

**TP 5 : Les chaines de caractères**

**Durée : 3h**

# Objectif

Le but de ce TP est de se familiariser avec les chaines de caractère en C.

**Introduction**

Une chaîne de caractères (appelée *string* en anglais) est une succession de n caractères avec n compris entre 0 et 255.

En **langage C**, une chaîne de caractères est déclarée comme un tableau de caractères dont le dernier élément est le caractère nul '\0' (son code ASCII est 0). Ce caractère est un caractère de contrôle (donc non affichable) qui permet d'indiquer une fin de chaîne de caractères.

Par exemple, en mémoire, la chaîne "Bonjour" est représentée de la manière suivante:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **B** | **o** | **n** | **j** | **o** | **u** | **r** | **\0** |

1. **Manipulation 5.1 : Manipulation de chaine de caractères**

**Déclaration et accès**

Pour définir une chaîne de caractères en **langage C**, il suffit de déclarer un tableau de caractères.

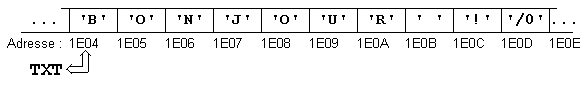
Le nombre maximum de caractères que comportera la chaîne sera égal au nombre d'éléments du tableau moins un (réservé au caractère de fin de chaîne).

Le nom d’une chaîne est le représentant de l’adresse du premier caractère de la chaîne.

Pour mémoriser une variable qui doit être capable de contenir un texte de N caractères, nous avons besoin de N+1 octets en mémoire:

**char TXT[10] = "BONJOUR !";**

Exemple:



## Déclaration d’une Chaîne

**char <Nom\_Variable> [<Longueur+1>] ;**

Syntaxe:

Exemple:

**char nom[20] ;**

**char prenom[20] ;**

## Initialisation d’une chaîne

Pour initialiser une chaîne de caractères, il suffit de remplir les cases du tableau avec des caractères, sachant que celui-ci devra obligatoirement contenir le caractère de fin de chaîne '\0'.

* **Lors de la déclaration :** Comme pour les tableaux, on peut initialiser un chaîne par l’indication de la liste des composantes entre {}. Chaque composante entre ‘  ’.

Exemple:

**char mot [4] = {‘ B ’, ’ O ’, ‘ N ’, ‘ \0 ’};**

* **Affectation globale :** On peut aussi initialiser une chaîne de caractères directement par une chaîne constante.

Exemple:

**char mot [] = "BON"; /\*réservation automatique de 4 octets\*/**

* **Affectation par caractère :** Remplir la chaîne caractère par caractère.

Exemple:

**#include <stdio.h>**

**void main()**

**{ char mot[3+1];**

**mot [0]= 'B';**

**mot [1]= 'O';**

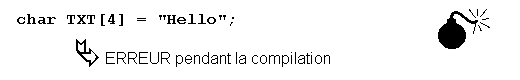
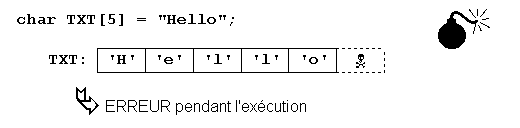
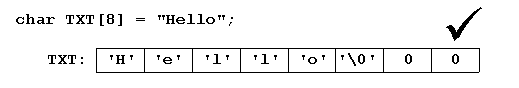
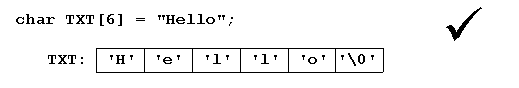
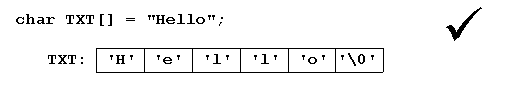
**mot [2]= 'N';**

**mot [3]= '\0';**

**}**

Lors de l'initialisation par **[]**, l'ordinateur réserve automatiquement le nombre d'octets nécessaires pour la chaîne, c.-à-d.: le nombre de caractères **+ 1** (ici: 6 octets). Nous pouvons aussi indiquer explicitement le nombre d'octets à réserver, si celui-ci est supérieur ou égal à la longueur de la chaîne d'initialisation.

Exemple:



## Les chaînes de caractères constants

* Les chaînes de caractères constantes sont indiquée entre ".
* La chaîne vide c’est : ""
* On peut utiliser toutes les séquences d’échappement dans une chaîne de caractère.

Exemple:

**"Ce \n texte\n est réparti en 3 lignes "**

* Pour représenter le symbole " dans une chaîne de caractères il faut le précéder par \

Exemple:

**"L’ordinateur \"dit \" "**

**Observation :**

Pour la mémorisation de la chaîne de caractères "Hello", C a besoin de **six (!!)** octets.

|  |  |  |
| --- | --- | --- |
|  | **'x'** | est un *caractère constant*, qui a une valeur numérique: |
|  |  | P.ex: 'x' a la valeur 120 dans le code ASCII. |
|  | **"x"** | est un *tableau de caractères* qui contient deux caractères: |
|  |  | la lettre 'x' et le caractère NUL: '**\0**' |
|  | **'x'** | est codé dans un octet |
|  | **"x"** | est codé dans deux octets |

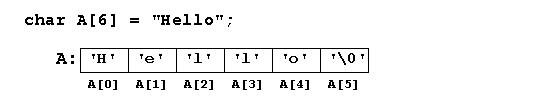
## L’accès à une chaîne

L’accès à une chaîne est pareil que les autres types de tableaux.

**mot [0] est le caractère ‘ B ’**

Exemple:

**mot [1] = ‘E ’; /\* mot devient  "BEN" \*/**



**Ecriture d’une chaine de caractères**

La fonction prédéfinie **printf** permet d’afficher une chaîne de caractères.

**printf ("<format>", NomChaine1, NomChaine2, …);**

* **<format>** : **%s**

Exemple:

**char Matiere [ ] = "Informatique";**

**printf("(%.8s)", Matiere); /\*Affiche (Informatique)\*/**

La fonction prédéfinie **puts** permet d’écrire une chaîne constante ou le contenu d’une variable chaîne dans une ligne isolée.

**puts (<chaîne>);**

**char NOM [ ] = "Toto";**

Exemple:

**puts(NOM);**

**Ou bien**

**puts ("Toto");**

**Lecture d’une chaine de caractères**

La fonction prédéfinie **scanf** permet de lire une chaîne de caractères

**scanf ("<format>", NomChaine1, NomChaine2, );**

* **<format>** : **%s**

Exemple:

**char prenom [50];**

**printf ("Quel est votre prénom ?: ");**

**scanf ("%s", prenom);**

La fonction prédéfinie **gets** permet delire une ligne se terminant par un retour à la ligne et la copie à l’adresse indiquée par le nom de la chaîne. Le retour à la ligne finale est remplacé par ‘\0’

* **gets (<chaîne>);**

Exemple:

**char prenom [50];**

**printf ("Quel est votre prénom ?: ");**

**gets(prenom);**

**Exercice 5.1.1**

Ecrire un programme C qui affiche pour chaque voyelle (a, e, i, o, u, y) d’un mot son nombre d’apparition.

**Exemple:**

Pour le mot « **informatique**» la fonction affiche :

La voyelle « i » apparaît 2 fois

La voyelle « e » apparaît 1 fois

La voyelle « o » apparaît 1 fois

La voyelle « a » apparaît 1 fois

La voyelle « u » apparaît 1 fois

Ecrire un programme en langage C qui appelle cette fonction pour une chaîne (mot) introduite au clavier.

**Exercice 5.1.2**

Ecrire un programme C qui permet de tester si une chaine lue au clavier est palindrome ou non. Une chaine est dite palindrome si elle est lue de droite à gauche et de gauche à droite de la même manière comme AZIZA

**Exercice 5.1.3**

Ecrire un programme C qui lit une chaine de caractere contenant des parentheses ouvrantes et fermantes et doit aficher si le parenthesge est coherent ou non.

**Manipulation 5.2 : Fonctions de manipulation d’une chaine de caractères**

De nombreuses fonctions de traitement des chaînes de caractères sont directement fournies. En **langage C**, ces fonctions se trouvent dans le fichier d'en-tête <string.h> et d’autres dans <stdlib.h>

**Fonctions de <*string.h>* :**

Le fichier <string.h> contient les prototypes des fonctions permettant de simplifier l'utilisation et la manipulation de chaînes (environ une quarantaine).

En voici quelques unes :

* **Strlen :** retourne le nombre de caractères d’une chaîne, sans compter le caractère '\0'.

**strlen(<chaîne>);**

Exemple:

**int lg;**

**lg = strlen ("Toto"); /\* lg = 4 \*/**

* **Strcpy :** permet de recopier en général une chaîne de caractères dans une variable de type chaîne de caractères.

**strcpy (<ch1>, <ch2>); /\* copie ch2 dans ch1 \*/**

Exemple:

**strcpy (ch1, "Toto");**

**ch1 = "Toto" ;**

* **Strcat :** permet de concaténer deux chaînes de caractères.

**strcat(<ch1>,<ch2>); /\* ajoute ch2 à la fin de ch1 \*/**

Exemple:

**strcpy (ch1, "Bon");**

**strcpy (ch2, "jour");**

**strcat (ch1, ch2); /\* (ch1: Bonjour)\*/**

* **Strcmp :** permet de comparer deux chaînes de caractères

Syntaxe:

**Strcmp(<ch1>,<ch2>);**

**compare ch1 et ch2 et rend la valeur :**

1. **: si elles sont identiques**
2. **-1 : si <ch1> avant <ch2>**
3. **1 : si <ch1> après <ch2>**

**strcpy (ch1, "Bon");**

Exemple:

**strcpy (ch2, "Ben");**

**if (strcmp (ch1, ch2))**

**printf ("Les 2 chaînes sont différentes");**

**else**

**printf ("Les 2 chaînes sont égales");**

**Fonctions de <*string.h>* :**

La bibliothèque *<stdlib>* contient des déclarations de fonctions pour la conversion de nombres en chaînes de caractères et vice-versa.

**Chaîne --> Nombre :**

Les trois fonctions définies ci-dessous correspondent au standard ANSI-C et sont portables. Le symbole <s> peut être remplacé par :

* une chaîne de caractères constante
* le nom d'une variable déclarée comme tableau de **char**
* un pointeur sur **char** (voir chapitre 9)

**Conversion de chaînes de caractères en nombres** :

|  |  |  |
| --- | --- | --- |
|  | **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** (!) |

**Règles générales pour la conversion:**

* Les espaces au début d'une chaîne sont ignorés
* Il n'y a pas de contrôle du domaine de la cible
* La conversion s'arrête au premier caractère non convertible
* Pour une chaîne non convertible, les fonctions retournent zéro

**Nombre --> Chaîne :**

Le standard ANSI-C ne contient pas de fonctions pour convertir des nombres en chaînes de caractères. Si on se limite aux systèmes fonctionnant sous DOS, on peut quand même utiliser les fonctions **itoa**, **ltoa** et **ultoa** qui convertissent des entiers en chaînes de caractères.

**Conversion de nombres en chaînes de caractères**

**- itoa (<n\_int>, <s>, <b>)**

**- ltoa (<n\_long>, <s>, <b>)**

**- ultoa (<n\_uns\_long>, <s>, <b>)**

Chacune de ces trois procédures convertit son premier argument en une chaîne de caractères qui sera ensuite attribuée à <s>. La conversion se fait dans la base <b>.

|  |  |  |
| --- | --- | --- |
|  | **<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, longueur maximale de la chaîne: 17 resp. 33 byte |
|  | **<b>** | est la base pour la conversion (2 ... 36) |
|  | **Exercice 5.2.1**  Ecrire un programme C qui lit 3 chaines de caractères et qui les afficher dans une même chaine commençant par la chaine la plus petite au plus grande.  **Exercice 5.2.2** |  |
|  | Un système de cryptographie consiste à décaler les lettres d’un message pour le rendre illisible. Ainsi, les **A** deviennent des **B**, les **B** des **C**, les **C** des **D** etc.  Ecrire un algorithme qui permet de coder une chaine de caractères entrée par l’utilisateur (sans espaces) selon ce principe, (le codage doit s’effectuer au niveau de la variable stockant la chaine, et pas seulement à l’écran).  Pour le cas de la vingt-sixième lettre (**Z**), elle doit être codée par la première lettre (**A**).  ***Indications:***  *Utiliser les variables de types chaines de caractères suivantes:*   * alpha ″ ABCDEFGHIJKLMNOPQRSTUVWXYZ″, qui va contenir la chaine représentant les lettres de l’alphabet. * **phrase**: une chaine de caractères qui va contenir la phrase à coder. * **code**: une chaine de caractères qui va contenir la nouvelle phrase codée. * **lettre** : un caractère qui va contenir la lettre courante de la phrase à coder   *Utiliser les fonctions prédéfinies sur les chaines de caractères suivantes:*  ***Exemple:***  La variable ***phrase*** contient "AZERTYCOURSALGO″, la variable résultat ***code*** contient "BAFSUZDPVSTBMHP". |  |

1. **Manipulation 5.3 : les tableaux de chaines de caractères**

Un tableau de chaînes de caractères correspond à un tableau à deux dimensions de type char, ou chaque ligne contient une chaîne de caractères.

**Déclaration:**

**Char < Nom Tableau> [<NbChaines>][<longueur+1];**

Exemple:

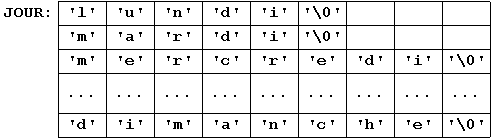
**char jours [7][9] ; /\* réserve 7 mots de longueur 9 caractères dont 8 significatifs \*/**

**Initialisation:**

Lors de la déclaration on peut initialiser les composantes par des chaînes constantes

**char jour[7][9] = {"lundi","Mardi",’"Mercredi","Jeudi", "Vendredi","Samedi", "Dimanche"};**

Exemple:



Les tableaux de chaînes sont mémorisés ligne par ligne. La variable JOUR aura donc besoin de 7\*9\*1 = 63 octets en mémoire.

**Accès**

Pour accéder à une chaîne, il faut indiquer la ligne correspondante

Exemple:

**Char jour[7][9] = {"lundi","Mardi",’"Mercredi","Jeudi", "Vendredi","Samedi", "Dimanche"};**

**Printf ("c'est %s\n" , jour [0] ); /\* c'est lundi \*/**

Pour accéder à un caractère d'une chaîne, il faut indiquer la ligne correspondante et l'indice du caractère dans la chaîne

Exemple:

**Printf ("%c%c\n", jours [0][0],jours [0][1]); /\* lu \*/**

**Affectation**

Utiliser **strcpy** pour affecter une chaîne à une composante d'un tableau de chaînes

**Strcpy ( jours[6],"Sunday"); /\* jours [6]="Sunday"; \*/**



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

Exemple :

La commande:

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

changera le contenu de la *5e* composante du tableau JOUR de "vendredi" en "Friday".

**Exercice 5.3.1**

L’ADN des êtres vivants est composée de 4 types de molécules {A, C, G, T}

Les séquences d’ADN qui existent dans la nature sont les suivantes :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GTG | CAC | CTG | ACT | CCT | GAG | AGG | TCT | GCC |

* Ecrire un programme C qui permet de**:**
  1. Saisir n le nombre de séquences ADN a Saisir.
  2. Remplir la chaine de caractères **adn** par n séquences ADN correctes
  3. Afficher le nombre d’occurrence de la séquence d’ADN **« CAC »**