[CYCLE 6 – BINAIRE]

RAPPEL BINAIRE

1. Pourquoi coder des informations ?

- Coder une **image**, du **son**, une **vidéo** (**numérisation**)
- Coder les **caractères alphanumériques** : de 0 à 9, de a à z en minuscule et MAJUSCULE, la **ponctuation**, les **caractères spéciaux**, les **caractères cachés** (ex : retour en début de ligne **c**arriage **r**eturn Cr)…etc
- Coder un prix **code-BARRES**
- Coder une adresse internet **Flash code**

2. Principe et unités du codage

- Tout **ordinateur**, du plus rudimentaire au plus puissant, n’est qu’une machine électronique qui en tant que telle n’est capable de traiter que **2 informations : *le courant électrique passe (symbolisé par le chiffre 1) ou ne passe pas (symbolisé par le chiffre 0).***

- L’unité élémentaire d’information, qui peut prendre la **valeur 0 ou 1,** *est appelée* ***bit*** *(contraction de* ***b****inary* ***d****igit qui signifie chiffre binaire en anglais).* A partir de cette unité élémentaire on compose des unités plus complexes, qui sont récapitulées dans le tableau ci-dessous.

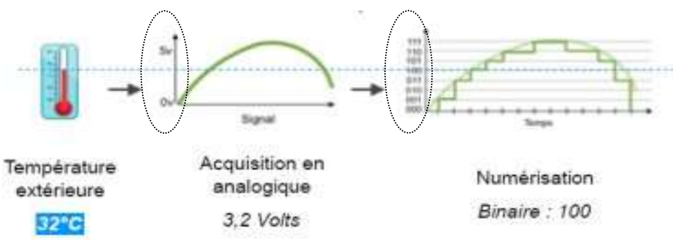
	Notation	Valeur
Bit		*1 ou 0*
Nibble		*4 bits consécutifs*
Octet (byte en anglais)	*O*	*8 bits consécutifs*
Kilo Octet	*Ko*	*1000 octets*
Méga Octet	*Mo*	*1 million d’octets*
Giga Octet	*Go*	*1 milliard d’octets*
Téra Octet	*To*	*1000 milliards d’octets*

- **Remarque :** Les unités de dénombrement de la quantité d’information présentées dans le tableau ci-dessus sont les unités normalisées par l’organisme international IEC afin d’être en accord avec le système international d’unités. Ce sont ces unités qui devraient être utilisées. Cependant dans l’ancienne notation, qui continue à être assez largement utilisée en dépit des recommandations officielles, on considère que 1 Ko = 2^{10} octets = 1024 octets, 1 Mo=2^{10} Ko=1 048 576 octets, 1 Go =2^{10} Mo, etc., alors que ces unités basées sur les puissances de 2 devraient être notées Kio, Moi et Gio. Si l’ordre de grandeur reste le même entre ces 2 systèmes d’unités, ceci peut expliquer certaines divergences dans le calcul de tailles de fichiers.

3. Numérisation (digitalisation ou conversion analogique numérique) d’un objet physique :

- Elle se fait par l’intermédiaire de **capteurs** qui permettent **de mesurer les propriétés jugées utiles** de l’objet à **numériser**.

Objet réel	Objet intermédiaire (capture acquisition)	Objet virtuel
Scène 3D statique	*Appareil photo numérique*	**Image numérique**
Photographie papier	*Scanner*	**Image numérique**
Document imprimé	*Scanner + application de reconnaissance des caractères (OCR)*	**Document texte brut ou texte mis en forme**
Son : variation de la pression de l’air	*Microphone + carte son*	**Son numérique**
Scène 3D en mouvement	*Caméra numérique + carte d’acquisition*	**Vidéo numérique**



4. Conversion binaire décimal

Nos mathématiques classiques sont appelées « **décimales** » : tous les nombres peuvent être écrits avec **0,1,2,3,4,5,6,7,8,9 (soit 10 chiffres)** et **décomposés en puissances de 10.**

En mathématiques **binaires**, tous les nombres peuvent être écrits **avec 0 et 1** uniquement (2 chiffres) et décomposés en **puissances de 2**.

5. Pour convertir un nombre binaire en nombre décimal, il faut connaître les puissances de 2^n

Attention : $2^0=1$!

__Exemples__ :
Que vaut en décimal le **nibble 1101** ?
On utilise le tableau suivant :

Bit	**1**	**1**	**0**	**1**
Poids du bit	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Valeur du bit x poids du bit	1x8 = 8	1x4 = 4	0x2 = 0	1x1 = 1
Décimal	**8+4+0+1= 13**			

Que vaut en décimal **l'octet 10101101** ?
On utilise le tableau suivant :

Bit	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**1**
Poids du bit	2^7=128	2^6=64	2^5=32	2^4=16	2^3=8	2^2=4	2^1=2	2^0=1
Valeur du bit x poids du bit	1x128=128	0x64=0	1x32=32	0x16=0	1x8=8	1x4=4	0x2=0	1x1=1
Décimal	**128+0+32+0+8+4+0+1 = 173**							

Si on rajoutait un bit à gauche de l'octet, son poids serait de 2^8, etc.

Attention : le bit de plus faible poids est celui __le plus à droite__ quand on écrit un nombre binaire.

Exercices :

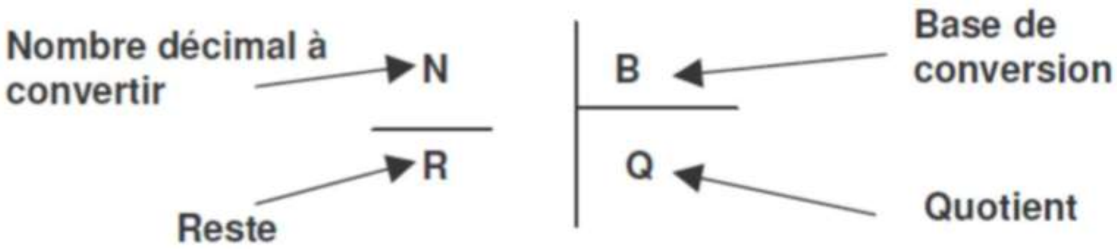
$1001\ 1011_{(2)}$ =

$0011\ 1100_{(2)}$ =

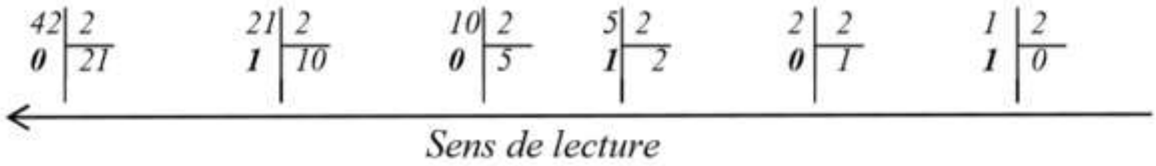
$0101\ 0101_{(2)}$ =

6. Conversion d’une base décimale en une base quelconque

Pour **convertir** un **nombre décimal** en **base quelconque** on divise le nombre par la base puis le **quotient** par la **base**, jusqu’à ce qu’on ait un **quotient** qui soit **nul**. On a le résultat en lisant **les restes de chaque division** en commençant par le **dernier obtenu** qui est le **poids fort**.



Exemple : soit à convertir en binaire le nombre $42_{(10)}$



$42_{(10)} = 101010_{(2)}$

Vérification : convertir en décimal $\mathbf{101010_2} = 1.2^5 + 0.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 0.2^0 = 42_{(10)}$