

## TP 5 : CHAINES DE CARACTÈRES

---

### TABLE DES MATIÈRES

But	1
Exercice 1 : palindrome	1
Exercice 2 : conversion de casse	1
Exercice 4 : statistiques	2
Annexe	3

---

### BUT

Vous devez maîtriser à la fin de cette séance la manipulation des chaînes de caractères (et donc des tableaux...).

### EXERCICE 1 : PALINDROME

On rappelle qu'un palindrome est un mot qui peut se lire dans les deux sens. Par exemple : kayak, été, laval, elle, serres, erre,... L'objectif de déterminer si un mot est ou non un palindrome.

- (1) Écrire une fonction `int est_palindrome(char *mot)` qui renvoie 1 si la chaîne de caractère passée en paramètre est bien un palindrome et 0 sinon.
- (2) Écrire un `main` qui demande une chaîne de caractère à l'utilisateur, et dit si c'est ou pas un palindrome.

### EXERCICE 2 : CONVERSION DE CASSE

On veut pouvoir modifier la casse d'une chaîne de caractère, c'est-à-dire transformer les majuscules en minuscules et vice-versa.

- (1) Écrire une fonction `void inverse_casse(char *source, char *dest)` qui renvoie dans la chaîne `dest` la chaîne `source` dont la casse est strictement inversée.
- (2) Écrire une fonction `void vers_majuscule(char *source, char *dest)`, qui transforme la chaîne `source` en une chaîne exclusivement composée de majuscule
- (3) Écrire une fonction `void vers_minuscule(char *source, char *dest)`, qui transforme la chaîne `source` en une chaîne exclusivement composée de minuscule
- (4) Écrire un `main` associé.

## EXERCICE 4 : STATISTIQUES

Écrire un programme qui demande à l'utilisateur de taper un texte au clavier, puis qui calcule quelques statistiques élémentaires sur ce texte :

- Lettre la plus (resp. moins) fréquente.
- Nombre de lettres minuscules (resp. majuscules).
- Nombre de caractères non alphabétiques (chiffres, ponctuations,...).
- Nombre de mots (un mot étant défini comme une chaîne de caractères situés entre deux espaces ou ponctuations).
- Longueur moyenne des mots.

(1) Écrire les fonctions élémentaires dont vous avez besoin.

(2) Écrire un `main` associé.

## ANNEXE

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	□

FIGURE 1. Table ASCII. Le caractère NULL est le caractère de fin de chaîne '\0'.