

|  |  |
| --- | --- |
| ***SIN*** | **SIN – Science Informatique et Numérique** |
| **TP : CODAGE DE L’INFORMATION** | |



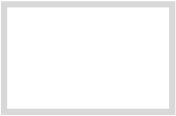
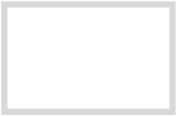
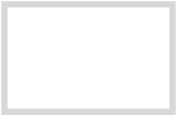
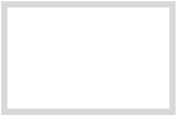
1. [– CODAGE D’UN NOMBRE 2](#_TOC_250016)
   1. [– L’INFORMATION 2](#_TOC_250015)
   2. [– LE SYSTEME BINAIRE 2](#_TOC_250014)
      1. [– Conversion binaire vers décimal 2](#_TOC_250013)
      2. [– Conversion décimal vers binaire 3](#_TOC_250012)
   3. [– LE SYSTEME HEXADECIMAL 3](#_TOC_250011)
      1. [– Conversion hexadécimal vers décimal 4](#_TOC_250010)
      2. [– Conversion décimal vers hexadécimal 4](#_TOC_250009)
      3. [– Conversion binaire vers hexadécimal 4](#_TOC_250008)
      4. [– Conversion hexadécimal vers binaire 5](#_TOC_250007)
   4. [– NOMBRES RELATIFS (ENTIERS SIGNES) 5](#_TOC_250006)
   5. [– NOMBRES A VIRGULE FLOTTANTE (NOMBRES REELS) 6](#_TOC_250005)
2. [– CODAGE BINAIRE D’UN TEXTE 8](#_TOC_250004)
   1. [– PROBLEMATIQUE 8](#_TOC_250003)
   2. [– LE CODE ASCII 8](#_TOC_250002)
   3. [– LES CODES ISO 8859–1 ET ANSI UTILISE PAR WINDOWS 9](#_TOC_250001)
   4. [– CODAGE UNICODE 11](#_TOC_250000)

# 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***

***Double mot (16 bits)***



**0 0 1 0**

**1 0 0 0**

**1 0 0 1**

**1 1 0 0**

***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****.*

***Demi-­‐mot = Quartet (4 bits)***

***Mot = Octet (8 bits)***

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 (2,7.104 ; 1021 ; …).

*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'

### – 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****.*

(1110)2 = (14)10

(10111001)2 = (128+32+16+8+1=186)10

### – Conversion décimal vers binaire

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



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Exemple*** | | | | | | |
| **155 2** |  |  |  |  |  |  |
| **1 77 2** |  |  |  |  |  |  |
| **1 38** | **2** |  |  |  |  |  |
| **0** | **19** | **2** |  |  |  |  |
|  | **1** | **9** | **2** |  |  | **155**  **(10011011)2** |
|  |  | **1** | **4** | **2** |  |  |
| **Sens de lecture** |  |  | **0** | **2** | **2** |  |
|  |  |  |  | **0** | **1** |  |

219-128 1

91-64 1

27-32 0

27-16 1

11-8 1

3-4 0

3-2 1

1-1 1

0d219 => 0b11011011

186-128 1

58-64 0

58-32 1

26-16 1

10-8 1

2-4 0

2-2 1

0-1 0

0d186 => 0b10111010

## 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***.*

### – 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.

0x5D

5\*16=> 80

D=> 13

0x5D => 0d93

0xF3C

F=> 16\*(16\*16)=>4096

3\*16=> 48

C=> 12

0xF3C => 0d4156

### – Conversion décimal vers hexadécimal

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



0d184

184/16 => B - 8

8 =>8 - 0

0d184=>0x8B

0d252

252/16 => E - 12

12 => C

0d252 => 0xEC

### – Conversion binaire vers hexadécimal

|  |  |  |
| --- | --- | --- |
| ***Exemple*** | | |
| **( 0110 1111 0011** | **)2 *Regroupement en quartet*** |  |
|    | ***Conversion binaire hexadécimal*** |  |
| **6 15 3** |  | **(0110 1111)2**  **(6F3)16** |
|    | ***Conversion hexadécimal décima*l** |  |
| **( 6 F 3** | **)16** |  |

### – Conversion hexadécimal vers binaire

|  |  |  |
| --- | --- | --- |
| ***Exemple*** | | |
| **( D 7 E** | **)16** |  |
|    | ***Conversion hexadécimal décimal*** |  |
| **13 7 14** |  | **(D7E)16**  **(1101 0111 1110)2** |
|    | ***Conversion décimal binaire*** |  |
| **( 1101 0111 1110** | **)2** |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Exercice n°6** | | | | | | | | | |
| **Convertir** les (1000 1001)2.  0b1001 1011  1001 => 9  1011 => B  0b10011011=>0x9B | nombres | hexadécimaux | suivants | vers | leur | équivalent | binaire | : | (1001 1011)2, |

0b1000 1001

1000=> 8

1001=> 9

0b10001001 => 0x89

## 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****) :*

|  |  |
| --- | --- |
| **b7 b6 b5 b4 b3 b2 b1 b0** | |
| ***Signe*** | ***Représentation du nombre*** |

*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'additionner 1 au complément du nombre****.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | | | | | | |

0d36

0 +

0

36-32 1

4-16 0

0d-115

1 -

115-64 1

51-32 1

19-16 1

0sb1100 1011

1 -

64

8

2

1

=> 0d-75

0sb0110 1001

0 +

64

32

8

1

=> 0d105

4-8 0

4-4 1

0-2 0

0-1 0

0d36 => 0sb0010 0100

3-8 0

3-4 0

3-2 1

1-1 1

0d-115=>0sb1111 0011

## 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 :

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 :

|  |  |  |
| --- | --- | --- |
| **b63** | **b62 b61 … b53 b52** | **b51 b50 b49 … b2 b1 b0** |
| ***Signe*** | ***Exposant décalé*** | ***Mantisse*** |

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

0dx 41 CA 0000

0|100 0001 1|100 1010 0000 0000 0000 0000

signe 0 => +

exposant => 1000 0011 => 1+2+128 = 131-127 = 4

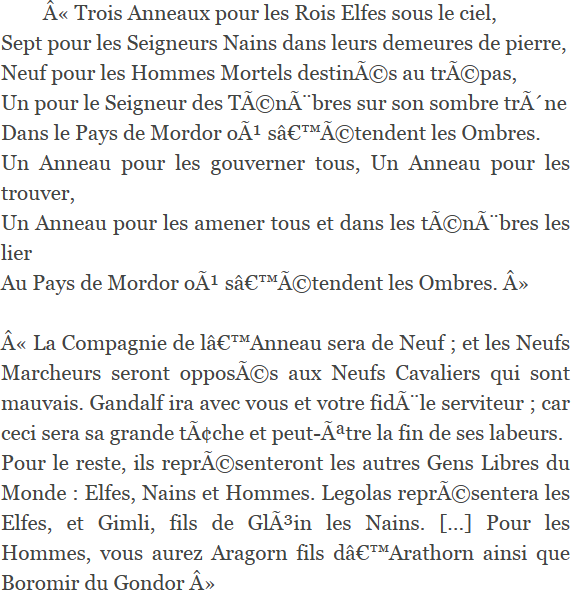
mantisse => 100 1010 0000 0000 0000 0000 => 1/(2^23) + 1/(2^20) + 1/(2^18) =

resultat = (signe)\*1.(mantisse)\*2^(exposant) = 1.

# 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.



## 2.2 – LE CODE ASCII

*Le code* ***ASCII*** *(****A****merican* ***S****tandard* ***C****ode for* ***I****nformation* ***I****nterchange), 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* ***27*** ***128 combinaisons*** *possibles).*

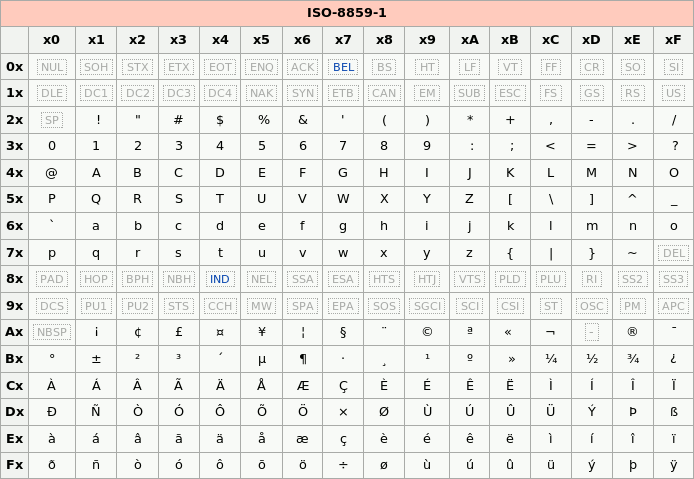
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x0** | | **x1** | **x2** | **x3** | **x4** | **x5** | **x6** | **x7** | **x8** | **x9** | **xA** | **xB** | **xC** | **xD** | **xE** | **xF** |
| **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** |

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…

## 2.3 – LES CODES ISO 8859–1 ET ANSI UTILISE PAR WINDOWS

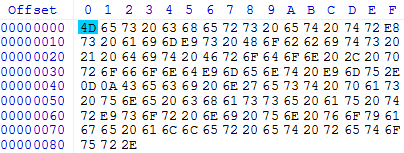
La nécessité de représenter des caractères non présents dans la table ASCII tels que ceux de l’alphabet latin comme le « à », le « é », « ç »… impose l’utilisation d’un autre code.

Ces codes sont des extensions du code ASCII. Pour cela le **8ième 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**.



Microsoft propose le codage **Windows-­‐1252** appelé aussi code **ANSI** (**A**merican **N**ational **S**tandard **I**nstitute). Ce code ne diffère de l’**ISO-­‐8859-­‐1** que pour quelque 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.

1. **Lancer** le logiciel **Free Hex editor** . **Ouvrir** le fichier **texte1.txt**.
2. **Justifier** la valeur des octets n° **0x3f** et **0x40**.

##### **Donner** le codage binaire du nom « Bilbon ».

1. **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.
2. **Ouvrir** le fichier **texte2.doc** avec **Word**. **Noter** bien le texte de ce fichier.

**Octets n° 0x40 et 0x41**

1. **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**.
2. **Lancer** le logiciel **Free Hex editor**. **Ouvrir** le fichier **texte2.doc**.

##### **Préciser** s’il est possible de « déchiffrer » facilement le contenu comme pour le fichier **texte1.txt**.

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

1. **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.
2. **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.**
3. **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.*](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***  ***Décimale Binaire*** | | ***Code UTF-­‐8 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***  ***Décimale Binaire*** | | ***Code UTF-­‐8 Binaire*** | ***Hexadécimal*** |
| **é** | **233** | **00011 101001** | **11000011 10101001** | **C3 A9** |
| **€** | **8364** | **0010 000010 101100** | **11100010 10000010 10101100** | **E2 82 AC** |