I– LE SYSTÈME DÉCIMAL

Notre mode de numération est le système *décimal*, ou système de base 10.
Chaque nombre s'écrit avec les 10 chiffres `0 1 2 3 4 5 6 7 8 9`.
Chaque chiffre correspond à un rang, qui est celui d'une puissance de 10.

Prenons par exemple le nombre en écriture décimale 5603_d (le petit $_d$ en indice est là pour rappeler que le nombre est écrit en écriture décimale).
Le chiffre `5` est dans le rang des milliers, donc de 10^3.
On peut donc écrire :

$$5603_d = 5 \times 10^3 + 6 \times 10^2 + 0 \times 10^1 + 3 \times 10^0$$

système décimal								
puissance	...	10^n	...	10^4	10^3	10^2	10^1	10^0
chiffre					5	6	0	3

Ce système de base 10 provient (très probablement) du fait que les humains ont dix doigts.
De cette manière, on pourrait imaginer que dans leur univers, les Simpsons compteraient en base 8 car ils n'ont que 4 doigts à chaque main.

Exercice 1
Dans l'univers des Simpsons, que vaut le nombre 63_s ?

II– LE SYSTÈME BINAIRE

Les appareils électroniques (ordinateurs, calculatrices, téléphones, etc.) travaillent avec un système de représentation des nombres différent, le système *binaire*. Dans ce système, appelé aussi *système de base 2*, chaque nombre s'écrit avec les deux chiffres `0` et `1` (correspondant en fait au passage ou non d'un courant électrique dans un transistor). Chaque chiffre correspond à un rang, qui est celui d'une puissance de 2.

▶ du binaire au décimal :
Prenons par exemple le nombre en écriture binaire 1101_b
Pour connaître la valeur de 1101_b en écriture décimale on peut donc s'aider du tableau :

système binaire								
puissance	...	2^n	...	2^4	2^3	2^2	2^1	2^0
chiffre					1	1	0	1

$$\begin{aligned} 1101_b &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\ &= 8 + 4 + 0 + 1 \\ &= 13_d \end{aligned}$$

Donc le nombre `1101` écrit en binaire correspond au nombre `13` écrit en décimal.

Exercice 2

1. Donner l'écriture décimale du nombre 10111_b.
2. Quelle est la valeur (en écriture décimale) du plus grand nombre binaire écrit sur un octet ? (un octet = 8 bits)
3. Expliquer cette blague d'informaticiens :
   « Dans la vie, il y a 10 sortes de personnes : celles qui connaissent le binaire, et celles qui ne le connaissent pas. »
4. Que devient l'écriture binaire d'un nombre lorsqu'on le multiplie par 2 ?

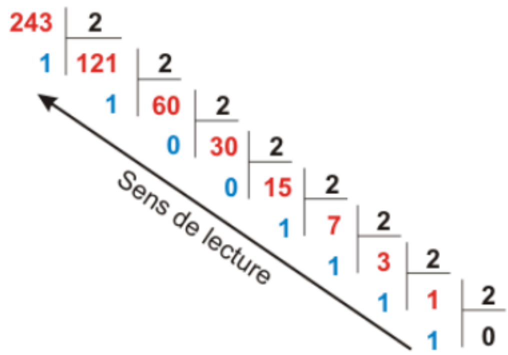
▶ **du décimal au binaire :**

Principe : dans chaque nombre décimal, il existe une plus grande puissance de 2 qui est inférieure au nombre. Par exemple, dans 243, il y a 128. Donc

$$\begin{aligned}
243 &= 128 + (115) \\
&= 128 + 64 + (51) \\
&= 128 + 64 + 32 + (19) \\
&= 128 + 64 + 32 + 16 + (3) \\
&= 128 + 64 + 32 + 16 + 2 + 1 \\
&= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0
\end{aligned}$$

Donc $243_{10} = 11110011_2$

Astuce : méthode des divisions successives



Exercice 3 Donner l'écriture binaire du nombre 28_d.

Exercice 4

Hex	Bin	Char	Hex	Bin	Char	Hex	Bin	Char	Hex	Bin	Char
0x00	00000000	NUL	0x20	00100000	space	0x40	01000000	@	0x60	01100000	`
0x01	00000001	SOH	0x21	00100001	!	0x41	01000001	A	0x61	01100001	a
0x02	00000010	STX	0x22	00100010	"	0x42	01000010	B	0x62	01100010	b
0x03	00000011	ETX	0x23	00100011	#	0x43	01000011	C	0x63	01100011	c
0x04	00000100	EOT	0x24	00100100	$	0x44	01000100	D	0x64	01100100	d
0x05	00000101	ENQ	0x25	00100101	%	0x45	01000101	E	0x65	01100101	e
0x06	00000110	ACK	0x26	00100110	&	0x46	01000110	F	0x66	01100110	f
0x07	00000111	BEL	0x27	00100111	'	0x47	01000111	G	0x67	01100111	g
0x08	00001000	BS	0x28	00101000	(	0x48	01001000	H	0x68	01101000	h
0x09	00001001	TAB	0x29	00101001	)	0x49	01001001	I	0x69	01101001	i
0x0A	00001010	LF	0x2A	00101010	*	0x4A	01001010	J	0x6A	01101010	j
0x0B	00001011	VT	0x2B	00101011	+	0x4B	01001011	K	0x6B	01101011	k
0x0C	00001100	FF	0x2C	00101100	,	0x4C	01001100	L	0x6C	01101100	l
0x0D	00001101	CR	0x2D	00101101	-	0x4D	01001101	M	0x6D	01101101	m
0x0E	00001110	SO	0x2E	00101110	.	0x4E	01001110	N	0x6E	01101110	n
0x0F	00001111	SI	0x2F	00101111	/	0x4F	01001111	O	0x6F	01101111	o
0x10	00010000	DLE	0x30	00110000	0	0x50	01010000	P	0x70	01110000	p
0x11	00010001	DC1	0x31	00110001	1	0x51	01010001	Q	0x71	01110001	q
0x12	00010010	DC2	0x32	00110010	2	0x52	01010010	R	0x72	01110010	r
0x13	00010011	DC3	0x33	00110011	3	0x53	01010011	S	0x73	01110011	s
0x14	00010100	DC4	0x34	00110100	4	0x54	01010100	T	0x74	01110100	t
0x15	00010101	NAK	0x35	00110101	5	0x55	01010101	U	0x75	01110101	u
0x16	00010110	SYN	0x36	00110110	6	0x56	01010110	V	0x76	01110110	v
0x17	00010111	ETB	0x37	00110111	7	0x57	01010111	W	0x77	01110111	w
0x18	00011000	CAN	0x38	00111000	8	0x58	01011000	X	0x78	01111000	x
0x19	00011001	EM	0x39	00111001	9	0x59	01011001	Y	0x79	01111001	y
0x1A	00011010	SUB	0x3A	00111010	:	0x5A	01011010	Z	0x7A	01111010	z
0x1B	00011011	ESC	0x3B	00111011	;	0x5B	01011011	[	0x7B	01111011	{
0x1C	00011100	FS	0x3C	00111100	<	0x5C	01011100	\	0x7C	01111100	|
0x1D	00011101	GS	0x3D	00111101	=	0x5D	01011101	]	0x7D	01111101	}
0x1E	00011110	RS	0x3E	00111110	>	0x5E	01011110	^	0x7E	01111110	~
0x1F	00011111	US	0x3F	00111111	?	0x5F	01011111	_	0x7F	01111111	DEL

Décoder la phrase ci-dessous :

`74, 39, 65, 73, 77, 69, 32, 76, 69, 32, 66, 73, 78, 65, 73, 82, 69`