

## 1 Les chaînes de caractères

Le langage C fait la distinction entre tableau de caractères et chaîne de caractères :

Une chaîne est un tableau de caractères dont le dernier élément significatif est le caractère `'\0'` (le caractère NUL de code ASCII 0).

C'est une convention très pratique car elle évite de devoir retenir le nombre de caractères de la chaîne.

- Déclaration et initialisation :

Comme tout tableau, un tableau de caractères peut être initialisé par une liste de constantes caractères.

✓ **Exemple 1**     `char ch[3] = {'a', 'b', 'c'}; /* Le compilateur ne complète pas avec le caractère NUL */`

Un tableau de caractères peut être initialisé par une chaîne littérale.

✓ **Exemple 2**     `char ch[8] = "exemple";`

Le compilateur complète toute chaîne littérale avec le caractère NUL, il faut donc que le tableau ait au moins une case de plus que le nombre de caractères de la chaîne littérale.

Il est admissible que la taille déclarée pour le tableau soit supérieure à la taille de la chaîne littérale.

✓ **Exemple 3**     `char ch[100] = "Exemple"; /* dans ce cas, seuls les 8 premiers caractères de ch seront initialisés. */`

On peut ne pas indiquer la taille du tableau et dans ce cas, le compilateur donne la taille adéquate au tableau (sans oublier le NUL).

✓ **Exemple 4**

```
char ch[] = "ch aura 22 caractères" ;  
char ch[] = "Une longue chaine peut être écrite sur plusieurs lignes à l'éditeur "  
           "ainsi, sans que cela ne lui pose de problème : le compilateur "  
           "concatène !" ;
```

Si on donne au tableau une taille égale au nombre de caractères de la chaîne, le compilateur ne peut ajouter le caractère NUL de la fin de chaîne.

✓ **Exemple 5**     `char ville[8] = "bordeaux" ; /*ceci n'est pas une chaîne mais seulement un tableau de caractères*/`

Il faut donc différencier la longueur maximale de la chaîne (c'est à dire la dimension donnée lors de la définition) de sa longueur réelle (c'est à dire le nombre de caractères jusqu'au premier `'\0'` rencontré) :

✓ **Exemple 6** *Vous pouvez initialiser la chaîne :*

- `char ch[20] = "bonjour" ;`
- `char ch[20] = {'b','o','n','j','o','u','r','\0'} ;`
- `char ch2[] = "bonjour" ; /* ch2 fait 8 caractères exactement */`

• Modification d'une chaîne, affectation :

```
char chaine1[] = "bonjour" ; /* dimensionne chaine1 à 8 caractères */
```

Pour changer la valeur de `chaine1`, il est nécessaire de modifier un à un les caractères composant la chaîne "bonjour", par exemple : `chaine1[0]='B' ;`

Comme pour tout tableau, pas d'affectation globale possible : `chaine1 = "Hello" ;` est erroné. Plutôt que de travailler directement sur chaque case du tableau de la chaîne à modifier on peut utiliser la fonction `sprintf` de la bibliothèque `string.h`

Ecriture dans une chaîne : tout comme `printf` écrit à l'écran, `sprintf` écrit dans une chaîne :

✓ **Exemple 7** `char chaine[100];`

```
int age = 15;
```

```
sprintf(chaine, "Tu as %d ans!", age); /* On écrit "Tu as 15 ans!" dans chaine */  
printf("%s", chaine); /* On affiche chaine */
```

• Saisie et affichage de chaînes :

• Les fonctions `scanf` et `printf` :

- Le format chaîne de caractères est indiqué aux fonctions `scanf` et `printf` par `%s`

- La fonction `scanf` remplace le premier séparateur (espace , tabulation, passage à la ligne) par `\0` ; elle est donc inefficace dans le cas de plusieurs mots
- La fonction `fgets` : (`<stdio.h>`) elle permet la lecture d'une chaîne contenant des espaces et des tabulations. La lecture ne s'arrête qu'à la réception d'un retour-chariot, lequel est remplacé par le caractère `'\0'`

Utilisation :

```
char chaine[20];
fgets (chaine,taille,stdin); //taille <= 19
```

- chaîne est un tableau de caractères.
  - taille est le nombre de caractères de chaîne.
- La fonction `fgets` rend `NULL` en cas d'erreur.

- Les fonctions de manipulation de chaînes : Les prototypes des fonctions manipulant les chaînes se trouvent principalement dans `string.h`. Les fonctions travaillent sur l'adresse des chaînes ; elles renvoient en général `NULL` en cas d'échec. Quelques unes des principales sont introduites ici :

1. Longueur de la chaîne : `strlen(chaîne)`

✓ **Exemple 8** `int lg=strlen("bonjour");` — — — > 7  
`char ch[]="coucou"; lg=strlen(ch);` — — — > 6  
`lg=strlen(" aa aa ");` — — — > 7

2. Concaténation de chaînes : `strcat( but, source)`

✓ **Exemple 9** `char ch1[50] = "Comment";` //il faut un nombre de cases  $\geq \text{strlen}(ch1)+\text{strlen}(ch2)+1$   
`char ch2[]=" es-tu venu ?";`  
`strcat(ch1,ch2);` `ch1` — — — > "Comment es-tu venu ?"

3. Comparaison de chaînes : `strcmp( ch1, ch2)`

compare 2 chaînes, en utilisant l'ordre des codes de caractère (ASCII).

- positif si `ch1 > ch2`
- nul si `ch1 = ch2`
- négatif si `ch1 < ch2`

✓ **Exemple 10** `int cmp=strcmp("bonjour", "madame");` — — — > négatif  
`cmp=strcmp("au revoir", "madame");` — — — > positif

4. Copie de chaînes : `strcpy( dest, source)`

copy source dans dest : dest doit être assez grand pour pouvoir contenir source ; ni le compilateur ni la fonction ne font de vérification, c'est au programmeur de s'assurer

que la taille de dest est suffisante.

```
✓ Exemple 11 char ch1[20] = "123456789" ;  
char ch3[] = "coucou" ;  
strcpy(ch1, ch2) ; --- > ch1 = "coucou456789"  
strcpy(ch1, ch3) ; --- > ch1 = "coucou" // on a copié aussi le caractère NUL
```

## 2 Exercices :

◇ Exercice 1 : Ecrire une fonction booléenne qui détermine si 2 chaînes de caractères sont différentes.(on pourra vérifier le résultat avec `strcmp`).

◇ Exercice 2 : Ecrire une fonction qui supprime les espaces (`'\t', '\n', ' '`) présents au début d'une chaîne.

– la chaîne pourra être lue avec `gets`

– on pourra utiliser la fonction `isspace` de la bibliothèque `ctype.h`

◇ Exercice 3 : Ecrire une fonction qui prend une chaîne (en paramètre d'entrée) et affiche les uns sous les autres chacun de ses caractères, avec, sur la même ligne, le code ASCII du caractère. Tester avec divers caractères : lettres minuscules, lettres accentuées, symboles de ponctuation...

◇ Exercice 4 :

1. Ecrire une fonction qui compte le nombre de voyelles majuscules d'une chaîne de caractères.
2. Ecrire une procédure qui transforme en majuscules toutes les lettres d'une chaîne de caractères.
3. Ecrire une procédure qui modifie une chaîne de caractères en supprimant toutes les occurrences de la lettre minuscule 'a'

Dans les 3 questions précédentes, pas de saisie ni d'affichage (**aucun `scanf` ni `printf`** ne doit être utilisé).

4. Tester ces 3 sous-programmes en affichant les résultats pour la chaîne "taRatAta".

◇ Exercice 5 : Ecrire une fonction qui prend pour paramètre une chaîne en minuscules et modifie cette chaîne en décalant chaque caractère de 13 caractères dans l'ordre alphabétique en supposant qu'après la lettre 'z' on revient à la lettre 'a'. Par exemple la chaîne « alphonse » devient « nycubafr ». On pourra utiliser le fait que les caractères alphabétiques sont codés (code ASCII) par des entiers consécutifs. (Pour les tests, utiliser le site <http://www.rot13.com/>)

◇ Exercice 6 : Ecrire une fonction qui pour un verbe régulier du premier groupe détermine sa forme au présent de l'indicatif ; par exemple :

<i>je programme</i>	<i>j'envoie</i>	<i>je mange</i>
<i>tu programmes</i>	<i>tu envoies</i>	<i>tu manges</i>
<i>il programme</i>	<i>il envoie</i>	<i>il mange</i>
<i>nous programmons</i>	<i>nous envoyons</i>	<i>nous mangeons</i>
<i>vous programmez</i>	<i>vous envoyez</i>	<i>vous mangez</i>
<i>ils programment</i>	<i>ils envoient</i>	<i>ils mangent</i>

Prenez garde à bien dimensionner le tableau. Commencer par une version basique que vous ferez évoluer pour traiter les cas particuliers : verbe en yer, verbe en ger, voyelle en debut de verbe.

◇ Exercice 7 : Créer un type structuré T\_etudiant contenant les champs :

- nom : 19 caractères
- prenom : 19 caractères
- adresse : 39 caractères
- notes : tableau de 5 entiers

Quelle est la taille d'une variable de ce type? Utiliser l'opérateur **sizeof** Ecrire un programme dans lequel vous créez un tableau statique d'étudiants : grâce au nouveau type : T\_etudiant tab[MAX\_ETUDIANTS\_STATIQUE]; Constatez l'existence d'une valeur limite de la constante MAX\_ETUDIANTS\_STATIQUE pour laquelle l'allocation se produit normalement. Quelle est l'origine de cette limitation?

Déterminez par dichotomie cette valeur limite.