

# INFO-H-100

## Séances d'exercices 9

Les chaînes de caractères

# Définitions de types

- Le mot clé **typedef** permet au programmeur de définir un synonyme pour un type de données.
- Par exemple, **simplifie l'écriture** des fonctions qui prennent une matrice en paramètre.

- Exemple :

```
#include <iostream>
using namespace std;

const int TAILLE = 128;
typedef int Vecteur[TAILLE];

void affiche(Vecteur v, int n);
```

- Convention : le nom d'un type défini **commence par une majuscule**

# Exemple avec une matrice

```
#include <iostream>
using namespace std;
```

```
const int L = 5;
const int C = 5;
typedef int Mat[L][C];
```

```
void print(Mat M, int lines, int columns);
```

```
int main()
{
    Mat board;
    // (...)
    print(board, L, C);
    return 0;
}
```

```
void print(Mat M, int lines, int columns)
{
    for(int i=0; i<lines; i++)
    {
        for(int j=0; j<columns; j++)
        {
            cout << M[i][j] << "\t";
        }
        cout << endl;
    }
}
```

# Caractères

- En C++, le type `char` est le type caractère.
- Exemple :
  - `char c = 'a';`
  - Remarquez l'utilisation des `guillemets simples`
- Caractères spéciaux :
  - `'\n'` : retour à la ligne
  - `'\t'` : tabulation
  - `'\0'` : fin d'une chaîne (non imprimable)
  - ...

# ASCII

- Le type char permet de stocker la **valeur ASCII** d'un caractère, c'est-à-dire **un nombre entier** sur un octet (8 bits).
- Exemples :
  - 'A' vaut 65, 'B' vaut 66, ...
  - 'a' vaut 97, 'b' vaut 98, ...
- Question :
  - comment mettre en majuscule un caractère ?

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Dec	Hex	Char									
128	80	Ç	160	A0	á	192	C0	Ł	224	E0	α
129	81	ù	161	A1	í	193	C1	ł	225	E1	β
130	82	é	162	A2	ó	194	C2	Ť	226	E2	Γ
131	83	â	163	A3	ú	195	C3	ţ	227	E3	π
132	84	ä	164	A4	ñ	196	C4	—	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	†	229	E5	σ
134	86	ã	166	A6	ª	198	C6	‡	230	E6	μ
135	87	ç	167	A7	º	199	C7	‡	231	E7	τ
136	88	ê	168	A8	¿	200	C8	Ł	232	E8	ϕ
137	89	ë	169	A9	ƒ	201	C9	Ŧ	233	E9	⊙
138	8A	è	170	AA	¬	202	CA	⊥	234	EA	Ω
139	8B	ì	171	AB	½	203	CB	Ŧ	235	EB	δ
140	8C	î	172	AC	¼	204	CC	‡	236	EC	∞
141	8D	ï	173	AD	ı	205	CD	=	237	ED	∞
142	8E	Ä	174	AE	«	206	CE	‡	238	EE	ε
143	8F	Å	175	AF	»	207	CF	⊥	239	EF	∏
144	90	É	176	B0	⋯	208	DO	⊥	240	FO	≡
145	91	æ	177	B1	⋮	209	D1	Ŧ	241	F1	±
146	92	Æ	178	B2	■	210	D2	Ŧ	242	F2	≥
147	93	ó	179	B3		211	D3	Ł	243	F3	≤
148	94	ö	180	B4	†	212	D4	Ł	244	F4	[
149	95	ò	181	B5	‡	213	D5	Ŧ	245	F5	]
150	96	û	182	B6	‡	214	D6	Ŧ	246	F6	÷
151	97	ù	183	B7	Ŧ	215	D7	‡	247	F7	≈
152	98	ÿ	184	B8	Ŧ	216	D8	‡	248	F8	°
153	99	Ö	185	B9	‡	217	D9	Ŧ	249	F9	•
154	9A	Û	186	BA		218	DA	Ŧ	250	FA	·
155	9B	◊	187	BB	Ŧ	219	DB	■	251	FB	√
156	9C	£	188	BC	Ŧ	220	DC	■	252	FC	²
157	9D	¥	189	BD	Ŧ	221	DD	■	253	FD	ˣ
158	9E	ℳ	190	BE	Ŧ	222	DE	■	254	FE	■
159	9F	f	191	BF	Ŧ	223	DF	■	255	FF	□

Attention, cette deuxième table dépend de l'encodage

# Manipulation de caractères

```
char a = 32; //le caractère espace
```

```
char chiffre = '0' + 7; //le caractère '7'
```

```
char lettre = 'a' + 4; //la lettre 'e'
```

```
if (c >= 'a' && c <= 'z')
```

```
    cout << c << " est minuscule";
```

```
int decalage = ('A' - 'a');
```

```
char c = 'c' + decalage; //la lettre 'C'
```

# Chaînes de caractères

- En C++, les chaînes de caractères sont simplement des **vecteurs de caractères terminés par le caractère `'\0'`**.

- Exemple :

```
char chaine[] = "mot";  
chaine[1] = 'e';  
for(int i=0; chaine[i]!='\0'; i++)  
{  
    cout << chaine[i];  
}
```

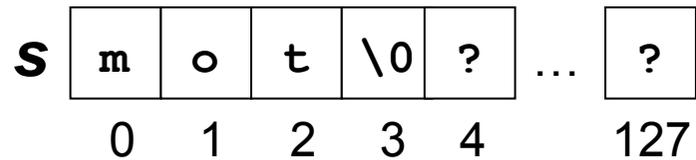
m	o	t	\0
0	1	2	3
m	e	t	\0
0	1	2	3

# Lecture et écriture

```
#include <iostream>
using namespace std;

const int TAILLE= 128;
typedef char Chaine[TAILLE];

int main()
{
    Chaine s;
    cin >> s;
    cout << s << endl;
}
```



# Exercices

- 85, 86, 87 + variante 1, 88, 89
- (90, 92, 93)