

Séance N°3

Tableaux statiques et chaînes de caractères

Les **tableaux** sont des **structures** permettant de stocker des **données** de même type (même taille mémoire en octets) pour faciliter les traitements itératifs sur un groupe de valeurs, comme par exemple sur les pixels d'une image. Pour ranger un ensemble de valeurs en mémoire, on pourra utiliser un **tableau statique** de **dimension fixe**. Les traitements **itératifs** sont réalisés à l'aide de boucles *for*.

Tableau statique : *type variable_tableau[nbre maxi d'éléments] ;*

La **taille** ou dimension du tableau est fixée lors de la **déclaration**. L'espace mémoire nécessaire est réservée à la compilation dans d'une zone de mémoire contigüe (**allocation mémoire**).

Ex 1 – Testez le programme suivant :

```
// Programme qui saisie la valeur de 6 entiers, puis affiche les 6 valeurs, la somme et la
moyenne
#include <iostream>
using namespace std;
void main()
{
int t[6]; // Tableau automatique de 6 entiers de nom t
int somme=0;
for(int i=0; i<6; i++)
    {
    cout<<endl<<" saisir une valeur : ";
    cin>>t[i] ;
    somme += t[i] ;
    }
for(int i=0; i<6; i++)      cout<<t[i]<<endl;
cout<< endl<<" somme = "<< somme;
cout<< endl<<" moyenne = "<< somme/6;
}
```

La variable *t* est le **nom** du tableau. La variable *i* est l'**indice** du tableau (toujours un **entier**) et permet d'accéder à la valeur *t[i]*.

Chaîne de caractères : *char nom[nbre maxi de caractères] ;*

Une chaîne de caractères est un tableau de type *char*.

Ex 2 – Testez le programme suivant :

```
// Programme qui initialise une chaîne de caractères
#include <iostream>
using namespace std;
void main()
{
char tt[20]; // Tableau statique de 20 caractères
tt[0] = 'B' ;
tt[1] = 'O' ;
tt[2] = 'N' ;
tt[3] = 'J' ;
tt[4] = 'O' ;
tt[5] = 'U' ;
tt[6] = 'R' ;
tt[7] = '\0' ;
cout << endl<< tt << endl;
}
```

A quoi correspond le caractère '\0' ?

Ex 3 – Testez le programme suivant :

```
// Programme qui initialise une chaine de caractères
#include <iostream>
using namespace std;

void main(){
    char str[] = "coucou";
    cout << str << endl;
    for (int i = 0; i < 6; i++)
    {
        cout << str[i] << endl;
    }
}
```

Un caractère se représente entre deux **apostrophes** 'B', alors qu'une suite de caractères se représente entre deux **guillemets** "coucou".

Chaque caractère est représenté par un code numérique (appelé **code ASCII American Standard Code for Information Interchange**) qui permet l'échange des données entre les éléments de l'ordinateur (clavier, unité centrale, écran ...).

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	.	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[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	[NEGATIVE 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	[ENG OF 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	[SUBSTITUTE]	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	[DEL]

Ex 4 – Testez le programme suivant :

```
// Code ASCII
#include <iostream>
using namespace std;

void main(){
    char lettrel = 'A';
    char lettre2 = lettrel + 1;
    int lettrel_int = 'A';
    int lettre2_int = lettrel_int + 1;
    cout << lettrel << endl;
    cout << lettre2 << endl;
    cout << lettrel_int << endl;
    cout << lettre2_int << endl;
}
```

Chaque caractère est ainsi associé à un entier (code ASCII). Selon le type, *cout* affiche soit le caractère, soit son code ASCII.

TRAVAIL PERSONNEL

Pour mettre en application ce que j'ai appris, je fais les exercices suivants :

Exercice 1

Compléter le programme suivant pour que la phrase saisie soit affichée en MAJUSCULES.

```
// Exercice 1
#include <iostream>
using namespace std;

void main(){
    char str[256];
    cout << endl << "Entre un mot en minuscules : " << endl;
    cin >> str ;

    // à compléter

    cout << endl << "Voici le mot en MAJUSCULES : ";
    cout << str << endl;
}
```

Exercice 2

Écrire le programme qui saisit 10 valeurs réelles dans un tableau, puis détermine et affiche le **minimum** et **maximum**.

Exercice 3

Modifier le programme précédent pour initialiser le tableau avec 10 valeurs **aléatoires**. On pourra utiliser la fonction *rand()* de la bibliothèque *stdlib.h*. Modifier le programme pour afficher les valeurs du tableau de la plus grande à la plus petite.

BILAN PERSONNEL

Ce que j'ai appris aujourd'hui : (à compléter)

Vocabulaire informatique : (à compléter)