

Chapitre V

Les chaînes de caractères

Les chaînes de caractères

Objectifs :

L'objectif de ce chapitre est de :

- étudier les chaînes de caractères en algorithmique
- écrire des algorithmes qui traitent des caractères et des chaînes de caractères.

Définition :

Une chaîne de caractères est une séquence ordonnée de caractères :

Rang	1	2	3	4	5
Ch	S	A	L	U	T

- La chaîne **Ch** est constituée des *caractères* : "S", "A", "L", "U", "T".
- **Chaque caractère** se trouve à une **position donnée** dans la chaîne (on parle alors de **rang**). **Le rang d'un caractère est à la chaîne ce que l'indice d'un élément est au tableau.**
- En algorithmique, le **premier caractère** se trouve au **rang 1**, le **deuxième caractère** se trouve au **rang 2** et ainsi de suite.

Les chaînes de caractères

Une chaîne de caractère est une **suite finie de caractères**.

Exemples :

- La chaîne de caractères **"SALUT"** est constituée des *caractères* : **"S", "A", "L", "U", "T",**.
- La chaîne de caractères **"Bonjour"** est constituée des *caractères* : **"B", "o", "n", "j", "o", "u", "r",**.
- La chaîne de caractères **"567"** est constituée des *chiffres 5, 6 et 7*. **Mais en aucun cas il ne faut la** confondre avec la grandeur numérique **567**.
- La chaîne **""** représente une *chaîne de caractères vide*
- La chaîne de caractères **" "** est une chaîne de caractères ne contenant qu'un seul caractère qui est ici le caractère *espace* (**à ne pas confondre avec la chaîne de caractères vide**).

Les chaînes de caractères sont constituées de caractères où chaque caractère *n'est autre qu'un symbole* du **code ASCII**

Les chaînes de caractères

1. Le type caractère

Le **type caractère** s'applique à tous **les caractères du code ASCII** (*American standard Code For Information Interchange*), la liste des caractères comprend :

- Les **lettres** : "A" .. "Z", "a" .. "z"
- Les **chiffres** : "0" .. "9"
- Les **caractères spéciaux** : "/" ; "*" ; "?" ; "&" ; etc.
- Les **caractères de contrôle** : <Retour Chariot>; <Echap>; etc.

Dans un **système informatique**:

- **à chaque caractère** est associé une valeur numérique : **son code ASCII**.
- Quand on **stocke un caractère en mémoire** (dans une **variable**), on mémorise en réalité son **code ASCII**.
- un **code ASCII** est codé sur un **octet (huit bits)**.

L'ensemble des codes est recensé dans une table nommée **Table des codes ASCII** (diapo suivant).

Table des Codes ASCII

DECIMAL VALUE	HEXA DECIMAL VALUE	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	BLANK (NULL)	▶	SP	0	@	P	'	p	Ç	É	á	⋮	⌋	⌌	∞	≡
1	1	☺	◀	!	1	A	Q	a	q	ü	æ	í	⋮	⌋	⌌	β	±
2	2	☹	↓	"	2	B	R	b	r	é	Æ	ó	⋮	⌋	⌌	Γ	∞
3	3	♥	!!	#	3	C	S	c	s	â	ô	ú	⋮	⌋	⌌	π	∞
4	4	♦	¶	\$	4	D	T	d	t	ä	ö	ñ	⋮	⌋	⌌	Σ	∫
5	5	♣	§	%	5	E	U	e	u	à	ò	Ñ	⋮	⌋	⌌	σ	∫
6	6	♠	¶	&	6	F	V	f	v	å	û	ä	⋮	⌋	⌌	μ	÷
7	7	BEL	↓	'	7	G	W	g	w	ç	ù	ó	⋮	⌋	⌌	τ	≈
8	8	BS	↑	(8	H	X	h	x	ê	ÿ	ï	⋮	⌋	⌌	ø	°
9	9	HT	↓)	9	I	Y	i	y	ë	Ö	⌋	⋮	⌋	⌌	θ	•
10	A	LF	→	*	:	J	Z	j	z	è	Ü	⌋	⋮	⌋	⌌	Ω	•
11	B	VT	←	+	;	K	I	k	{	ï	ç	½	⋮	⌋	⌌	δ	√
12	C	FF	FS	,	<	L	\	l	;	î	£	¼	⋮	⌋	⌌	∞	π
13	D	CR	GS	-	=	M	I	m	}	ì	¥	ì	⋮	⌋	⌌	φ	²
14	E	♪	RS	.	>	N	^	n	~	Ä	Ŕ	«	⋮	⌋	⌌	∞	■
15	F	☼	US	/	?	O	_	o	Δ	Å	ƒ	»	⋮	⌋	⌌	∞	∞

Exemples :

- Code ASCII(**S**) = **83** = (80)₁₀ + (3)₁₀ = (5)₁₆ (3)₁₆ = (0101 0011)₂
- Code ASCII(**M**) = **77** = (64)₁₀ + (13)₁₀ = (4)₁₆ (D)₁₆ = (0100 1101)₂
- Code ASCII(**I**) = **73** = (64)₁₀ + (9)₁₀ = (4)₁₆ (9)₁₆ = (0100 1001)₂
- Code ASCII(**A**) = **65** = (64)₁₀ + (1)₁₀ = (4)₁₆ (1)₁₆ = (0100 0001)₂

Les chaînes de caractères

Les fonctions standard sur les caractères

Fonction	Rôle	Exemple
Asc(c)	retourne le code ASCII du caractère c	$i \leftarrow \text{Asc}(\text{"A"})$ $\Rightarrow i$ contiendra 65
Car(i)	retourne le caractère dont le code ASCII est égal à i.	$c \leftarrow \text{Car}(65)$ $\Rightarrow c$ contiendra "A"
Succ(c)	retourne le successeur du caractère c	$c \leftarrow \text{Succ}(\text{"a"})$ $\Rightarrow c$ contiendra "b"
Pred(c)	retourne le prédécesseur du caractère c	$c \leftarrow \text{Pred}(\text{"b"})$ $\Rightarrow c$ contiendra "a"
Majus(c)	retourne le majuscule du caractère c	$c \leftarrow \text{Majus}(\text{"a"})$ $\Rightarrow c$ contiendra "A"

Les chaînes de caractères

Les fonctions standard sur les caractères

Exemple : Ecrire un algorithme qui affiche une table des codes ASCII contenant *des lettres minuscules sous la forme suivante* :

```
Le code ASCII de (a) = 97
Le code ASCII de (b) = 98
....
Le code ASCII de (z) = 122
```

```
Algorithme table_minuscule;
Var
    c : Caractère;
Début
    Pour c ← "a" à "z" Faire
        Ecrire("Le code ASCII de (" , c, ") =", asc(c));
    FinPour
Fin
```

En traduisant l'algorithme ci-après en **Langage C**,
l'exécution du programme donnera :

```
le code ASCII des caractères en Miniscules:
Code ASCII(a) = 97
Code ASCII(b) = 98
Code ASCII(c) = 99
Code ASCII(d) = 100
Code ASCII(e) = 101
Code ASCII(f) = 102
Code ASCII(g) = 103
Code ASCII(h) = 104
Code ASCII(i) = 105
Code ASCII(j) = 106
Code ASCII(k) = 107
Code ASCII(l) = 108
Code ASCII(m) = 109
Code ASCII(n) = 110
Code ASCII(o) = 111
Code ASCII(p) = 112
Code ASCII(q) = 113
Code ASCII(r) = 114
Code ASCII(s) = 115
Code ASCII(t) = 116
Code ASCII(u) = 117
Code ASCII(v) = 118
Code ASCII(w) = 119
Code ASCII(x) = 120
Code ASCII(y) = 121
Code ASCII(z) = 122
```

Les chaînes de caractères

Les fonctions standard sur les caractères

Exemple : Ecrire un algorithme qui affiche une table des codes ASCII contenant des lettres minuscules sous la forme :

```
Le code ASCII de A = 65  
Le code ASCII de B = 66  
....  
Le code ASCII de Z = 90
```

```
Algorithme table_majus;  
  Var  
    c : Caractère;  
  Début  
    Pour c ← "A" à "Z" Faire  
      Ecrire("Le code ASCII de ",c," est ",asc(c));  
    FinPour  
  Fin
```

En traduisant l'algorithme ci-dessus en **Langage C**, l'exécution du programme donnera :

```
le code ASCII des caractères en Majuscules:  
Code ASCII de A est 65  
Code ASCII de B est 66  
Code ASCII de C est 67  
Code ASCII de D est 68  
Code ASCII de E est 69  
Code ASCII de F est 70  
Code ASCII de G est 71  
Code ASCII de H est 72  
Code ASCII de I est 73  
Code ASCII de J est 74  
Code ASCII de K est 75  
Code ASCII de L est 76  
Code ASCII de M est 77  
Code ASCII de N est 78  
Code ASCII de O est 79  
Code ASCII de P est 80  
Code ASCII de Q est 81  
Code ASCII de R est 82  
Code ASCII de S est 83  
Code ASCII de T est 84  
Code ASCII de U est 85  
Code ASCII de U est 86  
Code ASCII de W est 87  
Code ASCII de X est 88  
Code ASCII de Y est 89  
Code ASCII de Z est 90
```


Les chaînes de caractères

Les fonctions standard sur les caractères

Exemple : Ecrire un algorithme qui affiche une table des codes ASCII contenant des chiffres (0..9) sous la forme :

```
Le code ASCII de 0 est 48  
Le code ASCII de 1 est 49  
....  
Le code ASCII de 9 est 57
```

En traduisant l'algorithme ci-après en **Langage C**, l'exécution du programme donnera :

```
le code ASCII des chiffres ( de 0 a 9):  
Code ASCII de 0 est 48  
Code ASCII de 1 est 49  
Code ASCII de 2 est 50  
Code ASCII de 3 est 51  
Code ASCII de 4 est 52  
Code ASCII de 5 est 53  
Code ASCII de 6 est 54  
Code ASCII de 7 est 55  
Code ASCII de 8 est 56  
Code ASCII de 9 est 57
```

```
Algorithme table_chiffre;  
Var  
    c : Caractère;  
Début  
    Pour c ← "0" à "9" Faire  
        Ecrire("Le code ASCII de ",c," est ",asc(c));  
    FinPour  
Fin
```

Exemples d'application sur les chaînes de caractères

Exemple 1: Ecrire un algorithme qui lit un caractère au clavier puis affiche son prédécesseur, son successeur et le code ASCII de son équivalent en majuscule.

Algorithme Caractère;

Var

c : Caractère;

Début

Ecrire("Entrer un caractère: ");

Lire(c);

Ecrire("son prédécesseur est :", pred(c));

Ecrire("son successeur est :", succ(c));

Ecrire("En majuscule est:", asc(majus(c)));

Fin

Exemples d'application sur les chaînes de caractères

Exemple 2: Ecrire un algorithme qui permet lire un caractère puis affiche son équivalent en majuscule sans utiliser la **fonction Majus(c)**;

Solution 1

AlgorithmecaractereMaj;

Var

c1, c2 : Caractère;

Début

Répéter

Ecrire("Entrer un caractère: ");

Lire(c1);

Jusqu'à((asc(c1) >= 97) ET (asc(c1) <= 122) OU ((asc(c1) >= 65)
ET (asc(c1) <= 90))

Si (asc(c1) >= 97) ET (asc(c1) <= 122) **alors**

c2 ← car(asc(c1) - 32);

Sinon

c2 ← c1;

FinSi

Ecrire(c2);

Fin

Remarque: Le but de la boucle répéter est d'obliger l'utilisateur à saisir une lettre.

Exemples d'application sur les chaînes de caractères

Algorithme convertEnMaj;

Var

C1,C2 : caractere;

I : entier;

Debut

Ecrire("Entrez le caractere à convertir : ");

Lire(C1);

Si (Asc(C1)>=65) ET (Asc(C1)<=90) alors

 Ecrire("Le caractère EST EN MAJUSCULE : ", C1);

Sinon

 Si(Asc(C1)>=97) ET (Asc(C1)<=122) alors

 C2 ← car(Asc(C1)-32);

 Ecrire("Le caractère convertie est : ", C2);

 Sinon

 Ecrire("Ce n'est pas une lettre pour le convertir!");

 Finsi

Finsi

Fin

Solution 2

Les chaînes de caractères

Le type chaîne de caractère

Une chaîne est une suite de caractères. La chaîne ne contenant aucun caractère est appelée *chaîne vide*.

1. Déclaration d'une chaîne

Var

c : Caractère;

ch : Chaîne;

chn : Chaîne[20];

La variable **ch** peut contenir jusqu'à **255 caractères** alors que **chn** peut contenir au **maximum 20**.

Opérations sur les chaînes de caractères

La concaténation

C'est l'assemblage de deux chaînes de caractères en utilisant l'opérateur « + ».

Exemples :

```
Chn1 ← "Turbo";
```

```
Chn2 ← "Pascal";
```

```
Chn3 ← chn1+" "+chn2;
```

→ La variable **chn3** contiendra **"Turbo Pascal"**

Opérations sur les chaînes de caractères

Les opérateurs relationnels (>, >=, <, <=, =, #)

Il est possible d'effectuer une **comparaison entre deux chaînes de caractères**, le résultat est de **type booléen**. La comparaison se fait **caractère par caractère** de la gauche vers la droite selon le **code ASCII**.

Exemples:

- L'expression ("**a**" > "**A**") est **vraie** puisque le code ASCII de "**a**" (97) est supérieur à celui de "**A**" (65)
- L'expression ("**programme**" < "**programmation**") est **fausse** puisque "**e**" > "**a**"
- L'expression ("" = " ") est **fausse** (le caractère vide est **différent** du caractère espace).

Opérations sur les chaînes de caractères

Accès à un caractère dans une chaîne

Pour accéder à un caractère de la chaîne, il suffit d'indiquer le nom de la chaîne suivi d'un entier entre crochets qui indique la position du caractère dans la chaîne.

Exemples:

```
Chn ← "Turbo Pascal";
```

```
c ← chn[7];
```

La variable **c** contiendra le caractère **"P"**

En général, **ch[i]** désigne le **i^{ème}** caractère de la chaîne **ch**

Procédures et fonctions standards sur les chaînes

1. Procédures standards

Procédure	Rôle	Exemple
Efface(Chaîne, P, N)	Enlève N caractères de Chaîne à partir de la position P donnée.	chn ← "Turbo Pascal" efface(chn,6,7) ⇒ chn contiendra "Turbo"
Insert(Ch1, Ch2, P)	Insère la chaîne Ch1 dans la chaîne Ch2 à partir de la position P.	ch1 ← "D" ch2 ← "AA" insert(ch1,ch2,2) ⇒ ch2 contiendra "ADA"
Convch(Nbr, Ch)	Converti le nombre Nbr en une chaîne de caractères Ch.	n = 1665 convch(n,chn) ⇒ chn contiendra la chaîne "1665"

Procédures et fonctions standards sur les chaînes

2. Fonctions standards

Fonction	Rôle	Exemple
Long(Chaîne)	Retourne la longueur de la chaîne.	chn ← "Turbo Pascal" n ← Long(chn) ⇒ n contiendra 12
Copie(Chaîne, P, N)	Copie N caractères de Chaîne à partir de la position P donnée.	ch1 ← "Turbo Pascal" ch2 ← Copy(ch1,7,6) ⇒ ch2 contiendra "Pascal"
Position(Ch1, Ch2)	Retourne la position de la première occurrence de la chaîne Ch1 dans la chaîne Ch2.	ch1 ← "as" ch2 ← "Turbo Pascal" n ← Position(ch1,ch2) ⇒ n contiendra 8

Procédures et fonctions standards sur les chaînes

3. Exemples d'application

1. Ecrire un algorithme «**Palindrome**» qui lit une chaîne de caractères et vérifie si cette chaîne est un palindrome ou non.

Un palindrome est un mot (phrase) qui peut être lu indifféremment de droite à gauche ou de gauche à droite

Exemples :

"AZIZA", "LAVAL", "RADAR", "2002", etc.

"Engage le jeu que je le gagne" (Alain Damasio)

2. Ecrire un algorithme qui lit un entier n positif puis affiche son équivalent en binaire (base 2). [Voir corrigé TD N°4](#)

Exemples :

$$(23)_{10} = (10111)_2$$

Solution

Algorithme Mot_Palindrome;

Var

ch : Chaîne;
i, L : Entier;
Pal : Booléen;

Début

Ecrire("Entrer une chaîne non vide : ");
Lire(ch);
L ← long(ch);
Pal ← Vrai;
i ← 1;

TantQue ((i <= L/2) ET (Pal)) Faire

Si (ch[i] = ch[L-i+1]) alors

i ← i + 1;

Sinon

Pal ← Faux;

FinSi

FinTQ

Si (Pal) Alors

Ecrire(ch, " est un palindrome")

Sinon

Ecrire(ch, " n'est pas un palindrome")

FinSi

Fin

Algorithme Decimal_Binaire;

Var

n, reste : Entier;

c : Caractère;

equiv : Chaîne;

Début

Ecrire("Saisir un entier :");

Lire(n);

equiv ← "";

TantQue (n <> 0) Faire

 reste ← n Mod 2;

 convch(reste, c);

 equiv ← c + equiv;

 n ← n Div 2;

FinTantQue

Ecrire("Nombre binaire équivalent =", equiv);

Fin

Solution

Fin.