Cours

La Numération
(décimal, binaire, hexadécimal)

Lycée Aragon

Correction

Sommaire

1. Mise en situation **2**

2. Représentation d'un nombre **2**

2.1 Les systèmes de numération (de codage) *2*

2.2 Codage décimal *2*

2.3 Codage binaire *2*

2.3.1 Les différentes puissances de 2 3

2.3.2 Vocabulaire 3

2.3.3 Valeurs maximum et minimum représentées sur n bits 3

2.3.4 Notations des valeurs binaires : 3

2.4 Codage hexadécimal *3*

2.4.1 Les différentes puissances de 16 : 4

2.4.2 Notations des valeurs hexadécimales : 4

3. Changements de bases (transcodage) **4**

3.1 Tableau de correspondance *4*

3.2 Conversion base décimale / base binaire *5*

3.2.1 Décimal en binaire 5

3.3 Conversion base Hexadécimale / base binaire *5*

3.3.1 Hexadécimal en binaire 5

3.3.2 Binaire en hexadécimal 5

3.4 Conversion base Hexadécimale / base décimale *6*

3.4.1 Décimal en hexadécimal 6

4. Codage du texte : le code ASCII **6**

1. Mise en situation

En information numérique nous sommes amenés à utiliser plusieurs bases numériques :

- **Décimale** : notre système de numération habituel.
- **Binaire** : la forme la plus « simple » de représenter une information et la plus facile à manipuler pour un système numérique (ordinateur,…).
- **Hexadécimal** : permet de manipuler (pour nous) plus « facilement » un code binaire.

2. Représentation d'un nombre

2.1 Les systèmes de numération (de codage)

On a l'habitude de compter l'argent, les quantités, les masses, les distances… dans la base 10. On peut décomposer par exemple 1324€ en $1324 = 1000 + 300 + 20 + 4 = 1 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$.

On peut généraliser les décompositions de la base 10 par :

$$(ABCD)_{10} = A \times 10^3 + B \times 10^2 + C \times 10^1 + D \times 10^0.$$

Cette généralisation fonctionne pour toutes les bases :

Base n $\boxed{(ABCD)_n = A \times n^3 + B \times n^2 + C \times n^1 + D \times n^0}$

Pour savoir la base dans laquelle on est, on l'affiche en indice.

2.2 Codage décimal

En base décimale, un nombre $N = (ABCD)_{10}$ s'exprime de la manière suivante :

$$\begin{aligned} \mathbf{N} &= \mathbf{A} * 10^3 + \mathbf{B} * 10^2 + \mathbf{C} * 10^1 + \mathbf{D} * 10^0 \\ &= \mathbf{A} * 1000 + \mathbf{B} * 100 + \mathbf{C} * 10 + \mathbf{D} * 1 \end{aligned}$$

- Une telle base s'appelle **base décimale** car elle s'exprime en fonction des différentes puissances de 10, les chiffres utilisés allant de 0 à 9.
- Les différentes puissances de 10 s'appellent les **poids (base$^{\text{rang}}$)**.

	Milliers	Centaines	Dizaines	Unités
Chiffre	A	B	C	D
Rang	3	2	1	0
Poids	10^3	10^2	10^1	10^0

Exemple : $N = (2015)_{10} = \mathbf{2}*1000 + \mathbf{0}*100 + \mathbf{1}*10 + \mathbf{5}*1$

2.3 Codage binaire

De la même manière, on peut coder un nombre en base binaire, en utilisant les différentes puissances de 2 : on l'appelle aussi la **base 2**.

Dans ce système chaque poids est une **puissance de 2**, les valeurs des chiffres A, B, C, D,… sont donc **0** ou **1** (bit).

Exemple :

$$\begin{aligned} N = (10110)_2 &= \mathbf{1} * 2^4 + \mathbf{0} * 2^3 + \mathbf{1} * 2^2 + \mathbf{1} * 2^1 + \mathbf{0} * 2^0 \\ &= 16 + 4 + 2 \\ &= (22)_{10} \end{aligned}$$

2.3.1 Les différentes puissances de 2

n	10	9	8	7	6	5	4	3	2	1	0
2^n	1024	512	256	128	64	32	16	8	4	2	1

2.3.2 Vocabulaire

- **Quartet :** nombre binaire formé de 4 éléments binaires.
- **Octet (byte) :** nombre binaire formé de 8 éléments binaires.
- **Mot (word) :** nombre binaire formé de 16, 32 ou 64 éléments binaires.
- **L.S.B. (Least Significant Bit):** bit le moins significatif ou bit de poids faible (élément le plus à droite d'un nombre binaire).
- **M.S.B. (Most Significant Bit):** bit le plus significatif ou bit de poids fort (élément binaire le plus à gauche d'un nombre binaire)

Exemple : Nombre binaire codé sur un octet N = $(10110011)_2$

	MSB							LSB
Rang	7	6	5	4	3	2	1	0
Poids	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
Chiffre	1	0	1	1	0	0	1	1

2.3.3 Valeurs maximum et minimum représentées sur n bits

La valeur minimum d'un entier représenté sur n bits est 0 quel que soit le nombre d'éléments binaires. En utilisant n bits, on peut former 2^n nombres différents et le plus grand d'entre eux est égal à 2^n-1

Exemple : si n = 8 alors on peut former 256 nombres différents et $N_{max} = (2^8 - 1) = 255$

2.3.4 Notations des valeurs binaires :

Un nombre binaire peut être précédé du signe **%** ou suivi de l'indice de base (2) ou d'un B.

Exemple : % 10110011 = $(10110011)_2$ = 10110011 B

Applications :

1. Combien vaut ce nombre en base 10 ? N = $(128+32+16+2+1)_{10} = (179)_{10}$
   Que vaut le LSB ? LSB = 1
   Que vaut le MSB ? MSB = 1
2. Donnez le code binaire de 255 : $(255)_{10} = (11111111)_2$

2.4 Codage hexadécimal

Le système hexadécimal est de **base 16** et utilise 16 symboles différents :

- les dix premiers chiffres décimaux : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- suivis des 6 premières lettres de l'alphabet : A, B, C, D, E, F.

Les lettres A à F correspondent respectivement aux nombres décimaux 10 à 15.

Dans ce système, le poids est une puissance de 16.

Exemple :

$N = (AC53)_{16}$

$N = A * 16^3 + C * 16^2 + 5 * 16^1 + 3 * 16^0$

$N = 10 * 16^3 + 12 * 16^2 + 5 * 16^1 + 3 * 16^0$

$N = (44115)_{10}$

2.4.1 Les différentes puissances de 16 :

n	4	3	2	1	0
16^n	65536	4096	256	16	1

2.4.2 Notations des valeurs hexadécimales :

Un nombre hexadécimal peut être précédé du signe **$** ou suivi de l'indice de base (16) ou d'un H.

Exemple : \$ F6B1 = $(F6B1)_{16}$ = F6B1 H

Applications : Convertir en base 10 :

1. N = $(7DF)_{16}$ = $7*16^2 + 13*16^1 + 12*1$ = $7*256 + 13*16 + 15$
   = 2015
2. N = $(501)_{16}$ = $5*16^2 + 0*16^1 + 1*1$ = $5*256 + 1$
   = 1281
3. N = $(CAFE)_{16}$ = $12*4096 + 10*256 + 15*16 + 14$
   = 51966

Remarques :- *le premier permet de voir qu'une date peut ne pas forcément être codée comme on le pense*
- *le second permet de voir qu'il est nécessaire parfois d'indiquer la base (si aucuns chiffres de A à F dans le nombre)*
- *le troisième aussi, on serait tenté de lire « café » ...*

3. Changements de bases (transcodage)

3.1 Tableau de correspondance

Décimal		Binaire				Hexadécimal
10^1	10^0	2^3	2^2	2^1	2^0	16^0
	0	0	0	0	0	0
	1	0	0	0	1	1
	2	0	0	1	0	2
	3	0	0	1	1	3
	4	0	1	0	0	4
	5	0	1	0	1	5
	6	0	1	1	0	6
	7	0	1	1	1	7
	8	1	0	0	0	8
	9	1	0	0	1	9
1	0	1	0	1	0	A
1	1	1	0	1	1	B
1	2	1	1	0	0	C
1	3	1	1	0	1	D
1	4	1	1	1	0	E
1	5	1	1	1	1	F

3.2 Conversion base décimale / base binaire

3.2.1 Décimal en binaire

Première méthode : Décomposition en somme de puissances

Nous décomposons le nombre décimal en une somme de puissances de 2.

<u>*Exemple :*</u> Conversion de $N=(187)_{10}$ en nombre binaire.

Evaluation du nombre de bits : $128 < 187 < 256$ donc $2^7 < 187 < 2^8$ → **N sera codé sur 8 bits**

MSB LSB

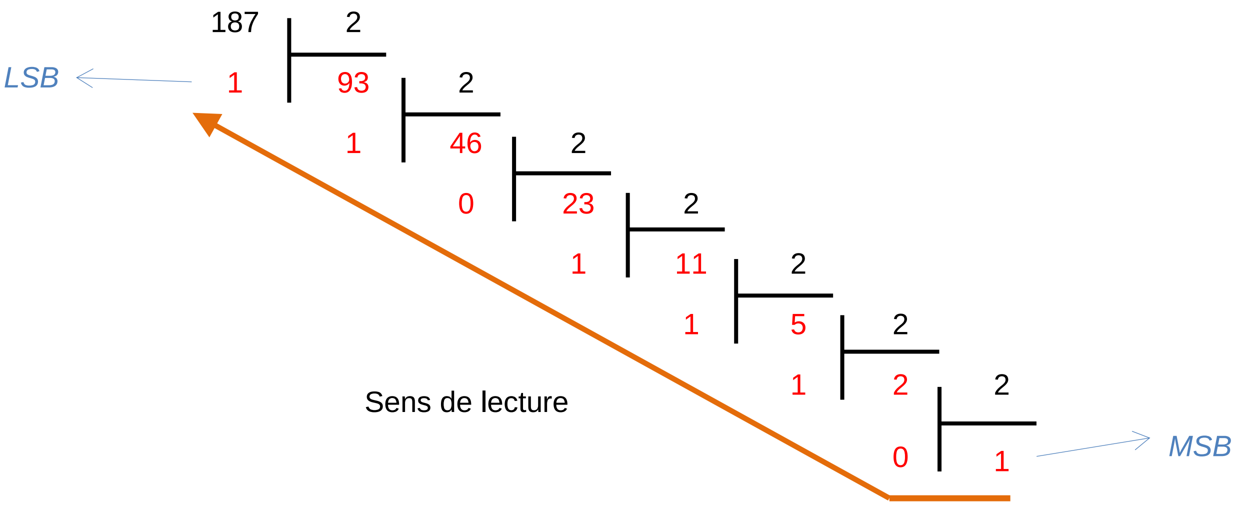
rang	7	6	5	4	3	2	1	0
poids	2^7=128	2^6=64	2^5=32	2^4=16	2^3=8	2^2=4	2^1=2	2^0=1
N	1	0	1	1	1	0	1	1

$N = (10111011)_2$

Deuxième méthode : méthode de la division

On obtient le nombre en base 2 par divisions successives par 2 du nombre. Le nombre en base 2 est constitué par les restes des divisions, la lecture s'effectuant du dernier reste vers le premier.

<u>*Exemple :*</u> Conversion de $N=(187)_{10}$ en nombre binaire.



$N = (10111011)_2$

3.3 Conversion base Hexadécimale / base binaire

3.3.1 Hexadécimal en binaire

Chaque symbole du nombre hexadécimal est remplacé par son équivalent écrit dans le système binaire.

<u>*Exemple :*</u> N = \$ B F 8

N = % 1011 1111 1000 = $(101111111000)_2$

B F 8

3.3.2 Binaire en hexadécimal

C'est l'inverse de la précédente. Il faut donc regrouper les 1 et 0 du nombre par quartet en commençant par la droite, puis chaque groupe est remplacé par le symbole hexadécimal correspondant.

<u>*Exemple :*</u> N = % 100001101111

N = % 1000 0110 1111

8 6 F

N = \$ 86F

3.4 Conversion base Hexadécimale / base décimale

3.4.1 *Décimal en hexadécimal*

Il convient d'appliquer la méthode des divisions successives :

Exemple : Conversion de N=$(124)_{10}$ en nombre hexadécimal.

$$\begin{array}{r|l} 124 & 16 \\ \cline{2-2} 12 & 7 \end{array}$$

N = $(7C)_{16}$ puisque $(12)_{10}$ correspond à $(C)_{16}$

Applications :

Convertir les nombres suivants en base décimale :

- A = $(0011)_2$ = 3
- B = $(1010)_2$ = 10
- C = $(80)_{16}$ = 128
- D = $(FF)_{16}$ = 255

Convertir les nombres suivants en base binaire (exprimer le nombre en octet) :

- E = $(80)_{10}$ = 01010000
- F = $(127)_{10}$ = 01111111
- G = $(80)_{16}$ = 10000000
- H = $(FF)_{16}$ = 11111111

Convertir les nombres suivants en base hexadécimale :

- I = $(10100101)_2$ = A5 H
- J = $(11001110)_2$ = CE H
- K = $(80)_{10}$ = 50 H
- L = $(135)_{10}$ = 87 H

Remarque : En conclusion, montrer la conversion avec « Calc.exe » sous Windows

4. Codage du texte : le code ASCII

Le codage ASCII (American Standard Code for Information Interchange) est devenu au fil du temps le standard pour coder les informations alphanumériques et autres caractères de commande. C'est un codage sur sept bits (valeurs de 0 à 127) qui permet de définir :

- Des caractères imprimables universels : lettres majuscules et minuscules, chiffres, symbole de ponctuation…
- Des caractères non imprimables : saut de ligne, fin de texte…

Codage ASCII

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]

	LSQ	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
MSQ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
0001	1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EN	SUB	ESC	FS	GS	RS	US
0010	2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0011	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

<u>Exemple :</u> 0010 0000 est le caractère « espace » (séparateur de mots)

Pour coder de manière universelle l'ensemble des symboles utilisés quelle que soit la langue, il faut attribuer à tout caractère **un nom et un identifiant numérique**, et ce de manière unifiée, quelle que soit la plate-forme informatique ou le logiciel.
C'est ce que propose la norme unicode (développée par le consortium du même nom, unicode.org)

<u>Exemple</u> : A « lettre majuscule latine A » U +0041
é « lettre minuscule latine E accent aigu » U +00E9
€ « symbole euro » U +20AC

Actuellement, un des systèmes d'encodage est le UTF-8 (unicode)

<u>Applications :</u>

Décrypter la chaîne ASCII suivante représentée sous forme d'une suite d'octets :

0101 0011 0101 0100 01001001 0010 0000 0011 0010 0100 0100 0000 1010

53 = S 54 =T 49 = I 20 = espace 32=2 44 = D 0A = LF *(line feed → fin de ligne et retour à la ligne)*