

Algorithmique et Structure de Données 1

Niveau MPI

Année universitaire
2019-2020

Dr. Aymen Sellaouti
Dr. Majdi Jribi

Chapitre 6

Les chaînes de caractères

Plan

3

- Partie 1: Introduction et définition
- Partie 2: Fonctions sur les chaines de caractères
- Partie 3: Les tableaux de chaines de caractères

Plan

4

- Partie 1: Introduction et définition
- Partie 2: Fonctions sur les chaines de caractères
- Partie 3: Les tableaux de chaines de caractères

Introduction et définition

5

Définition :

Une chaîne de caractères C est un tableau unidimensionnel de caractères.
Par convention, la fin de la chaîne de caractères est indiquée par le caractère nul '\0'.
Par conséquent, une chaîne de caractères occupe en mémoire un espace correspondant au nombre de caractères significatifs de la chaîne plus un.

Déclaration

Déclaration en langage algorithmique : **NomVariable: chaîne[longueur]**

Déclaration en C : **char <NomVariable> [Longueur+1];**

Exemples :

char NOM [20];

char PRENOM [20];

char PHRASE [100];

- 'x' est un caractère constant
- "x" est un tableau de caractères qui contient deux caractères

Introduction et définition

6

```
char TXT[5] = "Hello";
```

TXT:

'H'	'e'	'l'	'l'	'o'	☠
-----	-----	-----	-----	-----	---



ERREUR pendant l'exécution

```
char TXT[8] = "Hello";
```

TXT:

'H'	'e'	'l'	'l'	'o'	'\0'	0	0
-----	-----	-----	-----	-----	------	---	---



```
char TXT[6] = "Hello";
```

TXT:

'H'	'e'	'l'	'l'	'o'	'\0'
-----	-----	-----	-----	-----	------



Introduction et définition

7

La précedence des caractères dans l'alphabet 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, . . .

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'

Plan

8

- Partie 1: Introduction et définition
- Partie 2: Fonctions sur les chaines de caractères
- Partie 3: Les tableaux de chaines de caractères

Fonctions sur les chaines de caractères

9

Les fonctions de <stdio.h>

- **scanf** : `scanf("%s" , chaine)`
fonction scanf ne permet pas la saisie d'une chaîne comportant des espaces: les caractères saisis à partir de l'espace ne sont pas pris en compte mais rangés dans le tampon
- **gets** : Lecture de chaînes de caractères **gets(<Chaîne>)**
- **printf** : `printf("%s" , chaine)`
- **puts** : L'écriture de chaînes de caractères **puts(<Chaîne>)**

Fonctions sur les chaines de caractères

10

Les fonctions de <stdio.h>

- **sscanf** : C'est la version de scanf dédiée aux chaines de caractères, càd que la source de lecture n'est plus l'entrée standard **stdin** mais une chaine de caractère et sa syntaxe est : `sscanf(chaine_source, "format",&var_dest);`
- **sprintf**(chaine_source,"données à introduire", variables);

Fonctions sur les chaînes de caractères

11

Les fonctions de <string.h>

strlen(s) : fournit la longueur de la chaîne sans compter le '\0'

strcpy(s, t) : copie t vers s, Remarque : le '\0' est aussi copié

strncpy(s,t,n) : copie les n premiers caractères de t vers s

strcat(s, t) : ajoute t à la fin de s (à partir du '\0' de s qui sera décalé)

strncat(s,t,n) : concatène les n premiers caractères de t à la fin de s

strcmp(s, t) : compare s et t lexicographiquement :

 négatif si s précède t

 zéro si s est égal à t

 positif si s suit t

strncmp(s, t, n) compare les n premiers caractères de s et t

Fonctions sur les chaines de caractères

12

RAPPEL PRINTF ET SCANF (DE MÊME POUR SPRINTF ET SSCANF)

printf

- Concernant printf, un certain nombre de caractères optionnels peuvent être insérés entre le symbole % et le caractère spécifiant la conversion:
 - Le signe – pour demander un cadrage à gauche, au lieu du cadrage à droite par défaut.
 - un nombre indiquant la taille minimale en caractères du champs à imprimer. Des espaces jouant le rôle de caractère de remplissage.
 - un point décimal, suivi d'un nombre donnant la précision de la partie fractionnaire, c'est à dire le nombre de chiffre significatifs après le point. Si la donnée n'est pas du type flottant, ce nombre représente la taille maximale du champs à imprimer.

Fonctions sur les chaînes de caractères

13

RAPPEL PRINTF ET SCANF (DE MÊME POUR SPRINTF ET SSCANF)

printf

➤ Exemple :

- `%8d` : Imprime un nombre en décimal cadré à droite dont la longueur du champ imprimable est de huit caractères. Des espaces de remplissage précèdent le nombre.

`X=12345`

`Printf (« %8d »,x);`// affichera 3 espaces suivies de 12345

`Printf (« %-8d »,x);`// affichera 12345 suivi de 3 espaces

- `%-25s` : Imprime une chaîne de caractères cadrée à gauche assurant une longueur minimum de 25 caractères.
- `%.6f` : Imprime un nombre flottant avec un maximum de six chiffres significatifs.

Fonctions sur les chaines de caractères

14

RAPPEL PRINTF ET SCANF (DE MÊME POUR SPRINTF ET SSCANF)

scanf

- Pour limiter le nombre de caractère lu par scanf, on mentionne le nombre de caractères à lire entre le et le caractère spécifiant la conversion
 - `%3s` : lira seulement les 3 premiers caractères.
- Le caractère `*` précédé de `%` spécifie que la valeur lue sera ignoré, donc non affecté à la variable suivante.
 - **sscanf**(T,"%d%*s%d%*s%d%*s", &heure, &minutes, &secondes) ; permettra de correctement extraire les données pour l'entrée suivante :
T:17 H 35 min 30 secondes puisque les chaines "h", "min" et "secondes" seront ignorées.

Plan

15

- Partie 1: Introduction et définition
- Partie 2: Fonctions sur les chaines de caractères
- Partie 3: Les tableaux de chaines de caractères

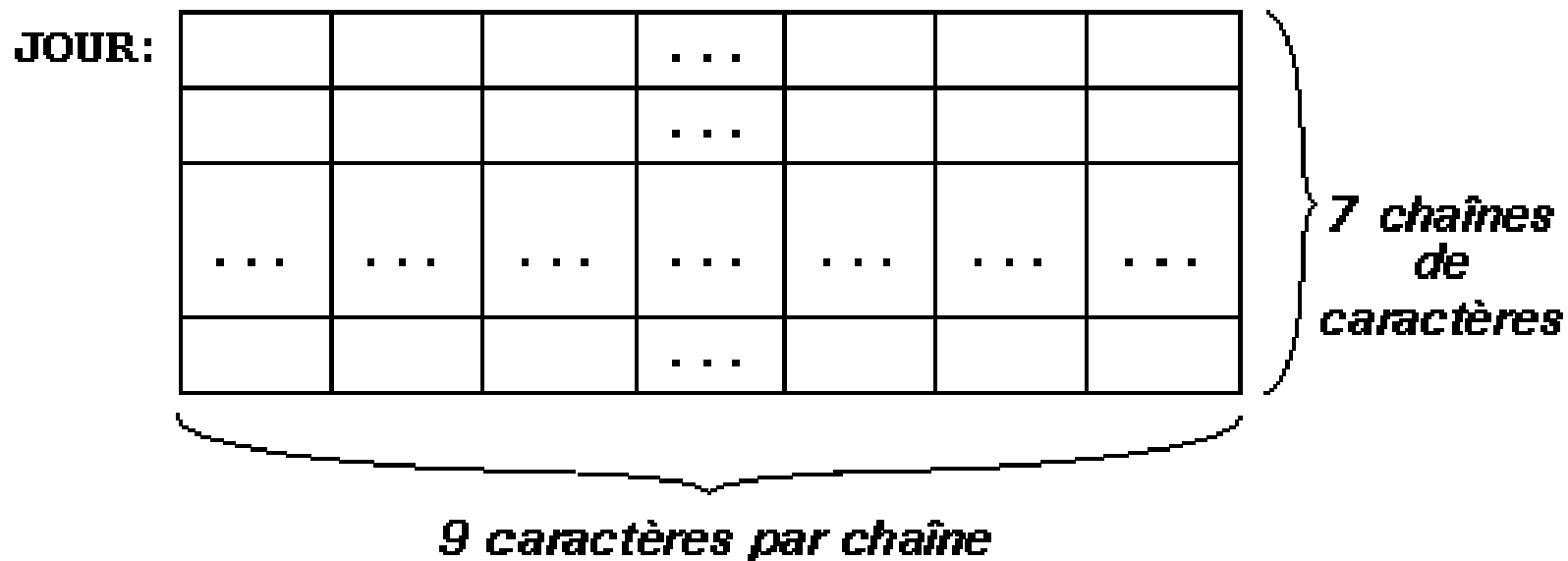
Les tableaux de chaînes de caractères

16

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

Déclaration :

La déclaration `char JOUR[7][9]`; réserve l'espace en mémoire pour 7 mots contenant 9 caractères (dont 8 caractères significatifs).



Les tableaux de chaines de caractères

17

JOUR:

'l'	'u'	'n'	'd'	'i'	'\0'			
'm'	'a'	'r'	'd'	'i'	'\0'			
'm'	'e'	'r'	'c'	'r'	'e'	'd'	'i'	'\0'
...
'd'	'i'	'm'	'a'	'n'	'c'	'h'	'e'	'\0'

Char JOUR[7][9]={"lundi","mardi","mercredi","jeudi","vendredi","samedi","dimanche"};

Les tableaux de chaînes sont mémorisés ligne par ligne.

La variable JOUR aura donc besoin de $7*9*1 = 63$ octets en mémoire.

Exemple : L'exécution des trois instructions suivantes:

```
char JOUR[7][9]= {"lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi", "dimanche"};
int I = 2;
printf("Aujourd'hui, c'est %s !\n", JOUR[I]);
affichera la phrase: Aujourd'hui, c'est mercredi !
```

Les tableaux de chaînes de caractères

18

Affectation

L'attribution d'une chaîne de caractères à une composante d'un tableau de chaînes se fait 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".

Accès aux caractères

Evidemment, il existe toujours la possibilité d'accéder directement aux différents caractères qui composent les mots du tableau.

Exemple : L'instruction

```
for(i=0; i<7; i++)
```

```
    printf("%c ", JOUR[i][0]);
```

va afficher les premières lettres des jours de la semaine: l m m j v s d