Chapitre II	**SYSTEME DE NUMERATION ET CODES**	
Leçon 1	**Système de numération**	Le .../..../201...

Prof : **RHIMI Jalel**

I- Mise en situation

Activités de découverte (MA page 16 et 17).

1) N = 4282
2) - 90 s'écrivait : 1.30
   - 45 s'écrivait : 0.45
   - 30 s'écrivait : 0.30

   N = 2423968
3)

MMCIII	LX	XXI	XIV	MMVI	LXVIII
2103	60	21	14	2006	68

II- Définitions :

1- Système de numération :

Un système de numération est une façon d'énoncer ou d'écrire des nombres. De nombreux systèmes de numération sont utilisés en technologie numérique. Les plus courants sont les systèmes décimal, binaire, octal et hexadécimal.

2- Base d'un système de numération :

La base d'un système de numération est le nombre de chiffres différents qu'utilise ce système de numération.

3- Système décimal:

Le système décimal est le système que nous utilisons tous les jours. Il comprend dix chiffres différents qui sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Soit le nombre 3276 de ce système; nous l'écrivons N = $(3276)_{10}$.

Ce nombre N peut être écrit sous la forme du polynôme suivant

$$\begin{aligned} N &= 3 \times 10^3 + 2 \times 10^2 + 7 \times 10^1 + 6 \times 10^0 \\ &= 3 \times 1000 + 2 \times 100 + 7 \times 10 + 6 \times 1 \\ &= 3000 + 200 + 70 + 6 \\ &= 3276 \end{aligned}$$

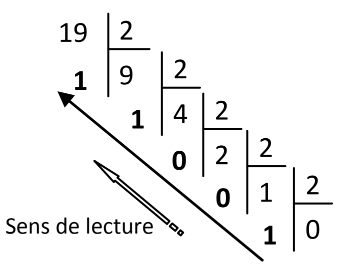
4- Système binaire:

Le système de numération binaire, système de base deux, n'utilise que les deux symboles 0 et 1. (A une grandeur physique qui ne peut prendre que deux états on associe les symboles 0 et 1). Les signaux véhiculés entre les différents composants d'un PC sont numériques. En effets les opérations avec dix chiffres (base 10) sont trop lentes à effectuer par les processeurs d'un PC.

Les composants d'un PC effectuent très rapidement et très simplement des opérations sur des nombres comportant uniquement deux éléments 0 et 1 appelés ***bits***.

- **Conversion d'un nombre écrit en décimale en un nombre écrit en binaire.**

Exemple 1 : Soit à convertir le nombre 19 en binaire.



$(19)_{10} = (10011)_2$ **:** Cette conversion s'appelle **le codage.**

L'opération inverse permet de convertir un nombre binaire en un nombre décimal :

$$(10011)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1$$

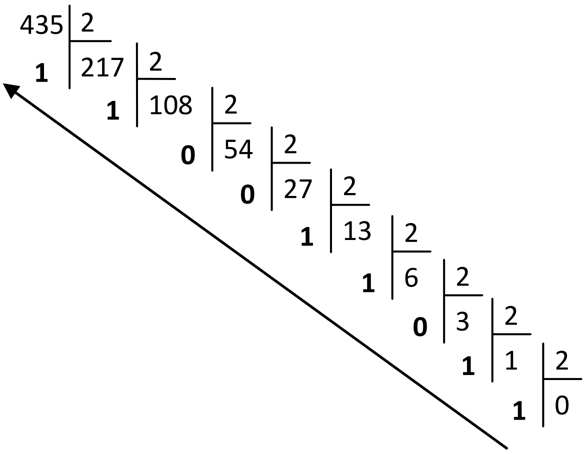
$$16 + 2 + 1 = 19$$

Un bit

Rq $\underbrace{10011}_{\text{Un mot binaire}}$

$(10011)_2 = (19)_{10}$: Cette opération s'appelle **le décodage**

Exemple 2 : Ecrire $(435)_{10}$ en base deux.



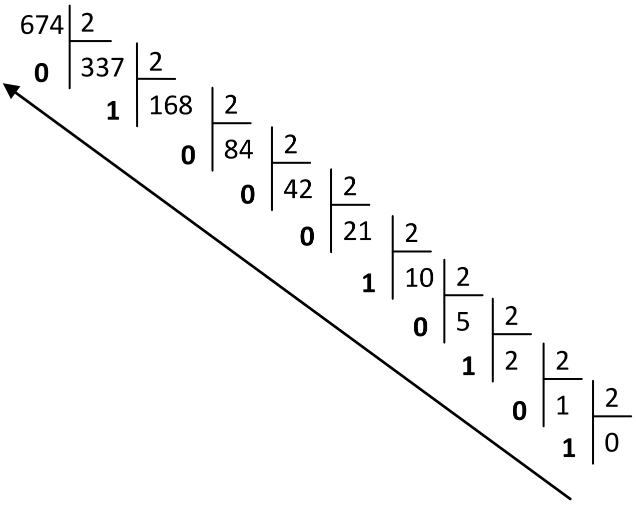
$(435)_{10} = (110110011)_2$

Le décodage :

$(110110011)_2 = 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^1 + 1 \times 2^0$

$256 + 128 + 32 + 16 + 2 + 1 = (435)_{10}$

Exemple3 : Ecrire $(674)_{10}$ en base deux.



$(674)_{10} = (1010100010)_2$

Le décodage :

$(1010100010)_2 = 1 \times 2^9 + 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^1$

$512 + 128 + 32 + 2 = (674)_{10}$

Activités pratiques (MA page 18)

5- Système octal:

C'est un système de base huit. Ses symboles sont au nombre de 8 (0,1,2,3,4,5,6,7)

Exemple1 : Soit à coder le nombre $(272)_{10}$ en hexadécimal.

```
279 | 8
  7 | 34 | 8
       2 | 4 | 8
           4 | 0
```

$(279)_{10} = (427)_8$

Le décodage :

$(427)_8 = 4 \times 8^2 + 2 \times 8^1 + 7 \times 8^0$

$256 + 16 + 7 = (279)_{10}$

Exemple2 : Soit à coder le nombre $(2005)_{10}$ en octal.

```
2005 | 8
   5 | 250 | 8
         2 | 31 | 8
              7 | 3 | 8
                  3 | 0
```

$(2005)_{10} = (3725)_8$

Le décodage :

$(3725)_8 = 3 \times 8^3 + 7 \times 8^2 + 2 \times 8^1 + 5 \times 8^0$

$1536 + 448 + 16 + 5 = (2005)_{10}$

Exemple3 : Soit à décoder le nombre $(507)_8$.

$(507)_8 = 5 \times 8^2 + 7 \times 8^0$

$320 + 7 = (327)_{10}$

6- Système hexadécimal:

C'est un système de base seize. Ses symboles sont au nombre de 16 (0,1,2,3,4,5,6,7,8,9,A ,B,C,D,E,F)

Exemple1 : Soit à coder le nombre $(348)_{10}$ en hexadécimal.

```
348 | 16
 12 | 21 | 16
       5 | 1 | 16
           1 | 0
```

$(348)_{10} = (15C)_{16}$

Le décodage :

$(15C)_{16} = 1 \times 16^2 + 5 \times 16^1 + 12 \times 16^0$

$256 + 8$ [illegible]

Exemple2 : Soit à coder le nombre $(2512)_{10}$ en hexadécimal.

```
2512 | 16
  0  |157 | 16
      13  |  9 | 16
             9 | 0
```

$(2512)_{10} = (9D0)_{16}$

Le décodage :

$(9D0)_{16}$ = $9 \times 16^2 + 13 \times 16^1$

$2304 + 208$ = $(2512)_{10}$

Exemple3 : Soit à décoder le nombre $(2AF6D)_{16}$.

$(2AF6D)_{16}$ = $2 \times 16^4 + 10 \times 16^3 + 15 \times 16^2 + 6 \times 16^1 + 13 \times 16^0$
$131072 + 40960 + 3840 + 96 + 13$ = $(175981)_{10}$

III- Opérations sur les nombres binaires :

1- Addition :

L'addition binaire est analogue à l'addition décimale. Il faut commencer par le bit de poids le plus faible en utilisant l'algorithme suivant :

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 avec un report de 1

Exemple1 : soit à additionner $(21)_{10} + (17)_{10}$

$(21)_{10} = (10101)_2$; $(17)_{10} = (10001)_2$; $(38)_{10} = (100110)_2$

Vérification : 21 +17 = 38

```
  10101
+ 10001
-------
 100110
```

Exemple2 : soit à additionner $(57)_{10} + (24)_{10}$

$(57)_{10} = (111001)_2$; $(24)_{10} = (11000)_2$; $(81)_{10} = (1010001)_2$

Vérification : 57 +24 = 81

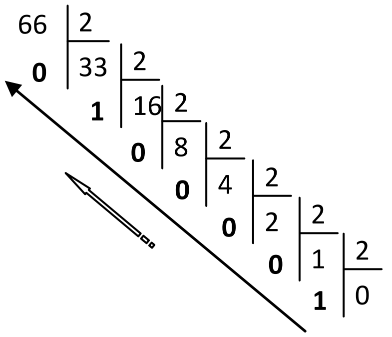
```
  111001
+  11000
--------
 1010001
```

Exemple3 : soit à additionner $(11)_{10} + (33)_{10} + (22)_{10}$

$(11)_{10} = (1011)_2$; $(33)_{10} = (100001)_2$; $(22)_{10} = (10110)_2$

```
  100001
+  10110
+   1011
--------
 1000010
```

Vérification : 33 + 22 + 11 = 66



$(66)_{10} = (1000010)_2$

2- Multiplication :

La multiplication de deux nombres binaires se fait en respectant l'algorithme suivant :

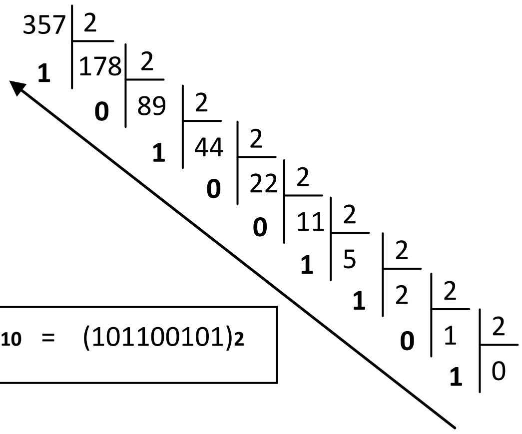
$0 \times 0 = 0$
$0 \times 1 = 0$
$1 \times 0 = 0$
$1 \times 1 = 1$

Exemple1 : soit à multiplier $(10101)_2$ par $(10001)_2$.

```
      10101
  ×   10001
  ---------
      10101
     00000
    00000
   00000
  10101
  ---------
  101100101
```

Vérification : $(10101)_2 = (21)_{10}$; $(10001)_2 = (17)_{10}$

$21 \times 17 = 357$



$(357)_{10} = (101100101)_2$

Prof : RHIMI

I- Code :

Les systèmes numériques traitent des signaux qui représentés par des symboles qui sont les éléments binaires (bits). La correspondance entre signaux et bits est définie par **un code binaire** .
Une succession des bits forme un code. Chaque code peut être composé d'un ou plusieurs bits. Suivant le nombre de bits qui le compose, on peut obtenir des combinaisons différentes de code (chaque code correspondant à une commande ou une donnée).

II- Code binaire pur :

Tout nombre décimal peut être converti en son équivalent binaire.
Quant on fait correspondre à un nombre décimal son équivalent binaire, par division successive par 2, on dit qu'on a réalisé un codage binaire pur.

III- Code Gray ou binaire réfléchi :

Ce codage permet de ne faire changer qu'un seul bit à la fois quand un nombre est augmenté d'une unité.
Pour passer de binaire pur au binaire réfléchi on procède ainsi :

- On choisit un code de départ : **0** et **1.**
- On symétries ces deux premiers lignes (comme une réflexion dans un miroir) et on ajoute 1 au début des nouveaux nombres et on ajoute 0 au début des anciens.

0	0	0	0	
0	0	0	1	
				Miroir
0	0	1	1	
0	0	1	0	
				Miroir
0	1	1	0	
0	1	1	1	
0	1	0	1	
0	1	0	0	
				Miroir
1	1	0	0	
1	1	0	1	
1	1	1	1	
1	1	1	0	
1	0	1	0	
1	0	1	1	
1	0	0	1	
1	0	0	0	

IV- Conversion entre codes binaires :

1) Conversion du binaire naturel en binaire réfléchi (Gray):

Soit **N** un nombre écrit en binaire naturel
$N = (B_{j+1}\ B_j\ B_{j-1} \ldots\ldots\ldots\ldots B_1\ B_0)_2$

- Le premier chiffre (de poids le plus fort) du naturel est le même que le chiffre du réfléchi.
- Si les bits B_{j+1} et B_j ont même valeur (0 ou 1), le chiffre correspondant en binaire réfléchi est 0.
- Si les bits B_{j+1} et B_j ont des valeurs différentes, alors le chiffre correspondant en binaire réfléchi est 1.

Exemple 1 : Convertir le nombre binaire naturel $(1011)_2$ en binaire réfléchi.

Nombre en binaire pur 1 0 1 1

Nombre en binaire réfléchi 1 1 1 0

$(1011)_2 = (1110)_{\text{réfléchi}}$

Exemple 2 : Convertir le nombre binaire naturel $(110101)_2$ en binaire réfléchi.

Nombre en binaire pur 1 1 0 1 0 1

Nombre en binaire réfléchi 1 0 1 1 1 1

$(110101)_2 = (101111)_{réfléchi}$

2) Conversion du binaire réfléchi en binaire naturel:

Soit **N** un nombre écrit en binaire réfléchi et **N*** son équivalent en binaire naturel.

$N = (G_{j+1}\ G_j\ G_{j-1}\ ....G_1\ G_0)_{réfléchi}$; $N^* = (B_{j+1}\ B_j\ B_{j-1}\ ....B_1\ B_0)_2$

- Le premier chiffre (de poids le plus fort) du réfléchi est le même que le chiffre du naturel.
- Si les bits B_{j+1} et G_j ont même valeur (0 ou 1), le bit B_j est 0.
- Si les bits B_{j+1} et G_j ont des valeurs différentes, alors le bit B_j est 1.

Exemple 1 : Convertir le nombre binaire réfléchi $(1011)_{réfléchi}$ en binaire naturel.

Nombre en binaire réfléchi 1 0 1 1

Nombre en binaire pur 1 1 0 1

$(1011)_{réfléchi} = (1101)_2$

Exemple 2 : Convertir le nombre binaire réfléchi $(110101)_{réfléchi}$ en binaire naturel.

Nombre en binaire pur 1 1 0 1 0 1

Nombre en binaire réfléchi 1 0 0 1 1 0

$(110101)_{réfléchi} = (100110)_2$

V- Code **D**écimal **C**odé **B**inaire : («binary coded decimal » ou **BCD**)

On obtient le code **BCD** par la représentation de chaque chiffre d'un nombre décimal par son équivalent binaire sur 4 bits.

Chiffre	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Les groupes utilisés dans le code **BCD** (0 à 9)

Les groupes non utilisés dans le code **BCD** (10 à 15)

(car le plus élevé des chiffres décimaux est 9)

Exemple1 : convertir le nombre décimal 385 en BCD

décimal	3	8	5
BCD	11	1000	0101

On obtient $(385)_{10} = (1110000101)_{BCD}$

Exemple2 : convertir le nombre décimal 850 en BCD

décimal	8	5	0
BCD	1000	0101	0000

On obtient $(850)_{10} = (100001010000)_{BCD}$

Exemple3 : convertir le nombre BCD 10011100000101 en décimal

BCD	10	0111	0000	0101
décimal	2	7	0	9

Rq Il faut fractionner en groupe de 4 bits le nombre BCD à partir de l'adroite.

On obtient $(10011100000101)_{BCD} = (2709)_{10}$

Exemple3 : convertir le nombre BCD 10111010001101 en décimal

BCD	10	1110	1000	1001
décimal	2	14	8	9

$- - - - \rightarrow \quad 14 \;\; n'est\ pas\ décimal$

1110 est un groupe du code inadmissible indiquant une erreur dans le nombre BCD donc le nombre BCD $(10111010001101)_{BCD}$ n'est pas un code BCD (il ne représente pas un nombre décimal).

ACTIVITES PRATIQUES (MA Page ......)

Chapitre II	SYSTEME DE NUMERATION ET CODES	
Leçon 3	Codes alphanumériques	Le ...../...../201..

Prof : RHIMI

I – Code ASCII : (**A**méricain **S**tandard **C**ode for **I**nformation **I**nterchange).
Un ordinateur doit être capable de traiter une information non numérique. C'est-à-dire il doit reconnaitre des codes qui correspondent à des nombres, des lettres, ders signe de ponctuation et des caractères spéciaux : Les codes de ce genre sont dit ***alphanumériques.***
Le code alphanumérique reproduit tous les caractères et les diverses fonctions que l'on trouve sur un clavier d'ordinateur : c'est un code utilisé pour communiquer entre le clavier d'un ordinateur et l'unité centrale.

Tableau du code ASCII
Le code ASCII standard est un code à 7 éléments, on peut donc représenter $2^7 = 128$ groupe de code.

ASCII à 7 éléments				**B6**	0	0	0	0	1	1	1	1
				B5	0	0	1	1	0	0	1	1
				B4	0	1	0	1	0	1	0	1
B3	**B2**	**B1**	**B0**									
0	0	0	0		NUL	DLE	SP	0	@	P		p
0	0	0	1		SOH	DC1	!	1	A	Q	a	q
0	0	1	0		STX	DC2	"	2	B	R	b	r
0	0	1	1		ETX	DC3	#	3	C	S	c	s
0	1	0	0		ETO	DC4	$	4	D	T	d	t
0	1	0	1		ENQ	NAK	%	5	E	U	e	u
0	1	1	0		ACK	SYN	&	6	F	V	f	v
0	1	1	1		BEL	ETB	‘	7	G	W	g	w
1	0	0	0		BS	CAN	(	8	H	X	h	x
1	0	0	1		HT	EM	)	9	I	Y	i	y
1	0	1	0		LF	SUB	*	:	J	Z	j	z
1	0	1	1		VT	ESC	+	;	K	[	k	{
1	1	0	0		FF	ES	,	<	L	\	l		
1	1	0	1		CR	GS	-	=	M	]	m	}
1	1	1	0		SO	RS	.	>	N	^	n	~
1	1	1	1		SI	US	/	?	O	_	o	DEL

II- Code-barres. (voir MC page 27)

Exercice 1 Convertir en binaire naturel les nombres décimaux suivants :

0 ; 13 ; 325 ; 512 ; 364 ; 127

Exercice 2 Convertir en hexadécimal les nombres décimaux suivants :

0 ; 16 ; 179 ; 171 ; 43981

Exercice 3 Convertir en décimal les nombres binaires suivants :

1 ; 101011 ; 1000110 ; 111111 ; 11011011 ; 1000001000000001

Exercice 4 Convertir en décimal les nombres hexadécimaux suivants :

63 ; 2FB6 ; 5A2B9C ; ABCDEF

Exercice 5 Convertir en binaire réfléchi les nombres binaires naturels suivants :

1 ; 101101 ; 1001110 ; 111111 ; 11011011 ; 1001001001001

Exercice 6 Convertir en binaire naturel les nombres binaires réfléchis suivants :

1 ; 101101 ; 1001110 ; 111111 ; 11011011 ; 1001001001001

Exercice 7 Convertir en BCD les nombres décimaux suivants :

235 ; 111 ; 6429 ; 5006 ; 99999 ; 102030405

Exercice 8 Convertir en décimal les nombres BCD suivants :

10011011010010110 ; 1011001100000110111 ; 10010101000010000011

Exercice 9 Compléter le tableau on convertissant les nombres suivants :

Code décimal	Code binaire pur	Code octal	Code hexadécimal	Code BCD	Code ASCII
33	....	....	....	....	....
....	01011	....	....	....	....
127	....	....	....	....	....
....	....	....	58	....	....
....	....	....	....	....	W
....	....	....	....	100100010	....
....	....	....	....	....	/
....	....	144	....	....	....