

Codage de l'information

Cours magistral N° 3:

Présentation de l'information

Objectifs

Comprendre comment les ordinateurs

- représentent une information (nombre, caractère, image, son etc.)
- convertissent des entiers ou des nombres à virgule flottante en représentation binaire et *vice versa*
- réalisent des opérations mathématiques de base (addition, soustraction et multiplication)

Systeme decimal

Dix chiffres différents de 0 à 9 pour écrire tous les nombres.
Soit un nombre décimal $N = 2348$. Ce nombre est la somme de 8 unités, 4 dizaines, 3 centaines et 2 milliers.
Nous pouvons écrire

$$N = (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (8 \times 10^0)$$

10 représente la base et les puissances de 0 à 3 le rang de chaque chiffre.

Systeme binaire

Dans les domaines de l'automatisme, de l'électronique et de l'informatique, nous utilisons la **base 2 (0 et 1)**

Un interrupteur est ouvert ou fermé

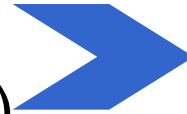
- Une diode est allumée ou éteinte
- Une tension est présente ou absente
- Une surface est réfléchissante ou pas (CD)
- Un champ magnétique est orienté Nord-Sud ou Sud-Nord (disque dur)



A chaque état du système technologique, on associe un état logique (binaire).

Codage

Le chiffre binaire qui peut prendre ces deux états est nommé "Bit" (Binary digit)



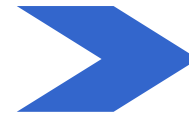
1

Avec un bit nous pouvons coder **deux** états



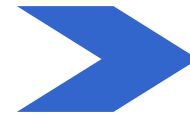
0
1

Avec deux bits nous pouvons coder **quatre** états



0	0
0	1
1	0
1	1

Avec trois bits nous pouvons coder **huit** états



0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Codage

A chaque nouveau bit, le nombre de combinaisons possibles est doublé.

Ce nombre est égal à **2 puissance N** (N étant le nombre de bits).

Un groupe de **bits** est appelé un **mot**, un mot de huit bits est nommé un **octet (byte)**.

0	1	0	0	0	1	0	1
---	---	---	---	---	---	---	---

Avec un octet, nous pouvons écrire 2 puissance 8 = **256 nombres binaires de 0 à 255**

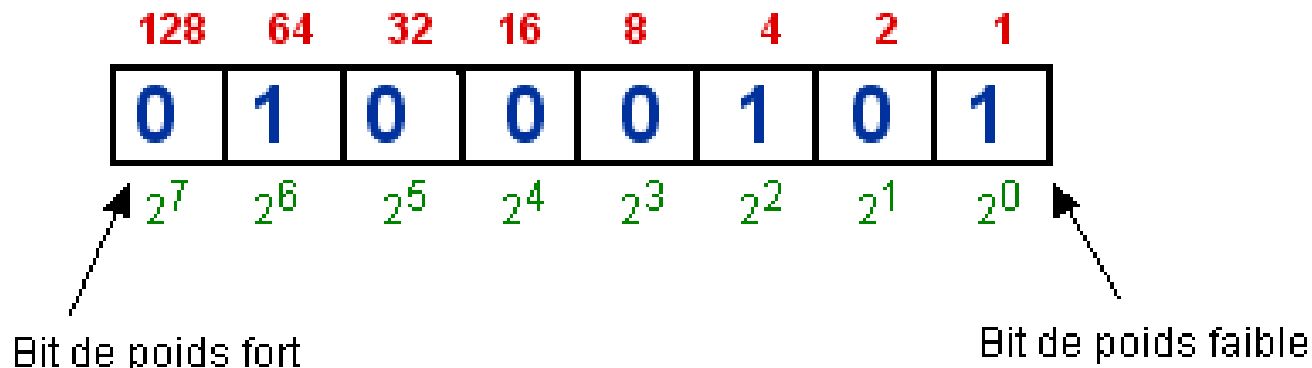
Codage

$$1011_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$$

$$1011_2 = (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1)$$

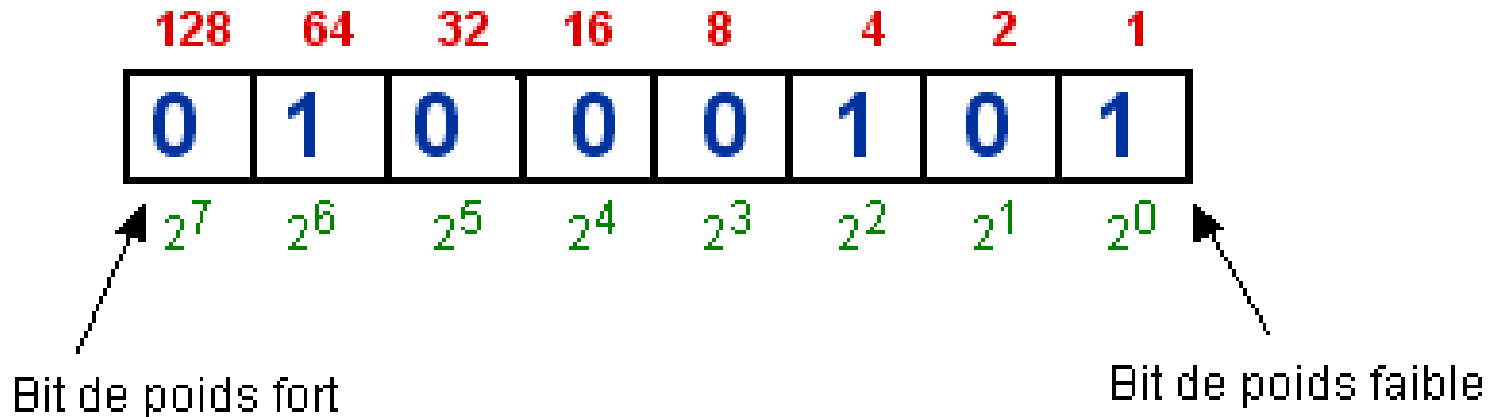
$$1011_2 = 11_{10}$$

Description d'un octet.



Correspondance entre binaire et décimal.

Conversion d'un nombre binaire en décimal



Il suffit donc de faire la somme des poids de chaque bit à 1
Le nombre ci dessus est égal à $64 + 4 + 1 = 69$

Conversion d'un nombre décimal (entier) en binaire

Exemple : Conversion d'un nombre décimal en binaire (exemple : N = 172)

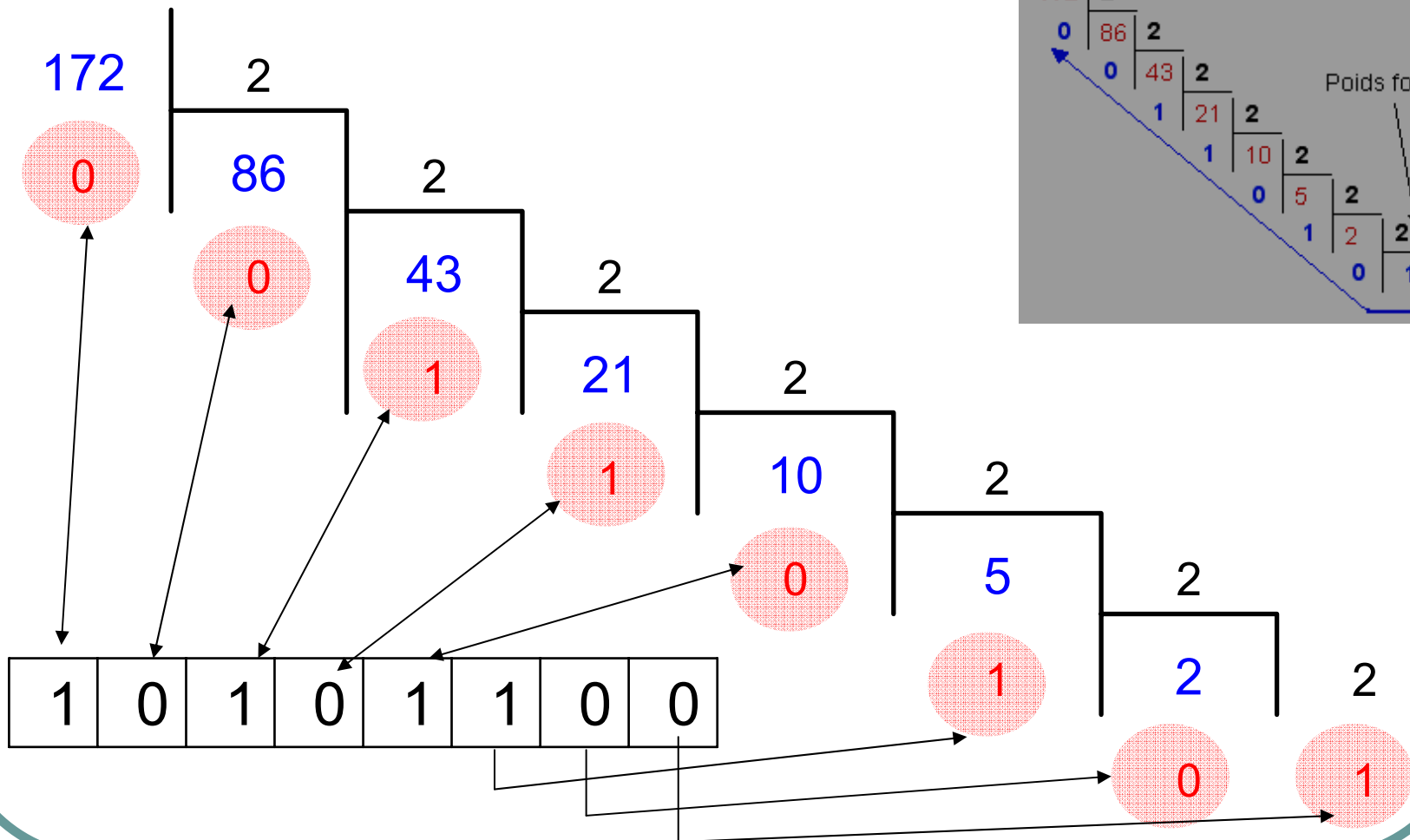
Méthode par soustraction

$$\begin{array}{r} 172 \\ - 128 \\ \hline 44 \end{array} \quad \begin{array}{r} 44 \\ - 32 \\ \hline 12 \end{array} \quad \begin{array}{r} 12 \\ - 8 \\ \hline 4 \end{array} \quad \begin{array}{r} 4 \\ - 4 \\ \hline 0 \end{array}$$

$$172 = 128 + 32 + 8 + 4$$

$$172_{(10)} = 10101100_{(2)}$$

Méthode par divisions



Conversion d'un nombre décimal (avec virgule) en binaire

Exemple 1 : 0.625

$$0.625 * 2 = 1.250 \text{ poids } 1 * 2^{-1}$$

$$0.250 * 2 = 0.500 \text{ poids } 0 * 2^{-2}$$

$$0.500 * 2 = 1.000 \text{ poids } 1 * 2^{-3}$$

$$\text{On a donc } (0.625)_{10} = (0.101)_2$$

Exemple 2 : 12.625

$$(12)_{10} = (1100)_2$$

et

$$(0.625)_{10} = (0.101)_2$$

$$\Rightarrow (12.625)_{10} = (1100.101)_2$$

Exemple 2 : 0.325

$$0.325 * 2 = 0.650$$

$$\text{poids } 1 * 2^{-1}$$

$$0.650 * 2 = 1.300$$

$$\text{poids } 0 * 2^{-2}$$

$$0.300 * 2 = 0.600$$

$$\text{poids } 0 * 2^{-3}$$

$$0.600 * 2 = 1.200$$

$$\text{poids } 1 * 2^{-4}$$

$$0.200 * 2 = 0.400$$

$$\text{poids } 0 * 2^{-5}$$

$$0.400 * 2 = 0.800$$

$$\text{poids } 0 * 2^{-6}$$

$$0.800 * 2 = 1.600$$

$$\text{poids } 1 * 2^{-7}$$

$$0.600 * 2 = 1.200$$

$$\text{poids } 1 * 2^{-8}$$

$$0.200 * 2 = 0.400$$

$$\text{poids } 0 * 2^{-9}$$

$$0.400 * 2 = 0.800$$

$$\text{poids } 0 * 2^{-10}$$

$$0.800 * 2 = 1.600$$

$$\text{poids } 1 * 2^{-11}$$

On a donc $(0.322)_{10} = (0.010\ 1001\ 1001\ 1001)_2$

Codage hexadécimal

La manipulation des nombres écrits en binaire est une opération fastidieuse en raison de la taille des codes obtenus. Il serait donc judicieux d'utiliser un autre système qui permet de réduire la longueur de ces codes. C'est pourquoi nous utilisons de préférence le système hexadécimal (base 16).

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Codage hexadécimal

Les règles sont ici aussi les mêmes que pour le décimal.

$$A3F_{(16)} = (A \times 16^2) + (3 \times 16^1) + (F \times 16^0)$$

$$A3F_{(16)} = (10 \times 256) + (3 \times 16) + (15 \times 1)$$

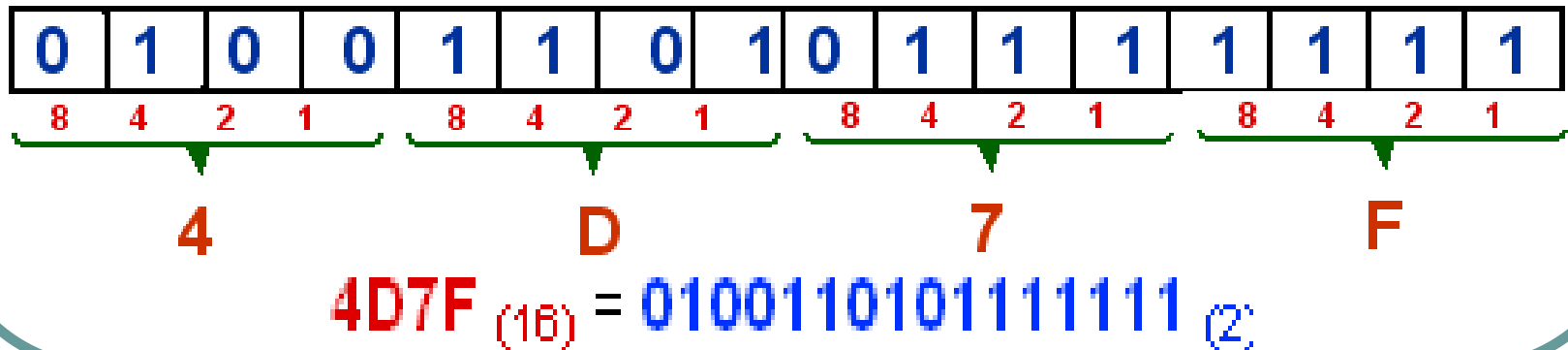
$$A3F_{(16)} = 2560 + 48 + 15 = 2623_{(10)}$$

Correspondance entre binaire et hexadécimal

La conversion du binaire en hexadécimal est très simple, c'est d'ailleurs la raison pour laquelle nous utilisons cette base.

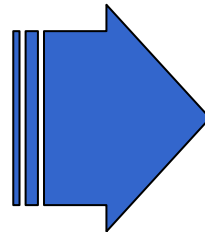
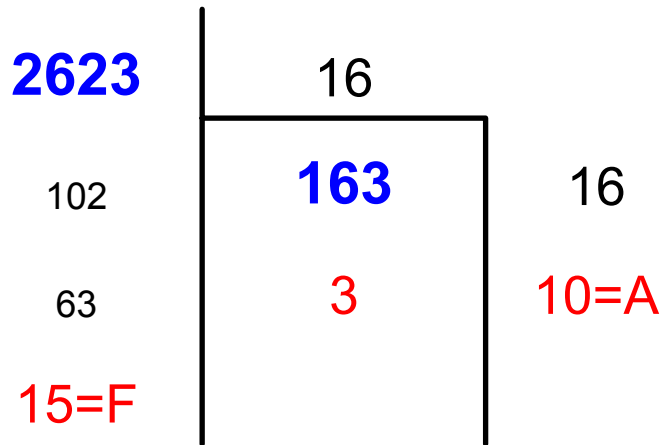
Il suffit de faire correspondre un mot de quatre bits (quartet) à chaque chiffre hexadécimal.

Conversion d'un mot de 16 bits entre binaire et hexadécimal

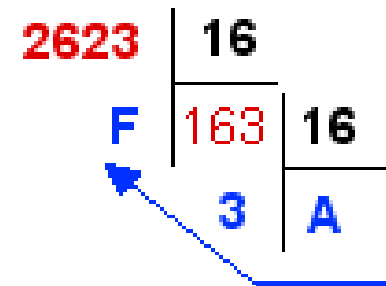


Correspondance entre décimal et hexadécimal

La méthodes par divisions s'applique comme en binaire (exemple : $N = 2623$).

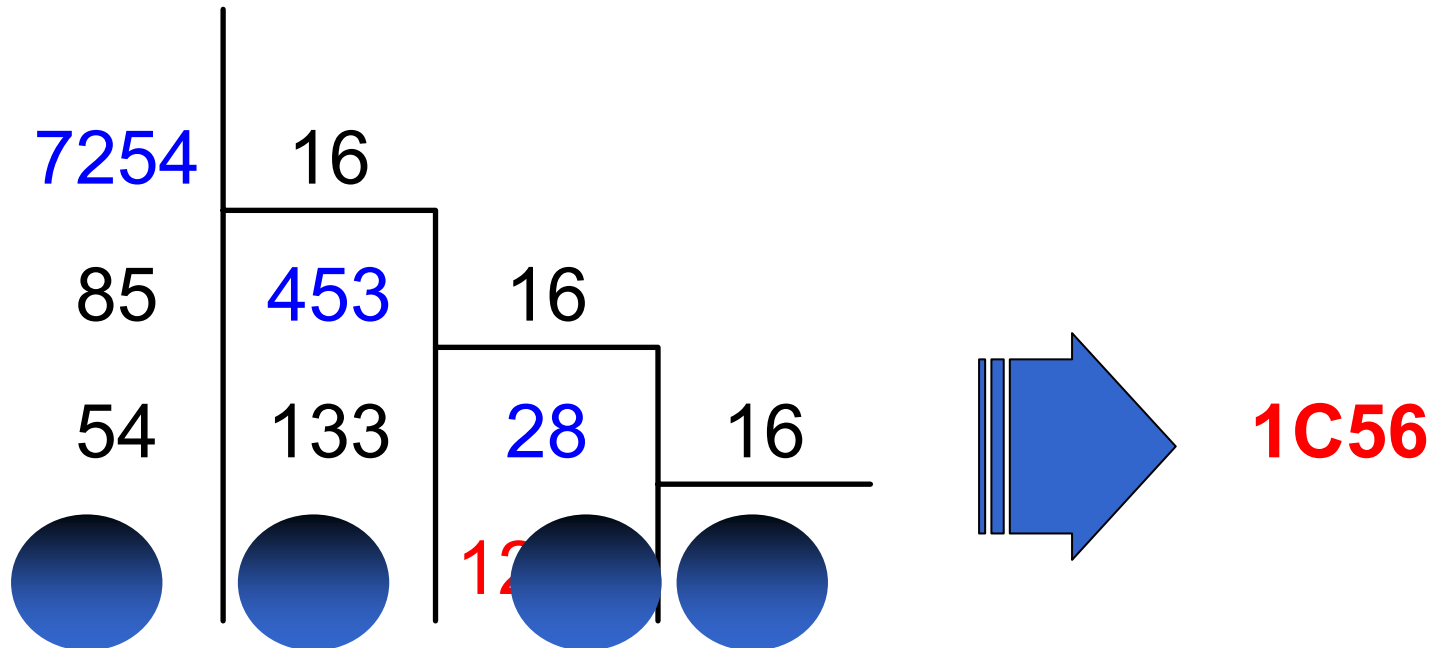


$$(A3F)_{16} = (2623)_{10}$$



$$(2623)_{10} = (A3F)_{16}$$

Exemple 2 : 7254



$$(1C56)_{16} = (7254)_{10}$$

Opérations arithmétiques et logiques

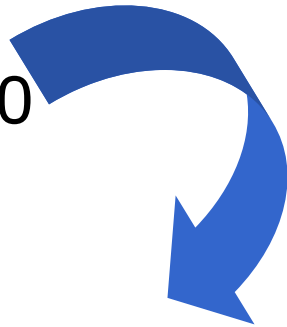
Addition en binaire

L'addition est réalisée bit à bit.

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$



10 en binaire correspond à 2 en décimal

Exemple

45

0	0	1	0	1	1	0	1
---	---	---	---	---	---	---	---

+

55

0	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---

=

100

0	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---

Produit logique en binaire

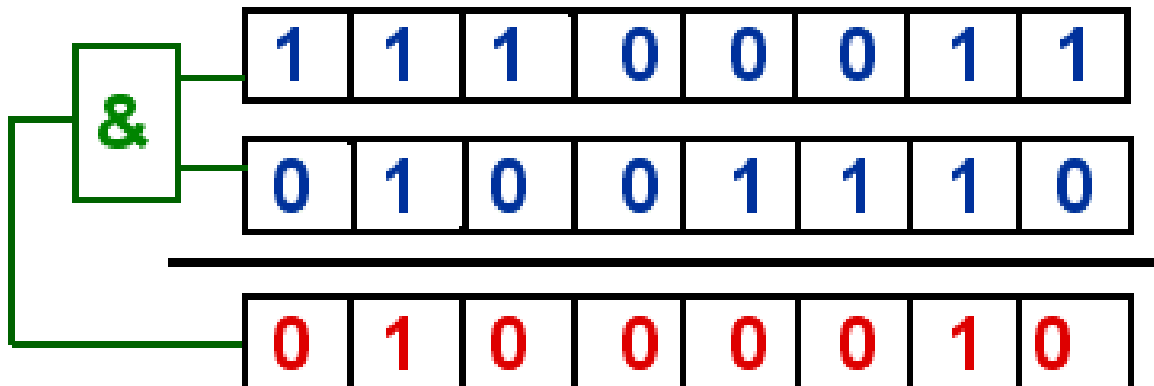
La fonction **ET** (&) est appliquée bit à bit

$$0 * 0 = 0$$

$$0 * 1 = 0$$

$$1 * 0 = 0$$

$$1 * 1 = 1$$



Codage ASCII

Pour coder les caractères, on associe à chacun d'entre eux un code binaire, c'est le codage ASCII (American Standard Code for Information Interchange).

Le caractère **A** code **65** soit **01000001** en binaire.

Le caractère **f** code **102** soit **001100110** en binaire

le point d'interrogation **?** code **63** soit **00111100** en binaire

Le chiffre **2** code **50** soit **00110010** en binaire

Table des codes ASCII

c	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	00	00	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	00	01	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	00	02	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	00	03	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	00	04	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	00	05	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	00	06	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	00	07	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	01	00	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	01	01	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
A	01	02	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
B	01	03	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
C	01	04	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
D	01	05	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
E	01	06	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
F	01	07	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
10	02	00	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
11	02	01	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
12	02	02	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
13	02	03	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
14	02	04	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
15	02	05	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
16	02	06	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
17	02	07	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
18	03	00	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
19	03	01	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
1A	03	02	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
1B	03	03	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
1C	03	04	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
1D	03	05	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
1E	03	06	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
1F	03	07	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		

ASCII étendu

28	Ç	144	É	161	í	177	⋯	193	⊥	209	⌞	225	β	241	±
29	ü	145	æ	162	ó	178	⋮	194	⌞	210	⌞	226	Γ	242	≥
30	é	146	Æ	163	ú	179		195	⌞	211	⌞	227	π	243	≤
31	â	147	ô	164	ñ	180	⌞	196	—	212	⌞	228	Σ	244	(
32	ä	148	ö	165	Ñ	181	⌞	197	⌞	213	⌞	229	σ	245)
33	à	149	ò	166	ª	182	⌞	198	⌞	214	⌞	230	μ	246	÷
34	ã	150	û	167	º	183	⌞	199	⌞	215	⌞	231	τ	247	∞
35	ç	151	ù	168	¿	184	⌞	200	⌞	216	⌞	232	Φ	248	°
36	ê	152	—	169	—	185	⌞	201	⌞	217	⌞	233	⊙	249	.
37	ë	153	Ö	170	¬	186	⌞	202	⌞	218	⌞	234	Ω	250	.
38	è	154	Ü	171	½	187	⌞	203	⌞	219	■	235	δ	251	√
39	ï	156	£	172	¼	188	⌞	204	⌞	220	■	236	∞	252	—
40	î	157	¥	173	¡	189	⌞	205	=	221	■	237	φ	253	²
41	ì	158	—	174	«	190	⌞	206	⌞	222	■	238	ε	254	■
42	Ä	159	ƒ	175	»	191	⌞	207	⌞	223	■	239	∩	255	
43	Å	160	á	176	⋯	192	⌞	208	⌞	224	α	240	≡		