Filière IGE – 2017/2018
Algorithmique & Programmation
Leçon 7 : Chaînes de caractères

Prof. *A. DARGHAM*

Chaînes de caractères

- **Définition**
  - Une **chaîne de caractères** (ou simplement **chaîne**) est une **séquence de caractères ASCII**.
  - C'est en fait :
    - Soit un **tableau de type char** : donne une **chaîne allouée en phase statique**.
    - Soit un **pointeur sur un char** : donne une **chaîne allouée en phase dynamique**.

Chaînes de caractères

- **Déclaration**
  - Comme **tableau de type char** :

```
char str[N]; /* un tableau de char */
```

  - Comme un **pointeur sur un char** :

```
char * str; /* un pointeur sur char */
```

Chaînes de caractères

- **Initialisation**
  - Comme **tableau de type char** :

```
char str1[8] = "bonjour";
char str2[ ] = "bonsoir";
char nom[30] = "jack";
```

  - Comme un **pointeur sur un char** :

```
char * str = "bonjour";
```

Chaînes de caractères

- **Le caractère de fin de chaîne '\0' :**
  - Le **compilateur met à la fin de chaque chaîne** un **caractère spécial** : `'\0'`.
  - C'est le **caractère de fin de chaîne** (**code ASCII 0**).
  - Attention, **ce n'est pas le chiffre** `'0'` (**code ASCII 48**).
  - En conclusion, si l'on veut déclarer une chaîne avec **N caractères significatifs**, il faut ajouter 1 (**N+1**) pour **tenir compte du caractère** `'\0'`.

Chaînes de caractères

- **Le caractère de fin de chaîne '\0' :**

```
#include <stdio.h>
main()
{
   printf("code ASCII de '\\0' : %d\n", '\0');
   printf("code ASCII de '0' : %d\n", '0');
   system("pause");
}
```

Chaînes de caractères

Chaînes de caractères

- **Affichage d'une chaîne :**
  - Avec la fonction **printf** en **utilisant le spécificateur de format %s**.
- **Exemple** :

```
char str[30] = "jack Moore.";
char * ch = "bonjour";
printf("%s\n", str); /* affiche jack Moore. */
printf("%s\n", ch); /* affiche bonjour */
```

Chaînes de caractères

- **Affichage d'une chaîne :**
  - Avec la fonction **puts**.
- **Exemple** :

```
char str[30] = "jack Moore.";
char * ch = "bonjour";
puts(str); /* affiche jack Moore. */
puts(ch); /* affiche bonjour */
```

- **Remarque :**
  - La fonction **puts ajoute automatiquement** un **retour à la ligne**.

Chaînes de caractères

- **Lecture d'une chaîne :**
  - Avec la fonction **scanf** en **utilisant le spécificateur de format %s**.
  - Avec la fonction **gets**.

Chaînes de caractères

- Exemple 1 :

```
#include <stdio.h>
main()
{
   char str[30] = ""; /* la chaine vide */

   printf("Donner une chaine : ");
   scanf("%s", str);
   printf("Votre chaine est : %s\n", str);
   system("pause");
}
```

Chaînes de caractères

Chaînes de caractères

- **Lecture d'une chaîne :**
  - Lorsqu'on lit une chaîne avec **scanf**, la **lecture s'arrête** dès que la **fonction rencontre le premier caractère blanc** (**espace** , **tabulation** ou **retour à la ligne**).
  - Par conséquent, **scanf** est **mauvaise pour lire une chaîne contenant des coupures** (des **caractères blancs**) et ne permet pas de lire une **chaîne vide**.
  - La **solution efficace** pour **lire les chaînes** est la fonction **gets**.

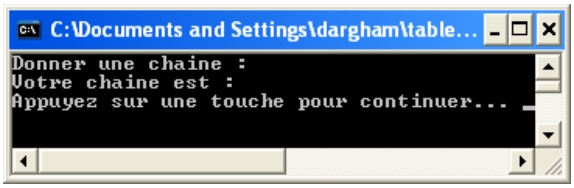
Chaînes de caractères

- **Exemple 2** :

```
#include <stdio.h>
main()
{
   char str[30] = ""; /* la chaine vide */

   printf("Donner une chaine : ");
   gets(str);
   printf("Votre chaine est : %s\n", str);
   system("pause");
}
```

Chaînes de caractères



Chaînes de caractères

- Exemple 3 :

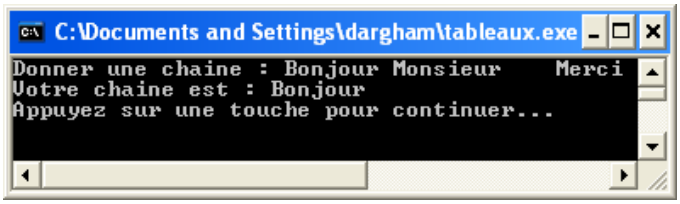
```
#include <stdio.h>
#include <stdlib.h>
main()
{
   char * ch;
   /* on doit allouer l'espace mémoire pour la
  chaîne pointée par ch */
   ch = (char*) malloc(sizeof(char) * 100);
   printf("Donner une chaine : ");
   scanf("%s", ch);
```

Chaînes de caractères

- Exemple 3 :

```
    printf("Votre chaine est : %s\n", ch);
    system("pause");
}
```

Chaînes de caractères



Chaînes de caractères

- Exemple 4 :

```
#include <stdio.h>
#include <stdlib.h>
main()
{
   char * ch;
   ch = (char*) malloc(sizeof(char) * 100);
   printf("Donner une chaine : ");
   gets(ch);
   printf("Votre chaine est : %s\n", ch);
   system("pause");
}
```

Chaînes de caractères

Chaînes de caractères

- **Les fonctions de la bibliothèque <string.h> :**
  - Elle offre des **fonctions** pour le **traitement des chaînes** :
    - **strlen** : calcule la longueur d'une chaîne.
    - **strcpy** et **strncpy** : copient des chaînes.
    - **strcat** et **strncat** : concatènent des chaînes.
    - **strcmp** et **strncmp** : comparent des chaînes.
  - Toutes ses **fonctions** sont **définies** dans la bibliothèque <**string.h**>.

Chaînes de caractères

- **La fonction strlen :**
  - Elle **retourne un entier** qui est la **longueur** de la **chaîne en paramètre**.
- **Syntaxe :**
  - Soit `ch` une variable représentant une chaîne.
  - L'appel de **strlen** pour ch est :

```
strlen(ch)
```

  - Attention, **strlen** n'est pas égale à `sizeof(ch)`

Chaînes de caractères

- **Exemple :**

```
#include <stdio.h>
#include <string.h>
main()
{
   char str[100];
   printf("Donner une chaine : ");
   gets(str);
   printf("Sa longueur est : %d\n", strlen(str));
   printf("Sa taille est : %d\n", sizeof(str));
   system("pause");
}
```

Chaînes de caractères

Chaînes de caractères

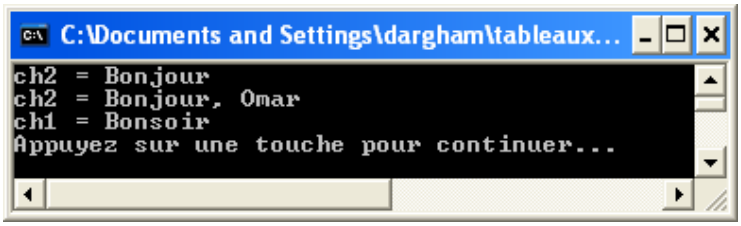
- **Les fonctions strcpy et strncpy :**
  - La fonction **strcpy(ch1, ch2) copie la chaîne ch2 sur la chaîne ch1**.
  - La fonction **strcpy(ch1, ch2, n) copie les n premiers caractères** de la **chaîne ch2 sur la chaîne ch1**.
  - Elles **retournent un pointeur sur la chaîne recopiée**, ou bien le **pointeur NULL en cas d'échec**.

Chaînes de caractères

- **Exemple :**

```
#include <stdio.h>
#include <string.h>
main()
{
   char ch1[25] = "Bonjour, Omar", ch2[25] = "";
   strncpy(ch2, ch1, 7);
   printf("ch2 = %s\n", ch2);
   strcpy(ch2, ch1);
   printf("ch2 = %s\n", ch2);
   strcpy(ch1 + 3, "soir");
   printf("ch1 = %s\n", ch1);
   system("pause");
}
```

Chaînes de caractères



Chaînes de caractères

- **Les fonctions strcat et strncat :**
  - La fonction `strcat(ch1, ch2)` **concatène** (**colle**) **la chaîne** `ch2` **à la fin de la chaîne** `ch1`.
  - La fonction `strncat(ch1, ch2, n)` **concatène les** `n` **premiers caractères** de la **chaîne** `ch2` **à la fin de la chaîne** `ch1`.
  - Elles **retournent un pointeur sur la chaîne concaténée**, ou bien le **pointeur NULL en cas d'échec**.

Chaînes de caractères

- Exemple :

```
#include <stdio.h>
#include <string.h>
main()
{
   char ch1[25] = "Bonjour";
   printf("ch1 = %s\n", ch1);
   strcat(ch1, " Monsieur");
   printf("ch1 = %s\n", ch1);
   strncat(ch1, " et Merci bien", 9);
   printf("ch1 = %s\n", ch1);
   system("pause");
}
```

Chaînes de caractères

Chaînes de caractères

- **Les fonctions strcmp et strncmp :**
  - La fonction `strcmp(ch1, ch2)` **compare les deux chaînes** `ch1` **et** `ch2`.
  - La fonction `strncmp(ch1, ch2, n)` **compare les** `n` **premiers caractères** de la **chaîne** `ch1` **avec ceux de la chaîne** `ch2`.
  - Elles **retournent un entier**.
  - La **comparaison** est effectuée selon **l'ordre lexicographique** (**alphabétique**).

Chaînes de caractères

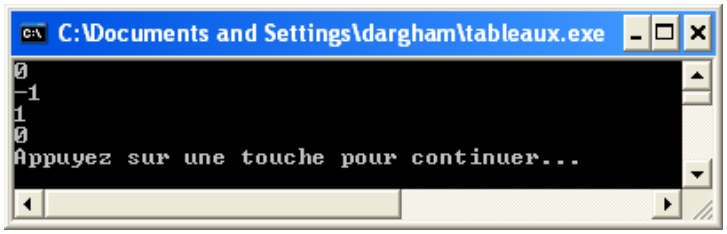
- **Les fonctions strcmp et strncmp :**
  - L'**entier** retourné est :
    - **0**, si **ch1 et ch2** sont **identiques**.
    - **Une valeur positive**, si **ch1 > ch2**.
    - **Une valeur négative**, si **ch1 < ch2**.

Chaînes de caractères

- Exemple :

```
#include <stdio.h>
#include <string.h>
main()
{
   printf("%d\n", strcmp("abc", "abc");
   printf("%d\n", strcmp("abc", "abd");
   printf("%d\n", strcmp("abd", "abc");
   printf("%d\n", strncmp("abcd", "abce", 3);
   system("pause");
}
```

Chaînes de caractères



Chaînes de caractères

- **Les fonctions de manipulation des caractères :**
  - Ces **fonctions sont définies** dans la bibliothèque **<ctype.h>**.
  - Les **fonctions dont les noms commencent** par « **is** » **prennent comme argument** un **caractère** et **effectuent un test particulier** sur ce **caractère**.
  - Elles **retournent** une **valeur non nulle** si **le test est vérifiée** (résultat vrai) et **0 sinon**.

Chaînes de caractères

- **Les fonctions « is… » :**
  - **`islower(car)`** : teste si le **caractère** « **`car`** » est une **lettre minuscule** ('a', 'b', …, 'z').
  - **`isupper(car)`** : teste si le **caractère** « **`car`** » est une **lettre majuscule** ('A', 'B', …, 'Z').
  - **`isdigit(car)`** : teste si le **caractère** « **`car`** » est un **chiffre décimal** ('0', '1', …, '9').
  - **`isxdigit(car)`** : teste si le **caractère** « **`car`** » est un **chiffre hexadécimal** ('0', '1', …, '9', 'a', 'b', 'c', 'd', 'e', 'f', 'A', 'B', 'C', 'D', 'E', 'F').

Chaînes de caractères

- **Les fonctions « is… » :**
  - **`isalpha(car)`** : teste si le **caractère** « **`car`** » est un **caractère alphabétique**, c'est-à-dire une **lettre** ('a', 'b', …, 'z', 'A', 'B', …, 'Z').
  - **`isalnum(car)`** : teste si le **caractère** « **`car`** » est un **caractère alphanumérique** ('a', 'b', …, 'z', 'A', 'B', …, 'Z', '0', '1', …, '9').
  - **`isspace(car)`** : teste si le **caractère** « **`car`** » est un **caractère blanc** (' ', '\t', '\n', '\r', '\f', '\v').

Chaînes de caractères

- **Exemple :**

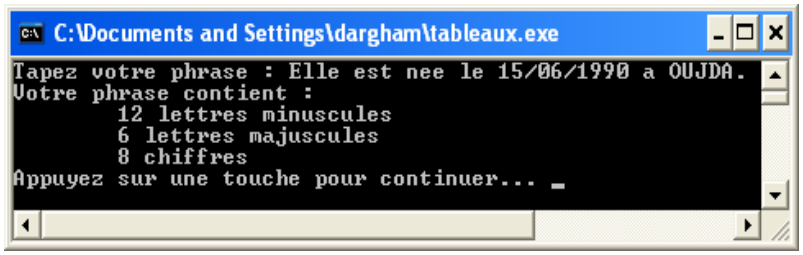
```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
/* ce programme lit une phrase et compte le
   nombre de ses lettres minuscules, le nombre de
   ses lettres majuscules et le nombre de ses
   chiffres */
main()
{
   char phrase[100];
   int l, i, min, maj, chf;

   printf("Tapez votre phrase : ");
   gets(phrase);
```

Chaînes de caractères

```
    min = maj = chf = 0;
    l = strlen(phrase); /* longueur de chaîne */
    for(i = 0; i < l; i++)  {
        if(islower(phrase[i])) min++;
        else if(isupper(phrase[i])) maj++;
        else if(isdigit(phrase[i])) chf++;
    }
    printf("Votre phrase contient :\n");
    printf("\t%d lettres minuscules\n", min);
    printf("\t%d lettres majuscules\n", maj);
    printf("\t%d chiffres\n", chf);
    system("pause");
}
```

Chaînes de caractères



Chaînes de caractères

- **Les fonctions « to… » :**
  - `tolower(car)` : **retourne** la **lettre majuscule correspondante à** « `car` », si ce dernier est une **lettre minuscule**; **autrement**, elle **retourne** `car` lui-même.
  - `toupper(car)` : **retourne** la **lettre minuscule correspondante à** « `car` », si ce dernier est une **lettre majuscule**; **autrement**, elle **retourne** `car` lui-même.

Chaînes de caractères

- **Exemple :**

```
#include <stdio.h>
#include <ctype.h>
main()
{
   printf("%c\n", tolower('X'));
   printf("%c\n", tolower('y'));
   printf("%c\n", toupper('X'));
   printf("%c\n", toupper('y'));
   printf("%c\n", tolower('+'));
   printf("%c\n", toupper('7'));
   system("pause");
}
```

Chaînes de caractères

- **Remarque :**
  - Les fonctions `toupper et tolower` **ne modifient pas leurs arguments**.
  - Pour ce faire, il faut écrire :

```
car = toupper(car);
car = tolower(car);
```

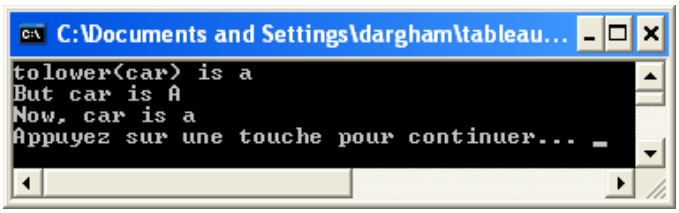
Chaînes de caractères

- **Exemple :**

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

main()
{
   char car = 'A';
   printf("tolower(car) is %c\n", tolower(car));
   printf("But car is %c\n", car);
   car = tolower(car);
   printf("Now, car is %c\n", car);
   system("pause");
}
```

Chaînes de caractères



Chaînes de caractères

- **Les fonctions atoi, atol et atof :**
  - `atoi(str)` : **convertit** une **chaîne** « `str` » en une **valeur entière** (de type `int`).
  - `atol(str)` : **convertit** une **chaîne** « `str` » en une **valeur entière longue** (de type `long int`).
  - `atof(str)` : **convertit** une **chaîne** « `str` » en une **valeur réelle double précision** (de type `double`).
- Elles **sont définies** dans <**stdlib.h**>.

Chaînes de caractères

- **Exemple :**

```
#include <stdio.h>
#include <stdlib.h>
main()
{
   int a;
   double b;
   char str[20];

   printf("Tapez une chaine : ");
   gets(str);
   printf("Vous avez tape : %s\n", str);
```

Chaînes de caractères

```
   a = atoi(str);
   printf("a = %d\n", a);
   b = atof(str);
   printf("b = %g\n", b);
   system("pause");
}
```

Chaînes de caractères