Les chaînes de caractères

Dans un programme informatique, les chaînes de caractères servent à stocker les informations non numériques comme par exemple une liste de nom de personne ou des adresses.

Il n'existe pas de type spécial *chaîne* ou *string* en C. **Une chaîne de caractères est traitée comme un tableau à une dimension de caractères (vecteur de caractères).** Il existe quand même des notations particulières et une bonne quantité de fonctions spéciales pour le traitement de tableaux de caractères.

Déclaration d’une chaîne

Une chaîne de caractères est un tableau de type char. La déclaration est identique à un tableau normal:

char <nom_chaine> [<dimension>]

La représentation interne d'une chaîne de caractères est terminée par le symbole '\0' (NULL). Ainsi, pour un texte de *n* caractères, nous devons prévoir *n*+1 octets.

Malheureusement, le compilateur C ne contrôle pas si nous avons réservé un octet pour le symbole de fin de chaîne; l'erreur se fera seulement remarquer lors de l'exécution du programme ...

Initialiser une chaîne de caractères

En général, les tableaux sont initialisés par l'indication de la liste des éléments du tableau entre accolades:

```
char MACHAINE[ ] = {'H','e','l','l','o','\0'};
```

Pour le cas spécial des tableaux de caractères, nous pouvons utiliser une initialisation plus confortable en indiquant simplement une chaîne de caractères constante:

```
char MACHAINE[ ] = "Hello";
```

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.

Exemples d'initialisation

char MACHAINE[ ] = "Hello";
char MACHAINE[6] = "Hello";
char MACHAINE[ ] = {'H','e','l','l','o','\0'};
char MACHAINE[8] = "Hello";

par contre:
char MACHAINE[5] = "Hello"; donnera une erreur à l'exécution

char MACHAINE[4] = "Hello"; donnera une erreur à la compilation.

Chaînes constantes et caractères

Attention à la déclaration:

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:
Par exemple: 'x' à 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

A savoir (1)

* Les chaînes de caractères constantes (string literals) sont indiquées entre guillemets. La chaîne de caractères vide est alors: ""

* Dans les chaînes de caractères, nous pouvons utiliser toutes les séquences d'échappement définies comme caractères constants: "Ce \ntexte \nsera réparti sur 3 lignes."

* Le symbole " peut être représenté à l'intérieur d'une chaîne par la séquence d'échappement \".

* Le symbole ' peut être représenté à l'intérieur d'une liste de caractères par la séquence d'échappement \' :

```
{'L','\'','a','s','t','u','c','e','\0'}
```

A savoir (2)

* Plusieurs chaînes de caractères constantes qui sont séparées par des signes d'espacement (espaces, tabulateurs ou interlignes) dans le texte du programme seront réunies en une seule chaîne constante lors de la compilation:
"un " "deux" " trois” sera évalué à "un deux trois"

=> il est possible de définir de très longues chaînes de caractères constantes en utilisant plusieurs lignes dans le texte du programme.

Accéder à une chaîne et à ses éléments

Une chaîne de caractères est une variable comme une autre pour un programme: on y accède en l'appelant par son nom de variable.

Une chaîne est un tableau de caractères: pour accéder a ses éléments on suit la logique d'un tableau.

Exemple:

char A[6] = "Hello";
-> A[0] contient 'H', A[1] contient 'e' … A[5] contient '\0'.

Précédence et opérations

Une chaîne de caractères est une variable : on peut utiliser des opérations logiques et mathématiques.

La précédence des caractères dans l'alphabet d'une machine est dépendante du code de caractères utilisé. Pour le code ASCII, nous pouvons constater l'ordre suivant:
. . . ,0,1,2, ... ,9, . . . ,A,B,C, ... ,Z, . . . ,a,b,c, ... ,z, . . .

Les symboles spéciaux (' ,+ ,- ,/ ,{ ,] , ...) et les lettres accentuées (é ,è ,à ,û , ...) se trouvent répartis autour des trois grands groupes de caractères (chiffres, majuscules, minuscules). Leur précédence ne correspond à aucune règle d'ordre spécifique.

Précédence et opérations

Précédence alphabétique des caractères induit une relation de précédence 'est inférieur à' sur l'ensemble des caractères. (idem pour les autres relations logiques)

Ainsi, on peut dire que '0' est inférieur à 'Z'
et noter '0' < 'Z'

Ceci est possible car dans l'alphabet de la machine, le code du caractère '0' (ASCII: 48) est inférieur au code du caractère 'Z' (ASCII: 90).

Exemples
"ABC" précède "BCD" car 'A'<'B'
"ABC" précède "B" car 'A'<'B'
"Abc" précède "abc" car 'A'<'a'
"ab" précède "abcd" car "" précède "cd"
" ab" précède "ab" car ' '<'a'

Tests logiques

En tenant compte de l'ordre alphabétique des caractères, on peut contrôler le type du caractère (chiffre, majuscule, minuscule).

Exemples

```
if (C>='0' && C<='9') printf("Chiffre\n", C);
if (C>='A' && C<='Z') printf("Majuscule\n", C);
if (C>='a' && C<='z') printf("Minuscule\n", C);
```

Il est facile, de convertir des lettres majuscules dans des minuscules:

```
if (C>='A' && C<='Z') C = C-'A'+'a';
```

ou vice-versa:

```
if (C>='a' && C<='z') C = C-'a'+'A';
```

Tableaux de chaînes

Une chaîne de caractères est un tableau à 1 dimension de caractères.

On peut également définir des tableaux à plusieurs dimensions qui peuvent contenir des mots:

```
char JOUR[7][9] = {“lundi” , ”mardi” , ”mercredi” , ”jeudi” ,
                   ”vendredi”,”samedi”,”dimanche”};
```

et on peut accéder à ces mots en utilisant la syntaxe suivante:

```
int I=2;
printf(“Aujourd’hui nous sommes %s”,JOUR[I]);
```

qui affichera “Aujourd’hui nous sommes mercredi”.

pour accéder à une lettre dans un mot: JOUR[I][j] avec %c pour l’affichage.

Fonctions de bibliothèque

Des fonctions de traitement des chaînes de caractères sont disponibles dans les bibliothèques standards:

<stdio.h> :

scanf, printf en utilisant %s dans le format
attention: scanf prend une adresse en argument (&x), une chaîne de caractères étant un tableau (ie, l'adresse du premier élément), il n'y a pas de &.

puts: puts(MACHAINE); est équivalent à printf("%s\n",TXT);

gets: gets(MACHAINE); lit une ligne jusqu'au retour chariot et remplace le '\n' par '\0' dans l'affectation de la chaîne.

Fonctions de bibliothèque

<string>:

strlen(<s>) fournit la longueur de la chaîne sans compter le '\0' final

strcpy(<s>, <t>) copie <t> vers <s>

strcat(<s>, <t>) ajoute <t> à la fin de <s>

strcmp(<s>, <t>) compare <s> et <t> lexicographiquement et fournit un résultat:

négatif si <s> précède <t>
zéro si <s> est égal à <t>
positif si <s> suit <t>

strncpy(<s>, <t>, <n>) copie au plus <n> caractères de <t> vers <s>

strncat(<s>, <t>, <n>) ajoute au plus <n> caractères de <t> à la fin de <s>

Attention:

La nature de tableau d'une chaîne de caractères (ie, l'adresse en mémoire du premier élément) interdit des affections du type A= "hello" en dehors de la phase d'initialisation.

char A[ ]="Hello"; est correct mais
char A[6]; A= "Hello"; ne l'est pas.

Il faut bien copier la chaîne caractère par caractère ou utiliser la fonction strcpy respectivement strncpy:

strcpy(A, "Hello");

Fonctions de bibliothèque

<stdlib>: conversion 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 (!)

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

Exercices:

Ex 15: calcul de la surface sous une courbe

En utilisant une méthode de type Monte-Carlo (cf calcul de Pi), écrivez un programme C qui permet de calculer la surface sous une courbe y=f(x) entre les points x1 et x2. x1 et x2 seront entrés par l'utilisateur avec un scanf. On considerera que la fonction f(x) est une fonction connue décrite par la déclaration suivante: float f(float x);

Ex 16 : tri direct d'un tableau

écrire un programme qui permet de trier un tableau de nombres réels de façon directe et qui affiche le résultat.