affichera la chaîne « Bonjour Paul » à l'écran .

On remarquera qu'il est important de dimensionner chaine1 à une taille suffisante, sans quoi on pourrait avoir des difficultés pour stocker la chaîne « Bonjour Paul » dans chaine1.

### La fonction strcpy:

La fonction strcpy(<s>, <t>) copie les caractères de la chaîne <t> au début de <s>.

```
#include <stdio.h>
#include <string.h>
int main () {
    char chaine1[20]="Bonjour ";
    char chaine2[20]="Edouard";
    strcpy(chaine1,chaine2); /* la chaine2 à l'adresse de chaine1 */
    printf("%s\n",chaine1);
    return 0;
}
```



```
}
```

## La fonction strcat:

La fonction `strcat(<s>, <t>)` ajoute la chaîne `<t>` à la fin de `<s>`.

```
#include <stdio.h>
#include <string.h>
int main () {
    char chaine1[20]="Bonjour ";
    char chaine2[20]="Edouard";
    strcat(chaine1,chaine2); /* ajoute la chaîne2 à la fin de chaine1 */
    printf("%s\n",chaine1);
    return 0;
}
```

## La fonction strcmp:

La fonction `strcmp(<s>, <t>)` compare les chaînes de caractères `<s>` et `<t>` de manière lexicographique et fournit un résultat :

- nul (0) si `<s>` est égale à `<t>`
- négatif si `<s>` précède `<t>`. Par exemple, `strcmp("AAAA","BBBB")` renverrait -1
- positif si `<s>` suit `<t>`. Par exemple, `strcmp("BBBB","AAAA")` renverrait +1

## Les fonctions sprintf et sscanf:

`sprintf(<chaîne cible>,<chaîne de formatage>,<expr1>,<expr2>,...)`

La fonction `sprintf` renvoie une valeur négative en cas d'erreur et le nombre de caractères stockés dans la chaîne cible sinon.

Dans cet exemple, la fonction va convertir l'entier `i` en chaîne de caractères et la stocker dans la variable `s`. À l'arrivée, `s` contiendra « 15 ».

```
char s[200];
int i=15;
int code;
code=sprintf(s,"%d",i);
```

La fonction `sscanf` fait le contraire.

```
char s[]="12.5 12.3 11.6";
float a,b,c;
int code;
code=sscanf(s,"%f%f%f",&a,&b,&c);
```

Les variables numériques `a`, `b` et `c` contiendront respectivement : 12.5 12.3 et 11.6.

En cas d'erreur, `sscanf` renvoie une valeur négative. S'il n'y a pas eu d'erreur, c'est le nombre de variables affectées qui est renvoyé.