



**CLUBSDAD**

**Faculté des Sciences et Techniques Guéliz  
Université Cadi Ayad,  
Marrakech**

**Tutoriel de -C-  
Semaine 03**

**Les Tableaux en langage C**

*Réalisé par : Z. ELMOURABIT ; K. NAIM ; I. ASAKOUR ; S. BASKAR*

**L'objectif :**

Comprendre la notion des **tableaux** en C, avec des manipulations en utilisant les **fonctions**, au but de comment déclarer un tableaux, comment faire le remplissage et aussi affichage .....

**Prêts ?**

**On Commence !...**

C'est quoi un tableau en C ?

Les tableaux sont des regroupements de plusieurs objets. Cependant, à l'inverse de celles-ci, les tableaux regroupent des données de même type et de manière contiguë

### Tableau de 1 dimension :

La définition d'un tableau nécessite trois informations :

- Le type des éléments du tableau (rappelez-vous : un tableau est une suite de données de même type) ;
- Le nom du tableau (en d'autres mots, son identificateur) ;
- La longueur du tableau (autrement dit, le nombre d'éléments qui le composent). Cette dernière doit être une expression entière.

A screenshot of a code editor window with a dark background and a purple border. The title bar reads "Declaration d'un tableau". The code is as follows:

```
type identificateur[longueur];  
//exemple  
int tab[20];
```

Comme pour les variables, il est possible d'initialiser un tableau ou, plus précisément, tout ou une partie de ses éléments. L'initialisation se réalise de la même manière que pour les structures, c'est-à-dire à l'aide d'une liste d'initialisation, séquentielle ou sélective.

Initialisation avec une longueur explicite

```
int tab[3] = { 1, 2, 3 };
```

Initialisation avec une longueur implicite

```
int tab[] = { 1, 2, 3 };
```

Comment recuperer les valeurs

```
int tab[3] = { 1, 2, 3 };  
//tab[0] = 1  
//tab[1] = 2  
//tab[2] = 3
```

## Tableau de 2 dimensions :

Jusqu'à présent, nous avons travaillé avec des tableaux linéaires, c'est-à-dire dont les éléments se suivent les uns à la suite des autres. Il s'agit de tableaux dit à une dimension ou unidimensionnels.

Cependant, certaines données peuvent être représentées plus simplement sous la forme de tableaux à deux dimensions

(autrement dit, organisées en lignes et en colonnes). C'est par exemple le cas des images (non vectorielles) qui sont des matrices de pixels ou, plus simplement, d'une grille de Sudoku qui est organisée en neuf lignes et en neuf colonnes.

Le langage C vous permet de créer et de gérer ce type de tableaux dit multidimensionnels (en fait, des tableaux de tableaux) et ce, bien au-delà de deux dimensions.

```
● ● ● declaration d'un tableau 2D

Type tab_name[NombreDeLigne][NombreDeColonnes];
//Exemple
int tab[20][35];
```

```
● ● ● Comment recuperer les valeurs Tableau 2D

int tab[2][2] = { { 1, 2 }, { 3, 4 } };
//tab[0][0] = 1
//tab[0][1] = 2
//tab[1][0] = 3
//tab[1][1] = 4
```

Prêt !!

Ex00 :

Ecrire une fonction qui remplit un tableau tab formé de n entiers. Cette fonction doit aussi afficher les valeurs de ce tableau

Elle devra être prototypée de la façon suivante :



```
lire_affiche  
  
void lire_affiche(int n);
```

Ex01 :

Ecrire une fonction qui retourne le nombre de fois de répétition d'une valeur entière x cherchée dans un tableau tab de n entiers.

Elle devra être prototypée de la façon suivante :



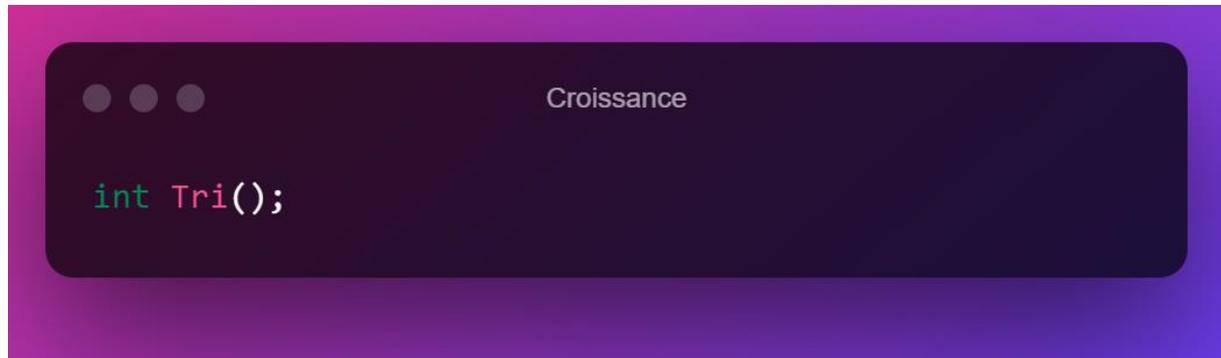
```
recherche_seq  
  
int recherche_seq(int x, int n);
```

Le remplissage et affichage du tableau se fait avec la **fonction** d'ex00

EX02 :

Ecrire une fonction qui trie un tableau Tb formé de n entiers.

Elle devra être prototypée de la façon suivante :

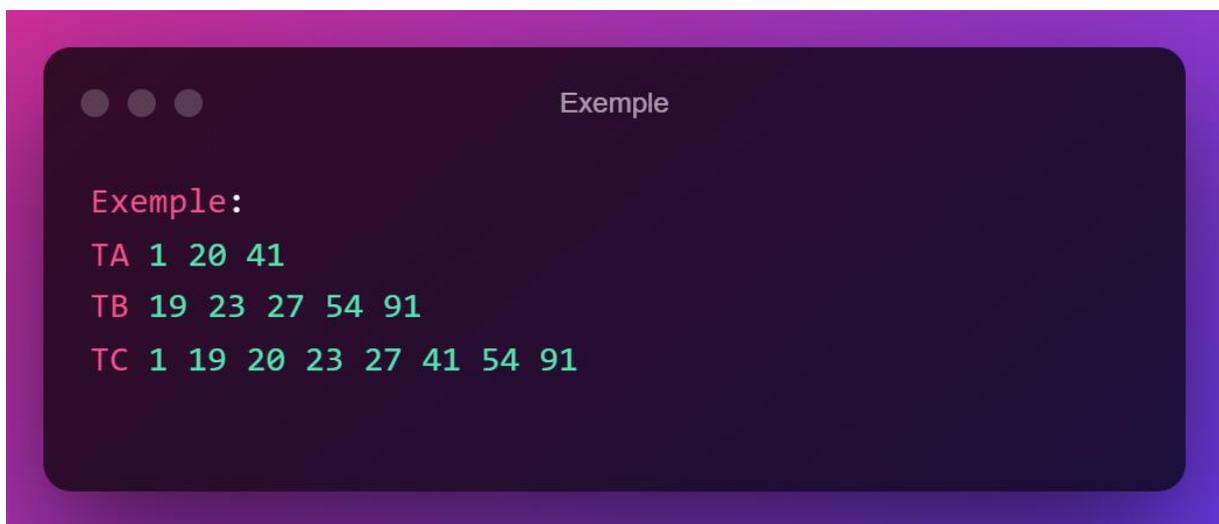
A screenshot of a code editor window with a purple gradient background. The window title is "Croissance". In the top-left corner, there are three small grey circles representing window control buttons. The main area of the editor contains the C function prototype: 

```
int Tri();
```

**EX03 :**

Ecrire un programme qui permet de fusionner deux tableaux triés TA et TB contenant respectivement n et m éléments. Le résultat est un tableau trié TC avec (n+m) éléments. Utiliser trois indices IA, IB et IFUS.

Comparer TA[IA] et TB[IB]; remplacer TC[IFUS] par le plus petit des deux éléments (appeler la fonction précédente) ; avancer dans le tableau TC et dans le tableau qui a contribué son élément. Lorsque l'un des deux tableaux TA ou TB est épuisé, il suffit de recopier les éléments restants de l'autre tableau dans le tableau TC.



```
Exemple:
TA 1 20 41
TB 19 23 27 54 91
TC 1 19 20 23 27 41 54 91
```

**EX04 :**

Ecrire une fonction qui retourne le minimum, le maximum et la moyenne des valeurs d'un tableau T passé en argument.

Elle devra être prototypée de la façon suivante :



```
int Min_Max_Mean(int tab[]);
```

**EX05 :**

Ecrire une fonction qui renvoie la somme, la soustraction, le produit des valeurs de 2 tableaux tab1 et tab2 :

Elle devra être prototypée de la façon suivante :



```
void Som_Sous_Prod(int tab1[], int tab2[]);
```

**EX06 :**

Ecrire un programme permettant de

- Remplir une matrice ;
- Afficher une matrice ;
- Calculer la somme de deux matrices M1 et M2 ;
- Calculer le produit de deux matrices M1 et M2 ;
- Calculer la transposition d'une matrice.

EX07 : (les chaînes de caractères )

Écrire un programme qui contient l'identique des fonctions suivantes :

Lire et afficher une chaîne de caractères ;

toupper() pour afficher la chaîne en majuscule ;

tolower() pour afficher la chaîne en minuscule ;

EX08 :

Écrire une fonction ft\_firstword qui affiche le premier mot d'une phrase donnée en chaîne de caractères .

Ex :

- " Bonjour " affiche "Bonjour"
- " Salut cv " affiche "Salut"
- " Salut,cv " affiche "Salut,cv"

EX10 :

Reproduire l'identique de la fonction atoi() **asciitointeger**

Elle devra être prototypée de la façon suivante :

```
ft_atoi()  
  
int ft_atoi(char chaine);
```

# ASCII TABLE

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]