

2 Tableaux & chaînes de caractères

2.1 Tableaux

2.1.1 Définition

Un tableau est un ensemble d'éléments consécutifs. Celui-ci peut être constitué de plusieurs lignes et colonnes. Nous n'utiliserons dans un premier temps que les tableaux à une seule ligne.

Exemple de tableau de caractères

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

Les cases d'un tableau sont numérotées à partir de 0.

2.1.2 Déclaration

☺ Un tableau se déclare de la manière suivante :
<type> <nom du tableau> [<taille du tableau>];

Exemples

Déclaration d'un tableau de 10 caractères.
char tab_char [10];

Déclaration d'un tableau de 10 nombres.
int tab_int [10];

☺ Un tableau à plusieurs lignes se déclare de la façon suivante :
<type> <nom du tableau> [<taille 1^{ère} dimension 1>][<taille 2^{nde} dimension>] ...;

Exemple

int table [5] [5]; représente un tableau d'entiers de 5 lignes * 5 colonnes.

2.1.3 Utilisation

On accède à un tableau en l'appelant par son nom et son numéro de case.

Exemple

Nom : tab_char Case : 3
tab_char [3] = 'C';

Nom : tab_int Case : 6
tab_int [6] = 10;

Attention : ⚠* Aucune limite n'est fixée sur un tableau ⚠*,
il vous est donc possible d'écrire à l'extérieur de votre tableau donc chez le voisin ...
c'est l'un des bugs le plus courant de l'informatique

2.1.4 Cas spécifique des chaînes de caractères

Les chaînes de caractères sont des tableaux de caractères suivis d'un 0 binaire (ne pas confondre avec le caractère 0, nous parlons ici du code ascii) qui est considéré lui aussi comme un caractère. Une chaîne s'écrit donc : chaîne + 0.

Exemple

Eric s'écrit dans un tableau de 5 caractères de la façon suivante E r i c 0.

E	r	i	c	0
0	1	2	3	4

2.1.5 Déclaration d'une chaîne de caractères

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 copain ...

Exemple

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

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

D'autre part, il est possible de déclarer une chaîne de caractères sans en spécifier la longueur de départ de la façon suivante :

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

C'est pratique dans la condition où l'on ne doit pas la réutiliser (on ne connaît sa taille que par son contenu).

2.1.6 Ecriture dans une chaîne de caractères

La première méthode permet de déclarer une chaîne mais non de l'initialiser. Pour se faire, une méthode consiste à utiliser la fonction `sprintf` de la façon suivante :

```
sprintf (<variable de type chaîne de caractères>, "%s", "<valeur d'init.>");
```

Exemple

```
sprintf (m_chaine, "%s", "Eric");
```

remplira m_chaine par Eric0.
(Le 0 n'est là que pour vous rappeler que l'on a besoin du 0 pour terminer la chaîne)

2.1.7 Affichage d'une chaîne de caractères

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

Exemple

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

2.1.8 Longueur d'un chaîne

La longueur d'une chaîne de caractères s'obtient par la fonction `strlen`. Le 0 de fin de chaîne n'est pas compté dans cette longueur.

Exemple

```
char ch [] = "toto" ;  
...  
printf ("La longueur de %s est : %d", ch, strlen (ch)) ;
```

Affichera 4 à l'écran.

2.1.9 Exercices

En utilisant une boucle (for),

≡ Remplissez un tableau de 10 caractères avec les lettres de l'alphabet en commençant par A (code ascii 65).

≡ Faites afficher la chaîne de caractères ainsi obtenue (n'oubliez pas de rajouter le 0).

≡ Faites afficher chaque caractère du tableau sous la forme "Caractère n° 0 : A".

Rappel :

Je peux écrire `Tab_car [i] = code_ascii;` où `code_ascii` est un entier représentant le code Ascii du caractère désigné.

Aide :

Pour faire afficher 1 caractère, on utilisera une syntaxe du style

```
int pos /* Position dans le tableau */  
printf ("Caractère n° %d : %c",pos, Tab_car [pos]);
```

Un élément de tableau peut être assigné à la valeur 0 de la façon suivant :

```
Tab_car [la bonne position] = 0; .
```

2.2 Gets : Saisie d'une chaîne de caractères

2.2.1 Description

La fonction **gets** permet de saisir une chaîne de caractère validée par un Retour Chariot.

Attention, bien que cette chaîne nécessite d'être validée par un retour chariot, celui-ci n'est pas enregistré dans le tableau de caractères.

2.2.2 Exemple d'utilisation

```
#include <stdio.h>  
void 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 de caractères saisie est : \n%s\n", line );  
    printf( "Notons qu'il n'y a qu'un retour chariot." );  
}
```

Exemple d'exécution

```
Saisissez une chaîne de caractère :  
Bonjour !
```

La chaîne de caractère saisie était :
Bonjour !

Notons qu'il n'y a qu'un retour chariot.

2.2.3 Passage d'une chaîne de caractères en paramètres

Il est bien agréable de pouvoir utiliser cette fonction mais son efficacité est très limitée. Nous allons donc créer notre propre fonction de saisie. Pour cela, il est nécessaire de passer en paramètre une chaîne de caractères.

Ceci s'effectue de la façon suivante :

```
int ma_saisie (char* chaine)
{
    Faire ce qu'il faut ...

    return (0);
}

void main ()
{
    char ma_chaine [30];

    ma_saisie (ma_chaine);

    return ();
}
```

On ne passe pas ici par le caractère **&** car en fait un tableau est une adresse mémoire sur une suite de cases. Par contre, on peut passer par **&** en faisant `ma_saisie (&chaine [0])`, `&chaine [0]` représentant l'adresse mémoire de la première case mémoire du tableau C'est un peu compliqué mais il suffit de se dire, je passe un tableau donc un ensemble de cases mémoire, donc je ne mets pas de **&**. Je transmets une case donc un élément d'un type comme un autre donc je mets le **&**.

2.3 Bug en chaînes ...

.Reprenez l'exercice 2.1.7 et mettez 12 caractères dans votre tableau de 10 cases.
.Essayez avec 80 caractères.

On peut facilement dire que le cas de dépassement de la taille d'un tableau est le bug le plus fréquent rencontré en programmation C alors prenez gare ...

N.B. Ce problème n'existe pas sous Turbo Pascal mais on ne peut pas concilier sécurité et rapidité ce qui fait que le C est de loin plus rapide que Turbo Pascal. Il suffit de s'imaginer qu'il est nécessaire d'effectuer un test à chaque assignation d'un élément du tableau !!!

2.4 Exercices

Réaliser votre propre fonction de saisie de chaîne de caractères ...

≡ Paramètres d'entrée : position x, y, chaîne de caractères, nombre de caractères maximum.

Algorithme :

```
/* Amorçage de la boucle */  
car = saisie d'un caractère  
tant que  
    (car est différent d'un retour chariot (0x0D)) ET  
    (nombre de caractères saisis < nombre de caractères maximum))  
faire  
    ajouter car dans le tableau  
    afficher car à l'écran  
    mettre à jour les positions dans le tableau et sur l'écran  
    car = saisie_car  
fin du tant que  
ajouter le 0 binaire de fin de chaîne.
```

Ajouter à votre fonction de saisie de caractères la gestion des touches backspace (code Ascii 8) et flèche gauche (code Ascii 0 suivi du code Ascii 75). Ces touches supprimeront le caractère qui précède en l'effaçant.

Pour vous aider, reportez-vous au chapitre 4 du cours Niveau 1.

Correction des exercices du chapitre 2

⌈ **Boucle for**

En utilisant une boucle (for),

Remplissez un tableau de 10 caractères avec les lettres de l'alphabet en commençant par A (code ascii 65).

Faites afficher la chaîne de caractères ainsi obtenue (n'oubliez pas de rajouter le 0).

Faites afficher chaque caractère du tableau sous la forme "Caractère n° 0 : A".

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

void main ()
{
    char tableau [11];      /* 10 caractères + 0 de fin de chaîne */
    int i=0; /* compteur */

    clrscr ();

    /* Remplissage du tableau avec les caractères */
    for (i=0; i<10; i++)
        tableau [i] = 65 + i;

    /* Ajout du 0 de fin de chaine */
    tableau [10] = 0;

    /* Affichage de la chaîne */
    printf ("Tableau : %s\n",tableau);

    /* Saut d'une autre ligne */
    printf ("\n");

    /* Affichage de chacun des caractères */
    for (i=0; i<10; i++)
    {
        printf ("Caractère n°%d : %c\n",i,tableau [i]);
    }

    /* Attente de la saisie d'une touche */
    getch ();
}
```

⌈ **Fonction de saisie (1^{ère} partie)**

Réaliser votre propre fonction de saisie de chaîne de caractère ...

Paramètres d'entrée : position x, y, chaîne de caractères, nombre de caractères maximum.

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

void saisie (int x, int y, char *chaine, int max_car)
{
```

```
char car = ' ';
int nb_car = 0;

gotoxy (x,y);
car = getch ();

while ((car != 0x0D) && (nb_car < max_car))
{
    /* Ajout du caractère dans le tableau */
    chaine [nb_car] = car;

    /* Affichage ... l'écran */
    printf ("%c",chaine [nb_car]);

    /* Mise ... jour de la position dans le tableau */
    nb_car ++;

    /* Mise à jour de la position ... l'écran */
    x++;
    gotoxy (x,y);

    /* Saisie d'un nouveau caractère */
    car = getch ();
}
chaine [nb_car] = 0;
}

void main ()
{
    char chaine [11];
    char message [] = "Votre saisie : ";

    clrscr ();

    gotoxy (1,3);
    printf ("%s", message);

    /* 10 : Laisser de la place pour le 0 binaire de fin de chaîne */
    saisie (strlen (message),3,chaine,10);

    /* Affichage du résultat */
    gotoxy (1,5);
    textcolor (GREEN);
    cprintf ("Votre saisie : %s",chaine);

    /* Attente pour visualisation */
    getch ();
}
```

(Fonction de saisie (2^{nde} partie)

Ajouter à votre fonction de saisie de caractères la gestion des touches backspace (code Ascii) et flèche gauche (code Ascii). Ces touches supprimeront le caractère qui précède en l'effaçant.

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

void beep ()
{
```

```
    sound (1000);
    delay (100);
    nosound ();
}

void saisie (int x, int y, char *chaine, int max_car)
{
    char car = ' ';
    int nb_car = 0;

    gotoxy (x,y);
    car = getch ();

    while ((car != 0x0D) && (nb_car < max_car))
    {
        switch (car)
        {
            case 0:
            {
                car = getch ();

                /* nb_car = 0 : d, but de la chaine */
                if ((car == 75) && (nb_car > 0))
                {
                    x--;
                    nb_car--;
                    gotoxy (x,y);
                    printf (" ");
                    /* printf avance le curseur de 1 position */

                    /* Réaffichage du curseur à la bonne position */
                    gotoxy (x,y);
                }
                else
                {
                    beep ();
                }
            }
            break;

            case 8:
            {
                if (nb_car > 0)
                {
                    x--;
                    nb_car--;
                    gotoxy (x,y);

                    printf (" ");
                    /* printf avance le curseur de 1 position */

                    /* Réaffichage du curseur à la bonne position */
                    gotoxy (x,y);
                }
                else
                    beep ();
            }
            break;

            default:
            {
```

*Initiation au langage C.
Niveau 2.*

```
        /* Ajout du caractère dans le tableau */
        chaine [nb_car] = car;

        /* Affichage à l'écran */
        printf ("%c",chaine [nb_car]);

        /* Mise à jour de la position dans le tableau */
        nb_car ++;

        /* Mise à jour de la position à l'écran */
        x++;
        gotoxy (x,y);
    }
    break;

} /* Fin du switch case */

/* Saisie d'un nouveau caractère */
car = getch ();
}

/* Ajout du 0 binaire de fin de chaîne */
chaine [nb_car] = 0;
}

void main ()
{
    char chaine [11];
    char message [] = "Votre saisie : ";

    clrscr ();

    gotoxy (1,3);
    printf ("%s", message);

    /* 10 : Laisser de la place pour le 0 binaire de fin de chaîne */
    saisie (strlen (message)+1,3,chaine,10);

    /* Affichage du résultat */
    gotoxy (1,5);
    textcolor (GREEN);
    cprintf ("Votre saisie : %s",chaine);

    /* Attente pour visualisation */
    getch ();
}
```