# TD 1 — Codage de l'information — Correction

## Exercice 1 — Changement de base

**Q. 1.1** Convertir en nombres décimaux les nombres binaires suivants : 11, 1101, 100101110.

**Correction**

- $11_2 = 2^1 + 2^0 = 3$
- $1101_2 = 2^3 + 2^2 + 2^0 = 13$
- $100101110_2 = 2^8 + 2^5 + 2^3 + 2^2 + 2^1 = 302$

◆

**Q. 1.2** Convertir en nombres binaires les nombres décimaux suivants : 7, 51, 128, 131, 234.

**Correction**

Pour les 4 premiers nombres on va directement décomposer en puissance de 2 :

- $7 = 4 + 2 + 1 = 2^2 + 2^1 + 2^0 = 111_2$
- $51 = 32 + 16 + 2 + 1 = 2^5 + 2^4 + 2^1 + 2^0 = 11\ 0011_2$
- $128 = 2^7 = 1000\ 0000_2$
- $131 = 2^7 + 2^1 + 2^0 = 1000\ 0011_2$

Pour 234 on va appliquer la méthode donnée dans le cours qui repose sur des divisions successives :

- $234 = 2 \times 117 + \mathbf{0}$
- $117 = 2 \times 58 + \mathbf{1}$
- $58 = 2 \times 29 + \mathbf{0}$
- $29 = 2 \times 14 + \mathbf{1}$
- $14 = 2 \times 7 + \mathbf{0}$
- $7 = 2 \times 3 + \mathbf{1}$
- $3 = 2 \times 1 + \mathbf{1}$
- $1 = 2 \times 0 + \mathbf{1}$

Ce sont les restes des divisions qui forment les bits du résultat. Le premier reste obtenu donne le bit de poids faible, le second donne celui de poids 1 et ainsi de suite. Donc $234 = 11101010_2$.

◆

**Q. 1.3** Convertir en nombres binaires puis en nombres décimaux les nombres hexadécimaux suivants : 12, *DADA* et $5F3$.

**Correction**

Pour la conversion hexadécimal→binaire :

- $12_{16} = 0001\ 0010_2$
- $DADA_{16} = 1101\ 1010\ 1101\ 1010_2$
- $5F3_{16} = 0101\ 1111\ 0011_2$

Pour la conversion hexadécimal→décimal :

- $12_{16} = 1 \times 16^1 + 2 \times 16^0 = 18$
- $DADA_{16} = 13 \times 16^3 + 10 \times 16^2 + 13 \times 16^1 + 10 \times 16^0 = 56\,026$
- $5F3_{16} = 5 \times 16^2 + 15 \times 16^1 + 3 \times 16^0 = 1\,523$

◆

## Exercice 2 — Opérations binaires

On travaille dans cet exercice sur des entiers naturels codés sur un octet.

**Q. 2.1** Quels sont les entiers naturels que l'on peut représenter sur un octet ?

**Correction**

Avec un octet (8 bits) on peut coder $2^8$ valeurs : tous les entiers naturels de 0 (codé 0000 0000 en binaire) à $2^8 - 1 = 255$ (codé 1111 1111 en binaire).

◆

**Q. 2.2** Donner le résultat des opérations suivantes en binaire puis en décimal.

(a) $0001\ 0101_2 + 1011\ 0111_2$
(b) $0100\ 0111_2 + 1101\ 1001_2$
(c) $134_{10}$ **ET** $244_{10}$
(d) $17_{10}$ **OU** $123_{10}$
(e) **NON** $27_{10}$
(f) $44_{10}$ **XOR** $157_{10}$

**Correction**

(a) Calcul de $0001\ 0101_2 + 1011\ 0111_2$

$$\begin{array}{cccccccccc} & & 1 & 1 & & 1 & 1 & 1 & & \text{retenues} \\ & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & \\ + & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & \\ \hline & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & = 2^7+2^6+2^3+2^2 = 204_{10} \end{array}$$

(b) Calcul de $0100\ 0111_2 + 1101\ 1001_2$

$$\begin{array}{cccccccccc} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & & \text{retenues} \\ & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & \\ + & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & \\ \hline & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & = 2^5 = 32_{10} \end{array}$$

On observe ici un dépassement de capacité : le résultat est codé sur 9 bits alors qu'on ne dispose que de 8 bits. Le résultat de l'opération est donc erroné.

(c) Calcul de $134_{10}$ **ET** $244_{10}$
$134_{10} = 1000\ 0110_2$ et $244_{10} = 1111\ 0100_2$

$$\begin{array}{cccccccccc} & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \\ \textbf{ET} & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & \\ \hline & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & = 2^7+2^2 = 132_{10} \end{array}$$

(d) Calcul de $17_{10}$ **OU** $123_{10}$
$27_{10} = 0001\ 0001_2$ et $123_{10} = 0111\ 1011_2$

$$\begin{array}{cccccccccc} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \\ \textbf{OU} & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & \\ \hline & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & = 2^6+2^5+2^4+2^3+2^1+2^0 = 123_{10} \end{array}$$

(e) Calcul de **NON** $27_{10}$
$17_{10} = 0001\ 1011_2$

$$\begin{array}{cccccccccc} \textbf{NON} & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & \\ \hline & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & = 2^7+2^6+2^5+2^2 = 228_{10} \end{array}$$

(f) Calcul de $44_{10}$ **XOR** $157_{10}$
$44_{10} = 0010\ 1100_2$ et $157_{10} = 1001\ 1101_2$

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & \\ \textbf{XOR} & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & \\ \hline & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & = 2^7+2^5+2^4+2^0 = 177_{10} \end{array}$$

◆

**Q. 2.3** Soit $x$ un octet quelconque. Que vaut le résultat (en binaire ou en décimal) des opérations suivantes :

(a) $x$ **OU** $255_{10}$
(b) $x$ **ET** $255_{10}$
(c) **NON** $x$

**Correction**

(a) $255_{10} = 1111\ 1111_2$. Or comme 1 **OU** $x = 1$ quel que soit $x$, on aura toujours $x$ **OU** $255_{10} = x$ **OU** $1111\ 1111_2 = 1111\ 1111_2 = 255$.
(b) Comme 1 **ET** $x = x$ quel que soit $x$, on aura toujours $x$ **ET** $255_{10} = x$ **ET** $1111\ 1111_2 = x$.
(c) $255 - x$

◆

## Exercice 3 — Décalage binaire

Coder en binaire les nombres 26 et 52. Que remarque-t-on ? En déduire une méthode rapide pour multiplier ou diviser par $2^k$ un nombre binaire. Généraliser à une base $B$ quelconque.

**Correction**

Le codage donne
— $26_{10} = 11010_2$
— $52_{10} = 110100_2$
On remarque que la représentation binaire de 52 est obtenue à partir de celle de 26 en rajoutant un bit de poids faible à 0. On appelle cette opération un décalage à gauche.

De manière générale, en rajoutant $k$ bits de poids faible à 0 on multiplie le nombre par $2^k$. Avant le décalage, la puissance de 2 associée au bit de poids $p$ est $2^p$. Après le décalage, cette puissance devient $2^{k+p}$. On a donc multiplié par $2^k$ chacune des puissances associées de même que la somme des puissances.

En suivant le même raisonnement, on déduit que, dans une base $B$ quelconque, décaler à gauche de $k$ chiffres revient à multiplier par $B^k$. Par exemple, en hexadécimal, $D3_{16} = 211$ et $D300_{16} = 211 \times 16^2 = 54\,016$. ◆

## Exercice 4 — Jeu de cartes

On s'intéresse à des jeux de 52 cartes réparties en 4 couleurs (pique, cœur, carreau et trèfle) de 13 cartes désignées par leurs rangs (As, 2, 3, …, 10, Valet, Dame et Roi).

**Q. 4.1** Proposer un schéma de codage binaire des cartes du jeu.

**Correction**

Il y a quatre couleurs. Pour coder ces quatre couleurs on a besoin de 2 bits (car $2^2 \geq 4$). On peut ensuite utiliser le codage suivant : $00 \to$ pique, $01 \to$ cœur, $10 \to$ carreau et $11 \to$ trèfle.
Pour coder les 13 rangs on a besoin de 4 bits (3 bits ne sont pas suffisants car $2^3 = 8$ mais avec 4 bits on peut en coder $2^4 = 16 \geq 13$). On peut ensuite utiliser le codage suivant : $0000 \to$ As, $0001 \to 2$, $0010 \to 3$, …, $1100 \to$ Roi.
Une carte peut donc être codée avec 6 bits $c_1, c_0, r_3, r_2, r_1, r_0$, les bits $c_1, c_0$ indiquant la couleur et $r_3, r_2, r_1, r_0$ le rang. ◆

**Q. 4.2** Donner, dans ce schéma, la représentation binaire du valet de trèfle.

**Correction**

Avec la représentation choisie, on a $c_1c_0 = 11$ pour la couleur trèfle et $r_3r_2r_1r_0 = 1010$ pour le valet. Donc 111010 est la représentation binaire du valet de trèfle. ◆

**Q. 4.3** On considère deux cartes dont les représentations binaires sont $C_1$ et $C_2$. Sous quelles conditions, ces deux cartes ont-elle la même couleur ? le même rang ? On utilisera l'opérateur binaire ET pour déterminer ces conditions.

**Correction**

(a) Pour obtenir la couleur d'une carte à partir de sa représentation binaire, on peut effacer (ou mettre à 0) les bits de poids faible $r_3r_2r_1r_0$ qui servent à coder le rang de la carte.
On peut utiliser le fait que, quel que soit $x$, $x$ **ET** $0 = 0$ pour mettre à 0 les bits du rang et que $x$ **ET** $1 = x$ pour garder uniquement les bits de la couleur.
La couleur d'une carte dont la représentation binaire est $C$ peut donc être obtenue avec l'opération suivante : $C$ **ET** 110000.
Les deux cartes ont donc la même couleur si $C_1$ **ET** $110000 = C_2$ **ET** 110000.

(b) En utilisant le même raisonnement, on trouve que les deux cartes ont le même rang si $C_1$ **ET** $001111 = C_2$ **ET** 001111 ◆

## Exercice 5 — Cryptage

Lorsque l'on envoie un message sur Internet et qu'on ne veut pas que ce message puisse être intercepté et compris par une personne autre que le destinataire on doit utiliser une méthode (ou un algorithme) de *cryptage*. Le principe général est d'altérer le message selon une méthode connue de l'émetteur et du récepteur et d'envoyer sur le réseau le message ainsi modifié. Bien entendu, seul le récepteur doit être en mesure de décrypter le message reçu pour retrouver le message initial, c'est-à-dire avant sa transformation par l'opération de cryptage.

Certains algorithmes de cryptage reposent sur l'utilisation d'une *clé* qui est une séquence de bits quelconque connue uniquement de l'émetteur et du destinataire du message. Le message crypté envoyé sur le réseau est obtenu en transformant le message initial à l'aide de cette clé (selon une méthode connue de l'émetteur et du récepteur) de même que le décryptage par le destinataire est obtenu en transformant le message crypté à l'aide de cette clé.

Un algorithme de cryptage très simple consiste à crypter le messsage en utilisant une clé $c$ de 8 bits et en transformant chaque octet $m$ du message par l'octet $m$ **XOR** $c$. Le destinataire effectue la même opération : il transforme chaque octet $m'$ qu'il a reçu par l'octet $m'$ **XOR** $c$ pour retrouver l'octet du message initial.

Cette méthode peut naturellement être généralisée pour une longueur de clé quelconque, et pas uniquement de 8 bits. Dans cet exercice, on s'intéresse au cryptage d'un texte codé en ASCII (donc avec des caractères codés sur 7 bits) à l'aide d'une clé de 7 bits.

**Q. 5.1** On veut envoyer le mot `yop`. Donner, à l'aide de la table ASCII donnée dans le cours, la séquence binaire correspondant à ce mot.

**Correction**

$$\begin{aligned} y &= \underbrace{111}_{7}\ \underbrace{1001}_{9} \\ o &= \underbrace{110}_{6}\ \underbrace{1111}_{F} \\ p &= \underbrace{111}_{7}\ \underbrace{0000}_{0} \end{aligned}$$

La séquence binaire est donc 1111001 1101111 1110000. ◆

**Q. 5.2** On utilise la clé 55. Quel sera alors le message envoyé après cryptage (en binaire et en texte) ?

**Correction**

La clé en binaire vaut 0110111. Les trois mots de 7 bits obtenus après cryptage sont :
— 1111001 **XOR** 0110111 = 1001110
— 1101111 **XOR** 0110111 = 1011000
— 1110000 **XOR** 0110111 = 1000111
Le message crypté est donc 1001110 1011000 1000111 en binaire.
Pour obtenir le message crypté en texte :
— $1001110_2 = 4E_{16}$ soit le caractère ASCII 'N'
— $1011000_2 = 58_{16}$ soit le caractère ASCII 'X'
— $1000111_2 = 47_{16}$ soit le caractère ASCII 'G'
Le texte crypté est donc `NXG`. ◆

**Q. 5.3** Vérifier qu'après décryptage, le destinataire retrouve bien le message initial.

**Correction**

Le décryptage donne :
— 1001110 **XOR** 0110111 = 1111001
— 1011000 **XOR** 0110111 = 1101111
— 1000111 **XOR** 0110111 = 1110000
On retrouve bien la séquence binaire initiale. ◆

**Q. 5.4** Prouver à l'aide d'une table de vérité que le destinataire peut toujours retrouver le message initial à partir de la clé et du message crypté.

**Correction**

Soit $m$ un mot de 7 bits du message non crypté et $c$ la clé. Le mot crypté est $m$ **XOR** $c$ et celui obtenu après décryptage est ($m$ **XOR** $c$) **XOR** $c$. La méthode est donc correcte si $m = (m \textbf{ XOR } c) \textbf{ XOR } c$. La table de vérité ci-dessous montre que c'est toujours le cas.

| $m$ | $c$ | $m$ **XOR** $c$ | ($m$ **XOR** $c$) **XOR** $c$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |

Intuitivement, un 1 dans la clé signifie qu'on inverse le bit dans le mot à crypter et un 0 signifie qu'on ne change pas ce bit. Dans les deux cas, appliquer deux fois l'opérateur XOR permet donc de retrouver le bit initial. ◆

**Q. 5.5** En pratique, une longueur de clé de 7 bits est insuffisante. On utilise des clés d'au moins 32 ou 64 bits. Pourquoi ?

**Correction**

Avec une longueur de clé de 7 bits, il n'y a que $2^7 = 128$ clés possibles, ce qui fait que 128 tentatives sont suffisantes pour décrypter le message crypté. Un programme de décryptage qui tente de décrypter le message en tentant avec toutes les clés possibles peut générer ces 128 solutions possibles en quelques milli-secondes. (Il faut bien entendu qu'il puisse analyser ensuite si le message décrypté est bien un message correct.)
Avec une clé de $2^{32}$ bits, on monte à environ 4 milliards de clés possibles. C'est mieux mais avec un ordinateur relativement puissant, on peut espérer décoder le message en quelques heures. ◆