

Représentation des entiers

Table des matières

1	Représentation binaire de l'information	1
1.1	Les bases de calcul	1
1.2	Pourquoi le bit est-il l'unité de base du codage des informations dans les systèmes informatiques ? .	1
1.3	Unité élémentaire	1
2	Principe de la numération binaire et hexadécimal	2
2.1	Principe de la représentation d'un nombre en binaire	2
2.2	Conversion décimal vers binaire	2
2.3	Vers le décimal à partir des autres bases	3
3	Application : le codage de la couleur	4
4	Codage binaire d'un texte	5
4.1	Problèmes	5
4.2	Le code ASCII	5
4.3	Les codes ISO 8859-1 et ANSI utilisé par WINDOWS	6
4.4	Codage Unicode	7

1) Représentation binaire de l'information

1.1) Les bases de calcul

Base	Système	Utilisation
2	Binaire	
8	Octal	
10	Décimal	
12	Duodécimal	
16	Hexadécimal	
20	Vicésimal	
60	Sexagécimal	

Tableau 1 – Les différentes bases de numération

1.2) Pourquoi le bit est-il l'unité de base du codage des informations dans les systèmes informatiques ?

Il est simple de représenter deux états physiques grâce à :

- la lumière (exemple : la fibre) ;
- la tension (exemple : le processeur en 0 ou 5V ou en 3,5V) ;
- le champ magnétique (exemple : le disque dur).

1.3) Unité élémentaire

L'unité élémentaire utilisée en informatique pour coder l'information est appelée bit, contraction de

Application au codage d'une couleur : Supposons que l'on souhaite coder des couleurs en binaire.

- Combien de couleurs peut-on coder sur un bit ?
- Si l'on souhaite coder davantage de couleurs, il va falloir utiliser une séquence de plusieurs bits appelée mot binaire (ou nombre binaire). Un mot de 8 bits est appelé unou encore un
- Combien de couleurs peuvent coder des mots de 2 bits, 3 bits ... n bits.
.....
.....
.....
.....

2) Principe de la numération binaire et hexadécimal

2.1) Principe de la représentation d'un nombre en binaire

Le texte de Leibniz (en Figure 1) datant du début du XVIII siècle nous explique le principe de la numération en binaire (base 2).

Le calcul ordinaire d'arithmétique se fait suivant la progression de dix en dix. On se sert de dix caractères, qui sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, qui signifient zéro, un et les nombres suivants jusqu'à neuf inclusivement. Et puis allant à dix, on recommence, et on écrit dix par 10, et dix fois dix ou cent par 100, et dix fois cent ou Mille par 1 000, et dix fois mille par 10 000, etc.
Mais au lieu de la progression de dix en dix j'ai employé depuis plusieurs années la progression la plus simple de toutes, qui va de deux en deux, ayant trouvé qu'elle sert à la perfection de la science des nombres. Ainsi je n'y emploie point d'autres caractères que 0 et 1, et puis allant à deux, je recommence. C'est pourquoi deux s'écrit ici par 10, et deux fois deux ou quatre par 100, et deux fois quatre ou huit par 1 000, et deux fois huit ou seize par 10 000, et ainsi de suite. Voici la table des nombres de cette façon, qu'on peut continuer tant que l'on voudra.

FIGURE 1 – Texte de Leibnitz

A partir de ce texte de Leibnitz, compléter le tableau 2 (ne pas tenir compte de la dernière colonne pour l'instant).

Décimal	Binaire (5bits)	Hexadécimal
0		
1		
2		
3		
4		

5
6
7
8
9
10
11
12
13
14
15

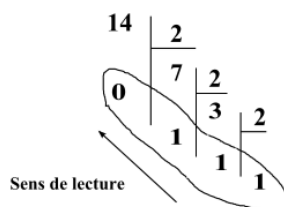
Tableau 2 – Représentation en binaire et en hexadécimal

2.2) Conversion décimal vers binaire

Pour effectuer cette conversion, on effectue une suite de divisions par 2. Il suffit ensuite de lister les restes des diverses divisions en commençant par la dernière effectuée.

Dividende	Diviseur
Reste	Quotient

Exemple : $(14)_{10} = (1110)_2$ car



Exercice :

- Convertir $(4)_{10}$ en base 2 ;

.....

.....

.....

- Convertir $(35)_{10}$ en base 2 ;

.....

.....

.....

- Convertir $(255)_{10}$ en base 2 ;

.....

.....

.....

- Convertir $(255)_{10}$ en base 16 ;

.....

2.3) Vers le décimal à partir des autres bases

Pour convertir un nombre N d'une base b dans la base 10, on décompose ce nombre dans l'ordre des puissances décroissantes de la base. Le nombre N s'écrit de façon unique sous la forme :

$$N = a_n \times b^n + \dots + a_2 \times b_2 + a_1 \times b_1 + a_0 \times b_0 \text{ avec } \begin{cases} n : \text{un entier naturel;} \\ b : \text{la base de numération;} \\ a_i : \text{les chiffres associés à la base tels que } 0 \leq a_i < b. \end{cases}$$

Exemple : $(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = (13)_{10}$

Exercice :

- Convertir $(10110)_2$ en base 10 :
- Convertir $(1111)_2$ en base 10 :
- Convertir $(2A1F)_{16}$ en base 10 :

Remarque : afin d'effectuer des conversions binaire vers décimal ou décimal vers binaire, il est également possible d'utiliser le Tableau 3 :

1 octet								
$2^8 = 256$	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

Tableau 3 – Conversion binaire vers décimal

Exercice : code en utilisant le tableau $(01010100)_2$ en décimal

3) Application : le codage de la couleur

Notre objectif est d'étudier le codage de la couleur à l'aide d'un logiciel de retouche d'images : "Gimp". Il est téléchargeable à l'adresse : <https://www.gimp.org/downloads/>.

Exercice : le logiciel "GIMP" est un logiciel libre.

- Qu'est-ce que cela signifie ? Pouvez-vous le télécharger librement ? Pouvez-vous le modifier librement ?
- Un freeware (encore appelé graticiel ou gratuiciel) est-il un logiciel libre ? Sinon, quelle est la différence ?
- Qu'est-ce qu'un shareware (encore appelé partagiciel ou contribuciel) ?

A faire en classe :

- Lancer le logiciel et charger une image depuis le menu fichier *Ouvrir* ;
- Si la boîte à outils n'est pas ouverte, l'ouvrir depuis le menu *Outils* puis *Boîte à outils*. Sélectionner l'outil pipette puis cliquer sur un pixel de l'image.
- Cliquer ensuite sur la couleur de premier plan sélectionnée afin d'afficher les informations relatives à la couleur du pixel.

Remarque : il est possible également d'obtenir les informations sur la couleur d'un pixel à l'aide de l'outil pipette en cliquant sur un pixel de l'image tout en maintenant la touche SHIFT enfoncée.

Il est possible de modifier manuellement la valeur des composantes RVB ainsi que les valeurs du codage hexadécimal de la couleur.

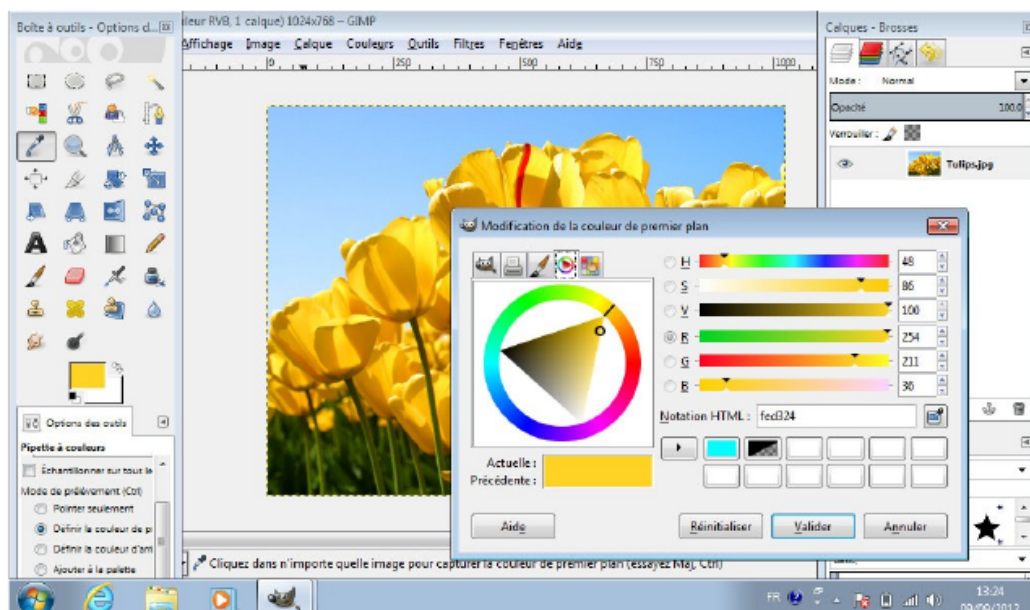


FIGURE 2 – GIMP

- Qu'appelle-t-on "composantes RVB" ?
- Quelles sont les valeurs minimales et maximales qui peuvent être attribuées à chaque composante RVB ? Sur combien d'octets chaque composante RVB est-elle codée ?
- Sur combien d'octets est codée chaque couleur de pixel ?
- Combien de couleurs le logiciel peut-il gérer ?
- Prévoir les couleurs dont les composantes RVB sont : $(FF - FF - FF)_{16}$ et $(255 - 255 - 0)_{10}$
- Donner le codage RVB en binaire de la couleur cyan.

4) Codage binaire d'un texte

4.1) Problèmes

En recherchant un extrait du « seigneur des anneaux », nous avons trouvé la page web représentée sur la Figure 3. Pourquoi une partie du texte n'est pas compréhensible ? Quel type de lettres pose problème ?

4.2) Le code ASCII

Le code ASCII (American Standard Code for Information Interchange), défini aux Etats-Unis en 1963 est basé sur un tableau contenant les caractères les plus utilisés en langue anglaise. Dans cette table, il existe 127 caractères.

Exemples :

- Le caractère "A" est codé en ASCII par 0x41 en hexadécimal (65 en décimal ou 10000012 en binaire).
- Le caractère "1" est codé en ASCII par 0x31 en hexadécimal (49 en décimal ou 01100012 en binaire).
- Les codes compris entre 000 et 1*F ne représentent pas des caractères, ils ne sont pas affichables. Ces codes, souvent nommés caractères de contrôles sont utilisés pour indiquer des actions comme passer à la ligne (Carriage Return CR 0D, Line Feed LF 0A), émettre un BIP sonore (BELL), etc...

Exercice

1. Sur combien de bits est-il possible de coder la totalité des codes ASCII ?

« Trois Anneaux pour les Rois Elfes sous le ciel,
 Sept pour les Seigneurs Nains dans leurs demeures de pierre,
 Neuf pour les Hommes Mortels destinés au trépas,
 Un pour le Seigneur des Ténébres sur son sombre trône
 Dans le Pays de Mordor où s'étendent les Ombres.
 Un Anneau pour les gouverner tous, Un Anneau pour les
 trouver,
 Un Anneau pour les amener tous et dans les ténébres les
 lier
 Au Pays de Mordor où s'étendent les Ombres. »

« La Compagnie de l'Anneau sera de Neuf ; et les Neufs
 Marcheurs seront opposés aux Neufs Cavaliers qui sont
 mauvais. Gandalf ira avec vous et votre fidèle serviteur ; car
 ceci sera sa grande tâche et peut-être la fin de ses labeurs.
 Pour le reste, ils représenteront les autres Gens Libres du
 Monde : Elfes, Nains et Hommes. Legolas représentera les
 Elfes, et Gimli, fils de Glóin les Nains. [...] Pour les
 Hommes, vous aurez Aragorn fils d'Arathorn ainsi que
 Boromir du Gondor »

FIGURE 3 – Extrait du Seigneur des Anneaux

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. klock	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 4 – Table ASCII

- Les ordinateurs travaillant avec des cases mémoires de 1 octet, quelle pouvaient être l'utilité du 8ième bit à une époque où les erreurs de mémoire étaient fréquentes ?
- Coder, à l'aide de la table ASCII, la phrase suivante : « la compagnie de l'anneau ».
- Retrouver le texte correspondant au code ASCII suivant : (46 72 6F 6E 64 6F 20 6C 65 20 47 6F 62 62 69 74)16.

5. Justifier, pourquoi il n'est pas possible de coder correctement le texte « Mon précieux » à l'aide du code ASCII. Comment peut-on palier à cet inconvénient.

4.3) Les codes ISO 8859-1 et ANSI utilisé par WINDOWS

La nécessité de représenter des caractères non présents dans la table ASCII tels que ceux de l'alphabet latin comme le « à », le « é », le « ç »... impose l'utilisation d'un autre code. Ces codes sont des extensions du code ASCII. Pour cela, le 8ème bit est utilisé. On parle de code ASCII étendu. L'ISO, organisation internationale de normalisation, propose plusieurs variantes de ce code, adaptées aux différentes langues. Nous utilisons le norme ISO-8859-1 nommée aussi ISO- latin-1.

ISO-8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HFI	VTS	PLD	PLU	RI	SS2	SS3
9x	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
Ax	NBSP	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ø	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Windows-1252 (CP1252)																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	€		‚	ƒ	„	…	†	‡	ˆ	%	Š	€	œ		Ž	
9x		‘	’	“	”	•	—	™	š	›	œ		ž	ÿ		
Ax	NBSP	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ø	ñ	ò	ó	ô	õ	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

FIGURE 5 – Codes ISO 8859-1 et ANSI

Microsoft propose le codage Windows-1252 appelé aussi code ANSI (American National Standard Institute). Ce code ne diffère de l'ISO-8859-1 que pour quelques caractères tels que le signe €, la ligature o-e ou certains guillemets. Il propose des extensions différents selon le code page retenu. Ce code est choisi lors de la configuration du système et permet de définir le jeu de caractères qui va être employé avec la machine.

Exercice :

1. Quel est le code hexadécimal, binaire et décimal de la lettre A dans chacune des 2 normes ?
2. En ANSI, coder 70 € en hexadécimal.

4.4) Codage Unicode

La généralisation de l'utilisation d'internet dans le monde a ainsi nécessité une prise en compte d'un nombre beaucoup plus important de caractère. Pour cela, en 1991 une nouvelle norme a vu le jour : Unicode. Unicode a pour ambition de rassembler tous les caractères existant afin qu'une personne utilisant Unicode puisse, sans changer la configuration de son traitement de texte, à la fois lire des textes en français ou en japonais. Unicode est uniquement une table qui regroupe tous les caractères existant au monde, il ne s'occupe pas de la façon dont les caractères sont codés dans la machine. Unicode accepte plusieurs systèmes de codage : UTF-8, UTF-16, UTF-32. Le plus utilisé, notamment sur le Web, est UTF-8. Pour encoder les caractères Unicode, UTF-8 utilise un nombre variable d'octets : les caractères "classiques" (les plus couramment utilisés) sont codés sur un octet, alors que

des caractères "moins classiques" sont codés sur un nombre d'octets plus important (jusqu'à 4 octets). Un des avantages d'UTF-8 c'est qu'il est totalement compatible avec la norme ASCII : les caractères Unicode codés avec UTF-8 ont exactement le même code que les mêmes caractères en ASCII.

Exercice : Quel est le code binaire du "b" minuscule Unicode codé avec UTF-8 ?