

Chapitre 8

Les Chaînes de caractères

Chaînes de caractères

- Il n'existe pas de type spécial chaîne ou string en C. Une chaîne de caractères est traitée comme un tableau de caractères
- Une chaîne de caractères en C est caractérisée par le fait que le dernier élément vaut le caractère '\0', ceci permet de détecter la fin de la chaîne
- Il existe plusieurs fonctions prédéfinies pour le traitement des chaînes de caractères (ou tableaux de caractères)

Déclaration

- Syntaxe : **char** <NomVariable> [<Longueur>]; //tableau de caractères

Exemple : **char** NOM [15];

- Pour une chaîne de N caractères, on a besoin de N+1 octets en mémoire (le dernier octet est réservé pour le caractère '\0')
- Le nom d'une chaîne de caractères est le représentant de l'adresse du 1^{er} caractère de la chaîne
- On peut aussi manipuler les chaînes de caractères en utilisant des pointeurs (de la même façon qu'un pointeur sur **int** peut contenir l'adresse d'un élément d'un tableau d'entiers, un pointeur sur **char** peut pointer sur les éléments d'un tableau de caractères)

Initialisation

- On peut initialiser une chaîne de caractères à la définition :
 - comme un tableau, par exemple : `char ch[] = {'e','c','o','l','e','\0'}`
 - par une chaîne constante, par exemple : `char ch[] = "école"`
 - en attribuant *l'adresse d'une chaîne de caractères constante* à un pointeur sur char, par exemple : `char *ch = "école"`
- On peut préciser le nombre d'octets à réserver à condition que celui-ci soit supérieur ou égal à la longueur de la chaîne d'initialisation
 - `char ch[6] = "école"` est valide
 - `char ch[4] = "école"` ou `char ch[5] = "école"` provoque une erreur

Traitement des chaînes de caractères

- Le langage C dispose d'un ensemble de bibliothèques qui contiennent des fonctions spéciales pour le traitement de chaînes de caractères
- Les principales bibliothèques sont :
 - La bibliothèque `<stdio.h>`
 - La bibliothèque `<string.h>`
 - La bibliothèque `<stdlib.h>`
- Nous verrons les fonctions les plus utilisées de ces bibliothèques

Fonctions de la bibliothèque <stdio.h>

- **printf()** : permet d'afficher une chaîne de caractères en utilisant le spécificateur de format %s.

Exemple : **char ch[]= " Bonsoir " ;**
printf(" %s ", ch);

- **puts(<chaîne>)** : affiche la chaîne de caractères désignée par <Chaîne> et provoque un retour à la ligne.

Exemple : **char *ch= " Bonsoir " ;**
puts(ch); /*équivalente à printf("%s\n ", ch);*/

Fonctions de la bibliothèque <stdio.h>

- **scanf()** : permet de saisir une chaîne de caractères en utilisant le spécificateur de format %s.

Exemple : **char Nom[15];**
printf("entrez votre nom");
scanf(" %s ", Nom);

Remarque : le nom d'une chaîne de caractères est le représentant de l'adresse du premier caractère de la chaîne, il ne doit pas être précédé de **&**

- **gets(<chaine>)** : lit la chaîne de caractères désignée par <Chaîne>

Exemple : **char phrase[100];**
printf("entrez une phrase");
gets(phrase);

Fonctions de la bibliothèque <string.h>

- **strlen(ch)**: fournit la longueur de la chaîne sans compter le '\0' final

Exemple : **char s[]= " Test";**

printf("%d",strlen(s)); //affiche 4

- **strcat(ch1, ch2)** : ajoute ch2 à la fin de ch1. Le caractère '\0' de ch1 est écrasé par le 1^{er} caractère de ch2

Exemple : **char ch1[20]=" Bonne ", *ch2=" chance ";**

strcat(ch1, ch2) ;

printf(" %s", ch1); // affiche Bonne chance

Fonctions de la bibliothèque <string.h>

- **strcmp(ch1, ch2):** compare ch1 et ch2 lexicographiquement et retourne une valeur : - nul si ch1 et ch2 sont identiques
 - négative si ch1 précède ch2
 - positive si ch1 suit ch2
- **strcpy(ch1, ch2) :** copie ch2 dans ch1 y compris le caractère '\0'

Exemple : **char ch[10];**
 strcpy(ch, " Bonjour ");
 puts(ch); // affiche Bonjour

- **strchr(char *s, char c) :** recherche la 1^{ère} occurrence du caractère c dans la chaîne s et retourne un pointeur sur cette 1^{ère} occurrence si c'est un caractère de s, sinon le pointeur NULL

Fonctions de la bibliothèque <stdlib.h>

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

- **atoi(ch)**: retourne la valeur numérique représentée par ch comme **int**
- **atof(ch)**: retourne la valeur numérique représentée par ch comme **float**
(si aucun caractère n'est valide, ces fonctions retournent 0)

Exemple : **int x, float y;**

```
char *s= " 123 ", ch[]= " 4.56 ";  
x=atoi(s); y=atof(ch); // x=123 et y=4.56
```

- **itoa(int n, char * ch, int b)** : convertit l'entier n en une chaîne de caractères qui sera attribué à ch. La conversion se fait en base b

Exemple : **char ch[30]; int p=18;**

```
itoa(p, ch, 2); // ch= " 10010 "
```