



# ISN – Informatique et Sciences du Numérique

## TP : CODAGE DE L'INFORMATION

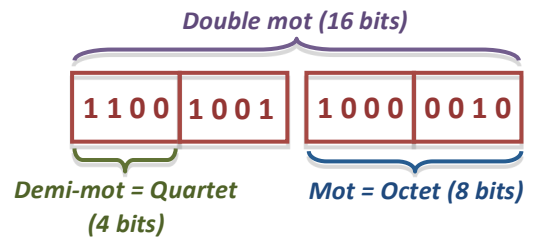


<b>1 – CODAGE D'UN NOMBRE .....</b>	<b>2</b>
1.1 – L'INFORMATION .....	2
1.2 – LE SYSTEME BINAIRE .....	2
1.2.1 – Conversion binaire vers décimal .....	2
1.2.2 – Conversion décimal vers binaire .....	3
1.3 – LE SYSTEME HEXADECIMAL.....	3
1.3.1 – Conversion hexadécimal vers décimal .....	4
1.3.2 – Conversion décimal vers hexadécimal .....	4
1.3.3 – Conversion binaire vers hexadécimal .....	4
1.3.4 – Conversion hexadécimal vers binaire .....	5
1.5 – NOMBRES RELATIFS (ENTIERS SIGNES) .....	5
1.6 – NOMBRES A VIRGULE FLOTTANTE (NOMBRES REELS).....	6
<b>2 – CODAGE BINAIRE D'UN TEXTE .....</b>	<b>8</b>
2.1 – PROBLEMATIQUE .....	8
2.2 – LE CODE ASCII.....	8
2.3 – LES CODES ISO 8859–1 ET ANSI UTILISE PAR WINDOWS .....	9
2.4 – CODAGE UNICODE .....	11

# 1 – CODAGE D'UN NOMBRE

## 1.1 – L'INFORMATION

Un système numérique doit, avant de pouvoir manipuler les nombres, les textes, les images ou les sons, de les **représenter comme des suites de 0 et de 1**. La valeur 0 ou 1 est appelée **booléen**, **chiffres binaires** ou encore **bit (binary digit)**. Une **suite de chiffres binaires** est appelée **mot binaire**. Suivant le processeur, la taille du mot sera différente. Les mots les plus courants ont une taille de **8, 16, 32** ou **64 bits**. On rencontre aussi les notions de **demi-mots** ou de **double-mot**.

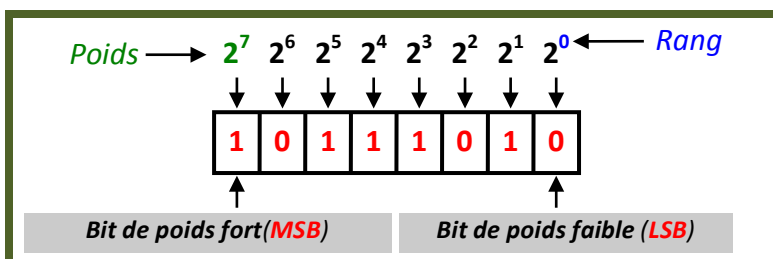


Un nombre peut être de différents types : nombre entier positif ou nul (0 ; 1 ; 245 ; ...); nombre entier négatif (- 1 ; - 324 ; ...); nombre fractionnaire (3,1415 ; - 0,6 ; ...); Nombre en notation scientifique ( $2,7 \cdot 10^4$  ;  $10^{21}$  ; ...).

Pour représenter un nombre, il est possible d'utiliser différentes bases parmi lesquelles on peut citer : la **base 10 (décimale)** qui est la base universelle ; la **base 2 (binaire)** qui est la base utilisée en informatique et dans l'**algèbre de Boole** ; la **base 16 (base hexadécimale)** qui est une base utilisée en informatique.

## 1.2 – LE SYSTEME BINAIRE

Dans un système binaire ou **base 2**, il n'y a que **deux chiffres possibles** : **0** et **1**.



Le système de numération binaire est indiqué par l'**indice 2** ou par les **symboles %** ou **B**.

$$(1000\ 1011)_2 = \%10001011 = B'1000\ 1011'$$

### 1.2.1 – Conversion binaire vers décimal

Pour trouver l'équivalent décimal d'un nombre binaire, il suffit de faire la **somme des produits de chaque bit par le poids de son rang**.

**Exemple**

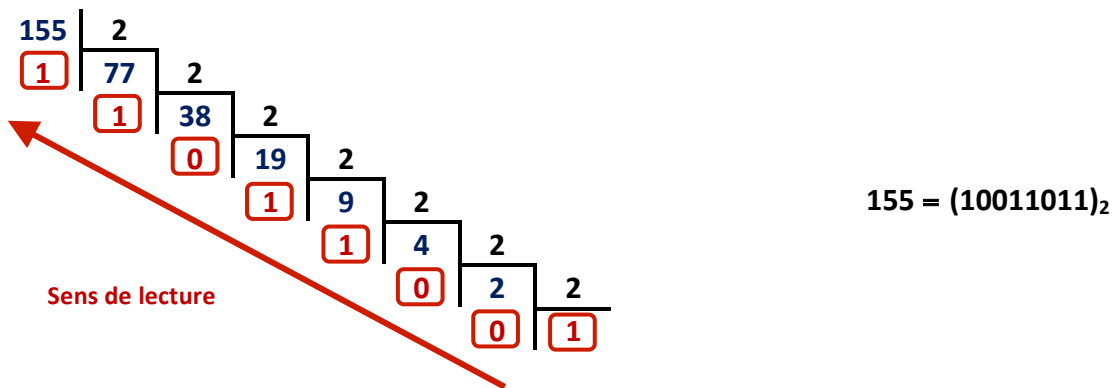
$$(10111010)_2 = (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = 186$$

**Exercice n°1**

**Convertir** les nombres binaires suivants vers leur équivalent décimal :  $(1110)_2$ ,  $(10111001)_2$ .

**1.2.2 – Conversion décimal vers binaire**

Cette conversion peut être réalisée par la **méthode des divisions successives par 2**.

**Exemple****Exercice n°2**

**Convertir** les nombres décimaux suivants vers leur équivalent binaire : **219, 186**.

**1.3 – LE SYSTEME HEXADECIMAL**

Le langage binaire est difficilement manipulable par l'homme pour de grandes séries binaires. On utilise alors le **système hexadécimal (base 16)**. Celui-ci comporte **16 symboles (base 16)** : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, et F**. Pour indiquer la base 16, on peut utiliser les indices **16** mais en programmation on place le symbole **\$ (dollar)** ou les symboles **0x** devant le nombre :  $(AA)_{16} = \$AA = 0xAA$ .



### 1.3.1 – Conversion hexadécimal vers décimal

Pour trouver l'équivalent décimal d'un nombre hexadécimal, il suffit de faire la somme des produits de chaque symbole hexadécimal par le poids de son rang.

#### Exemple

$$\begin{array}{cccc}
 (1 & D & E & 8)_{16} \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 1 & 13 & 14 & 8
 \end{array}$$

$$\begin{aligned}
 (1DE8)_{16} &= 1 \times 16^3 + 13 \times 16^2 + 14 \times 16^1 + 8 \times 16^0 \\
 (1DE8)_{16} &= 1 \times 4096 + 13 \times 256 + 14 \times 16 + 8 \times 1 \\
 (1DE8)_{16} &= 7656
 \end{aligned}$$

#### Exercice n°3

Convertir les nombres hexadécimaux suivants vers leur équivalent décimal :  $(5D)_{16}$ ,  $(F3C)_{16}$ .

### 1.3.2 – Conversion décimal vers hexadécimal

Cette conversion peut être réalisée par la méthode des divisions successives par 16.

#### Exemple

$$\begin{array}{r|l}
 978 & 16 \\
 \hline
 2 & 61 \\
 & \hline
 & 13 & 16 \\
 & \hline
 & & 3
 \end{array}$$

Sens de lecture

$$978 = (3D2)_{16}$$

#### Exercice n°4

Convertir les nombres décimaux suivants vers leur équivalent hexadécimal : 184, 252.

### 1.3.3 – Conversion binaire vers hexadécimal

#### Exemple

$$\begin{array}{ccc}
 ( & 0110 & 1111 & 0011 & )_2 & \text{Regroupement en quartet} \\
 \downarrow & \downarrow & \downarrow & & & \text{Conversion binaire hexadécimal} \\
 6 & 15 & 3 & & & \\
 \downarrow & \downarrow & \downarrow & & & \text{Conversion hexadécimal décimal} \\
 ( & 6 & F & 3 & )_{16}
 \end{array}$$

$$(0110 \ 1111)_2 = (6F)_{16}$$

#### Exercice n°5

Convertir les nombres hexadécimaux suivants vers leur équivalent binaire :  $(F3)_{16}$ ,  $(45A)_{16}$ .



### 1.3.4 – Conversion hexadécimal vers binaire

#### Exemple

(	D	7	E	) <sub>16</sub>	
	↓	↓	↓		<i>Conversion hexadécimal décimal</i>
	13	7	14		
	↓	↓	↓		<i>Conversion décimal binaire</i>
(	1101	0111	1110	) <sub>2</sub>	$(D7E)_{16} = (1101\ 0111\ 1110)_2$

#### Exercice n°6

**Convertir** les nombres hexadécimaux suivants vers leur équivalent binaire :  $(1001\ 1011)_2$ ,  $(1000\ 1001)_2$ .

## 1.5 – NOMBRES RELATIFS (ENTIERS SIGNES)

La représentation d'un **nombre binaire signé** est régie par les règles suivantes (pour un nombre stocké sur un **octet**) :

$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$
<b>Signe</b>	<b>Représentation du nombre</b>						

Si le bit de signe a la **valeur 0**, le nombre est **positif** : La valeur du nombre est codée en **binaire naturel**.

Si le bit de signe a la **valeur 1**, le nombre est **négatif** : La valeur du nombre est codée selon le **complément à 2**. Pour obtenir le complément à 2 d'un mot binaire, il suffit **d'ajouter 1 au complément du nombre**.

#### Exemple

Le nombre binaire signé correspondant à la valeur **78** :

$$78 = \overbrace{0}^{\text{Signe}} \overbrace{1\ 0\ 0\ 1\ 1\ 1\ 0}^{\text{Valeur en binaire naturel}}$$

Le nombre binaire signé correspondant à la valeur **-23** :

$$-23 = \overbrace{1}^{\text{Signe}} \overbrace{1\ 1\ 0\ 1\ 0\ 0\ 1}^{\text{Valeur en complément à 2}}$$

23	→	0	0	1	0	1	1	1
Complément	→	1	1	0	1	0	0	0
	+							1
Complément à 2	→	1	1	0	1	0	0	1



## Exercice n°7

1. **Convertir** les nombres suivants vers leur équivalent binaire (signé) : **36, –115**.
2. **Convertir** les nombres binaires signés suivants vers leur équivalent décimal : **(1100 1011)<sub>2</sub>, (0110 1001)<sub>2</sub>**.

## 1.6 – NOMBRES A VIRGULE FLOTTANTE (NOMBRES REELS)

Pour représenter les nombres à virgule, on utilise une représentation similaire à la «notation scientifique» des calculatrices, sauf qu'elle est en base deux et non en base dix. Il s'agit de la représentation en **virgule flottante**. Un tel nombre est représenté sous la forme suivante :

$$S m \times 2^n$$

**S** : **Signe** du nombre

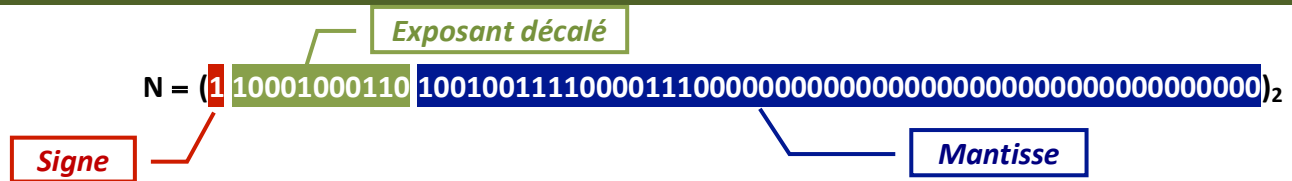
**m** : **Mantisse** du nombre comprise ( $1 \leq m < 2$ )

**n** : **Exposant** du nombre (entier relatif)

Les nombres à virgule flottante doivent respecter une forme normalisée, afin que la représentation ne varie pas d'un matériel à l'autre. La norme standard est la norme **IEEE 754**. Elle définit **4 formats** pour représenter des nombres à virgule flottante. Le plus courant est le format double précision :

b <sub>63</sub>	b <sub>62</sub>	b <sub>61</sub>	...	b <sub>53</sub>	b <sub>52</sub>	b <sub>51</sub>	b <sub>50</sub>	b <sub>49</sub>	...	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
<b>Signe</b>		<b>Exposant décalé</b>					<b>Mantisse</b>					

Élément	Nb de bits	Description
<b>Signe</b>	<b>1</b>	<b>0 = + ; 1 = –</b>
<b>Exposant décalé</b>	<b>11</b>	L'exposant <i>n</i> est un entier relatif compris entre <b>–1 022 et 1 023</b> . Hors il doit être représenté comme un entier naturel. Pour cela on utilise un <b>décalage de 1023</b> . L' <b>exposant décalé</b> possède donc une valeur <b>comprise entre 1 et 2046</b> . Les deux entiers <b>0 et 2 047</b> sont réservés pour certaines situations : <b>NaN</b> = Not a Number ; <b>+∞, –∞</b> .
<b>Mantisse</b>	<b>52</b>	Il s'agit d'un <b>nombre binaire à virgule compris entre 1 inclus et 2 exclu</b> comprenant <b>52 chiffres après la virgule</b> . Il y a toujours seul chiffre avant la virgule et ce chiffre est toujours un 1 ; il est donc inutile de le représenter. Les <b>52 bits</b> pour <b>représenter les 52 chiffres après la virgule</b> .

**Exemple****Signe = 1 : Nombre négatif.****Exposant décalé = 10001000110 = 1094. Exposant = 1094 – 1023 = 71****Mantisse = 1.10010011110000111000000000000000000000000000000000**  
 $= 2^0 + 2^{-1} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-15} + 2^{-16} + 2^{-17}$   
 $= (2^{17} + 2^{16} + 2^{13} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^2 + 2^1 + 1) / 2^{17} = 206\,727 / 131\,072 = 1,57720$ **Nombre = –1,57720 · 2<sup>71</sup> = –3.72406 × 10<sup>21</sup>****Exercice n°8**

1. **Trouver** le nombre à virgule représenté par le mot : **(41 CA 0000)**
2. **Donner** le code binaire et hexadécimal du nombre **10,50**.

**Exercice n°9 : Programmes de conversion**

1. **Etablir** un script qui permettent de la conversion décimale d'un nombre binaire entré par l'utilisateur et la conversion binaire d'un nombre décimal entré par l'utilisateur.
2. **Etablir** un script qui permettent de réaliser toutes les conversions.



## 2 – CODAGE BINAIRE D'UN TEXTE

### 2.1 – PROBLEMATIQUE

En recherchant un extrait du « Seigneur des anneaux », nous sommes tombés sur la page WEB ci-contre.

« 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 Ténèbres 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 Ténèbres 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 »

#### Exercice n°10

1. Indiquer pourquoi une partie du texte n'est pas compréhensible.
2. Préciser quel type de lettres pose problème.

### 2.2 – 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**. Chaque caractère est représenté sur **7 bits** (ce qui donne  $2^7 = 128$  combinaisons possibles).

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BELL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DEL	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





## Exemples

Le caractère « **A** » est codé en ASCII par **0x41** en hexadécimal (**65** en décimal ou **1000001<sub>2</sub>** en binaire).  
Le caractère « **1** » est codé en ASCII par **0x31** en hexadécimal (**49** en décimal ou **0110001<sub>2</sub>** en binaire).

Les codes compris entre **0x00** et **1F** 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 (**CR**, **LF**), émettre un bip sonore (**BEL**), etc...

## Exercice n°11

- Coder**, à l'aide de la table ci-dessus, la phrase suivante: "**La compagnie de l'anneau**".
- Retrouver** le texte correspondant au code ASCII suivant : **(46 72 6F 6E 64 6F 6E 20 6C 65 20 48 6F 62 62 69 74)<sub>16</sub>**.
- Justifier**, pourquoi il n'est pas possible de coder correctement le texte "Mon précieux" à l'aide du code ASCII.

## 2.3 – 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 », « ç »... impose l'utilisation d'un autre code.

Ces codes sont des extensions du code ASCII. Pour cela le **8<sup>ème</sup> bit** est utilisé ce qui permet de coder **256 caractères** (128 caractères supplémentaire par rapport au code ASCII de base). On parle de code **ASCII étendu**. L'ISO, organisation internationale de normalisation, propose plusieurs variantes de ce code, adaptée aux différentes langues. Nous utilisons la norme **ISO-8859-1** nommée aussi **ISO-Latin1**.

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

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 différentes 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 n°12

1. **Rechercher** à partir de ses propriétés la taille en octet du fichier **texte1.txt**. **Justifier** la taille de ce fichier à partir du texte visualisé avec le **Bloc-Note** de Windows.
2. **Lancer** le logiciel **Free Hex editor** **Ouvrir** le fichier **texte1.txt**.
3. **Justifier** la valeur des octets n° **0x3f** et **0x40**.
4. **Donner** le codage binaire du nom « Bilbon ».
5. **Remplacer**, sous **Free Hex editor**, les octets numéro **0x26** à **0x2B** par le codage binaire précédent. **Enregistrer** les modifications. **Vérifier** à partir du **Bloc-Note** que les modifications apportées sont satisfaisantes.
6. **Ouvrir** le fichier **texte2.doc** avec **Word**. **Noter** bien le texte de ce fichier.
7. **Rechercher** à partir de ses propriétés la taille en octet du fichier **texte2.doc**. **Indiquer** pourquoi la taille de ce fichier **texte2.doc** est différente de celle du fichier **texte1.txt**.
8. **Lancer** le logiciel **Free Hex editor**. **Ouvrir** le fichier **texte2.doc**.
9. **Préciser** s'il est possible de « déchiffrer » facilement le contenu comme pour le fichier **texte1.txt**.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	4D	65	73	20	63	68	65	72	73	20	65	74	20	74	72	E8
00000010	73	20	61	69	6D	E9	77	20	48	6F	62	62	69	74	73	20
00000020	21	20	64	69	74	20	46	72	6F	64	6F	6E	20	2C	20	70
00000030	72	6F	66	6F	6E	64	E9	6D	65	6E	74	20	E9	6D	75	2E
00000040	0D	0A	43	65	63	69	20	6E	27	65	73	74	20	70	61	73
00000050	20	73	6E	65	20	63	68	61	73	73	65	20	61	75	20	74
00000060	72	E9	73	6F	72	20	6E	69	20	75	6E	20	76	6F	79	61
00000070	67	65	20	61	6C	6C	65	72	20	65	74	20	72	65	74	6F
00000080	75	72	2E													

Octets n° 0x26  
à 0x2B

Octets n° 0x40  
et 0x41

Le codage binaire du texte se trouve à partir de l'octet n°0x0A00.

10. **Remplacer**, sous **Free Hex editor**, les octets numéro **0xA26** à **0xA2B** par le codage binaire trouvé à la question 9. **Enregistrer** les modifications. **Vérifier** à partir de **Word** que les modifications apportées sont satisfaisantes.
11. **Remplacer**, sous **Free Hex editor**, la valeur de l'octet numéro **0x0000** par une valeur quelconque. **Ouvrir** à nouveau le fichier **texte2.doc** avec **Word** et **indiquer** ce qui se passe. **Préciser** à quelle information doit correspondre la valeur de l'octet numéro **0x0000**.
12. **Remplacer**, sous **Free Hex editor**, la valeur de l'octet numéro **0x0040** par une valeur quelconque. **Ouvrir** à nouveau le fichier **texte2.doc** avec **Word** et **indiquer** ce qui se passe.

## 2.4 – 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 caractères. La **norme Unicode** permet le codage de texte écrit en donnant à tout caractère de n'importe quel système d'écriture un **charset, constitué d'un nom d'un identifiant numérique**, et ce de manière unifiée, quelle que soit la plate-forme informatique ou le logiciel. Le répertoire Unicode peut contenir **plus d'un million de caractères**. La norme Unicode définit des **méthodes standardisées pour coder et stocker cet index sous forme de séquence d'octets** : **UTF-8, UTF-16, UTF-32** et leurs différentes variantes. L'UTF-8 est l'encodage (Encoding) le plus répandu. Les navigateurs Internet utilisent le codage UTF-8 par défaut. L'index des **caractères UNICODE est disponible à l'adresse suivante** : <http://www.unicode.org/fr/charts/charindex.html>.

L'encodage **UTF-8** utilise **1, 2, 3 ou 4 octets** en respectant certaines règles :

- Un caractère déjà codé en ASCII de base est **codé de manière identique en UTF-8**.

Caractère	Valeur Unicode		Code UTF-8	
	Décimale	Binaire	Binaire	Hexadécimal
A	65	01000001	01000001	41

- Les bits de poids fort du premier octet **forment une suite de 1 indiquant le nombre d'octets utilisés pour coder le caractère suivi d'un 0**. Les octets suivants commencent tous par le **bloc binaire 10**.

Caractère	Valeur Unicode		Code UTF-8	
	Décimale	Binaire	Binaire	Hexadécimal
é	233	00011 101001	11000011 10101001	C3 A9
€	8364	0010 000010 101100	11100010 10000010 10101100	E2 82 AC

### Exercice n°13

1. **Déterminer** le code UTF-8 du caractère ≤ sachant qu'il est codé sur 3 octets
2. **Lancer** le logiciel **Free Hex editor**. **Créer** un nouveau fichier. **Insérer** le code UTF-8 du caractère ≤. **Enregistrer** sous le nom **inf\_egal.txt**. **Vérifier** à partir du **Bloc-Note** que le caractère affiché correspond au caractère attendu.
3. **Lancer**, le navigateur WEB **Firefox**. **Aller** sur la page d'accueil du site du lycée Gustave Eiffel <http://www.eiffel-bordeaux.org/>.
4. **Cliquer** à partir du bouton droit de la souris sur « **Code source de la page** ».
5. **Rechercher**, dans l'entête (balise <head>) du code html, la ligne qui permet de configurer le codage utilisé par le site WEB du lycée.
6. **Modifier**, l'encodage du texte en cliquant sur « **Encodage des caractères** » dans le menu « **Affichage** ». **Choisir** le codage « **Occidental** » ou « **Europe Centrale (Windows)** ». **Indiquer** ce qui se passe.
7. **Justifier** pourquoi le caractère « é » (e accent aigu) a été remplacé par les caractères « **Ã©** ».