But : être capable de changer de base un nombre codé en décimal, binaire ou hexadécimal. Utiliser un codage pour un entier signé, ou un nombre à virgule flottante.

Vu de l'extérieur, les ordinateurs et les programmes que nous utilisons tous les jours permettent de mémoriser, de transmettre et de transformer des nombres, des textes, des images, des sons, etc.
Pourtant, un ordinateur ne manipule que des 0 et des 1…
Il va donc falloir **coder**, c'est-à-dire représenter comme des suites de 0 et de 1 les nombres, les textes, les images, les sons, etc. Une telle valeur, 0 ou 1, s'appelle un **booléen**, un **chiffre binaire** ou encore un **bit** (*binary digit)*. Une suite de bits, par exemple 0000110100 , est appelé un **mot**. Vous allez donc aujourd'hui apprendre à coder en binaire des nombres.

1 Les systèmes de numérations : représentation des entiers naturels

1.1 Numération décimale :

La numération décimale utilise 10 chiffres : **0, 1, 2, 3, 4, 5, 6, 7, 8 et 9**.
L'écriture du nombre $(275)_{10}$ se traduit par $275= 2x10^2 + 7x10^1 + 5*10^0$, on dit que 275 est la représentation en base 10 appelée écriture décimale.

1) Ecrire une égalité semblable pour les nombres 1 234 et 208.

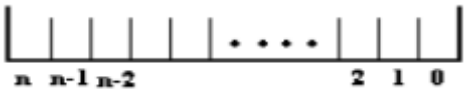
1.2 Numération binaire :

L'informatique utilise des courants électriques, des aimantations, des rayons de lumière…
Chacun de ces phénomènes met en jeu deux états possibles. :

- Tension nulle ou tension non nulle (5V par ex),
- Aimantation dans un sens ou dans l'autre sens,
- Lumière ou pas de lumière.

Il suffit de deux chiffres pour traduire ces états : c'est la numération binaire qui utilise les chiffres **0** et **1**.
L'écriture du nombre $(1110)_2$ se traduit par $1110 = 1x2^3+1x2^2+1x2^1+0x2^0=8+4+2+0=14$, on dit que 1110 est la représentation en base 2 de 14.
Chaque chiffre binaire se nomme BIT (de BInary digiT)

Voici le schéma d'une mémoire à n+1 bits.
Les cases du schéma représentent les bits, le chiffre marqué en-dessous d'une case indique la puissance de 2 à laquelle est associé ce bit (on dit aussi rang du bit).



- Le bit de **rang 0** est appelé le bit de **poids faible.**
- Le bit de **rang n** est appelé le bit de **poids fort**.

2) Traduire en écriture décimale les nombres binaires 0101 et 11001.

Un nombre binaire de huit chiffres est un **octet**.

3) Quel est le plus grand nombre que l'on peut représenter avec un octet ?

4) Justifier qu'avec un **mot** de **n** bits, on peut représenter les nombres de 0 à 2^n - 1.

Pour multiplier par dix un entier naturel exprimé en base 10, il suffit d'ajouter un 0 à sa droite, par exemple 12 x 10 = 120

5) Quelle est l'opération équivalente pour les entiers naturels exprimés en base 2? Exprimer en base 2 les nombres 3, 6, et 12 pour illustrer cette réponse.

Les multiples et sous-multiples des unités employées en informatique

	Valeur exacte	en pratique dans le système international SI	Exemples d'utilisation
1 kilo-octet(Ko)	2^{10} = 1024 octets	10^3 octets	un modem téléphonique débite à 56,6 Kbps, un téléphone GSM à 9.6 Kbps une page de texte correspond environ 2 Ko.
1 Méga-octet(Mo)	2^{10} Ko = 1 048 576 octets	10^6 octets	un CDROM contient 640 Mo
1 Giga-octet (Go)	2^{10} Mo =1 073 741 824 octets	10^9 octets	1 DVD a une capacité de stockage de 4,7 Go 10 mètres de livres dans une bibliothèque tiendraient sur environ 1 Go un étage rempli complètement de journaux équivaut à 100 Go, c'est à dire au disque dur d'un PC
1 Téra-octet (To)	2^{10} Go =1 099 511 627 776 octets	10^{12} octets	une grande bibliothèque publique tiendrait sur 1 To environ l'ordinateur Blue Gene/L d'IBM calcule à la vitesse de 36Tflops (36 Téra opérations par seconde) YouTube annonce plus de mille milliards de vidéos vues

1.3 Numération hexadécimale :

La numération hexadécimale utilise 16 chiffres : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F**.
A représente donc le nombre "10", B le nombre "11"....
L'écriture du nombre $(23A)_{16}$ se traduit par $23A= 2x16^2 + 3x16^1 + 10*16^0=512+48+10=570$.
On dit que 23A est la représentation en base 16 de 570 .

6) Traduire en nombre décimal $(B8C)_{16}$

7) A quel nombre décimal correspond le nombre FF?

8) Quelle remarque faites-vous par rapport à l'octet?

La taille rapidement encombrante de l'écriture binaire a conduit les informaticiens à faire un usage intensif du système hexadécimal.

2 Changement de base :

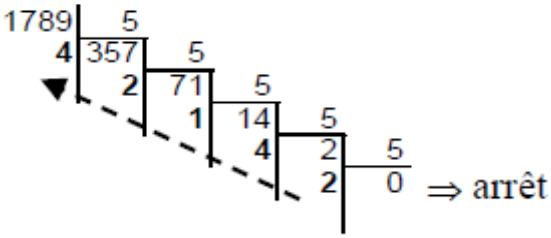
2.1 Passage de la base b à la base 10:

$(a_k a_{k-1}....a_2 a_1 a_0)_b$ (en base b) désigne l'entier $a_k \times b^k + a_{k-1} \times b^{k-1} + ...+ a_2 \times b^2 + a_1 \times b^1 + a_0 \times b^0$.

9) Donner l'écriture décimale de $(76401)_8$ et $(BA9865)_{12}$.

2.2 Passage de la base 10 à la base b:

Pour écrire les entiers naturels en base *b*, on a besoin de *b* chiffres. Quand on a *n* objets, on les groupe par paquets de *b*, qu'on regroupe à leur tour en paquets de *b* paquets, etc.
Autrement dit, on fait une succession de divisions par *b*, jusqu'à obtenir un quotient égal à 0.



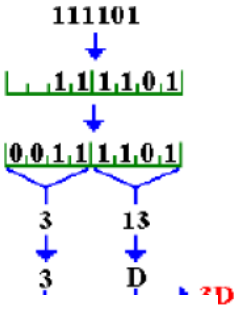
donc 1789 s'écrit $(24124)_5$

10) Trouver la représentation en base 16 puis en base 2 des nombres 6725 et 18 379 (écrits en base 10)

2.3 Passerelle entre la base 2 et la base 16:

Pour passer de la base 2 à la base 16 :

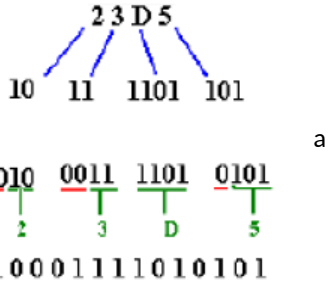
- On décompose ce nombre par tranches de 4 bits à partir du bit de poids faible ($2^4 = 16$)
- On complète la dernière tranche (celle des bits de poids forts)par des 0 s'il y a lieu.
- On convertit chaque tranche en son symbole de la **base 16.**
- On réécrit à sa place le nouveau symbole par changements successifs de chaque groupe de 4 bits.



Donc $(111101)_2 = (3D)_{16}$

Pour passer de la base 16 à la base 2 :
Cette conversion est l'opération inverse de la précédente.

- On convertit chaque symbole hexadécimal du nombre en son écriture binaire (nécessitant au plus 4 bits).
- Pour chaque tranche de 4 bits, on complète les bits de poids fort par des 0 s'il y a lieu.
- On regroupe toutes les tranches de 4 bits à partir du bit de poids faible, sous forme d'un seul nombre binaire.



Donc $(23D5)_{16} = (10001111010101)_2$

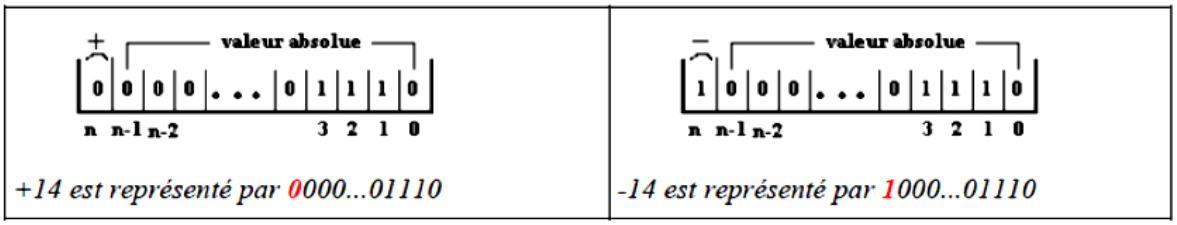
11) Convertir $(1100011010)_2$ en base 16 et $(7CF21)_{16}$ en base 2.

3 Représentation des entiers relatifs (entiers signés):

Pour représenter les entiers relatifs en notation binaire, on doit étendre la représentation des positifs aux nombres négatifs.
1ère solution avec les premières machines :

Une solution est de réserver le bit de poids fort pour le signe de l'entier à représenter (1 pour – et 0 pour +) et d'utiliser les autres pour représenter sa valeur absolue:

Exemple du codage en binaire signé des nombres **+14** et **-14** :

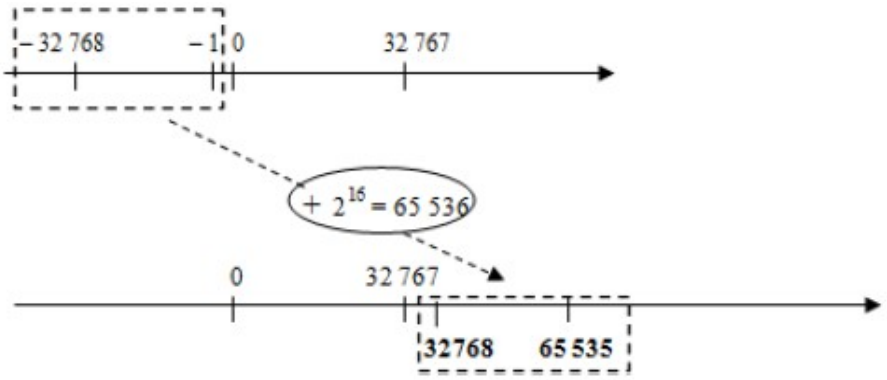


12) Ainsi, avec des mots de 16 bits, en utilisant 1 bit pour le signe et 15 pour la valeur absolue, quels sont les entiers que l'on peut représenter?

Ce codage n'est pas utilisé en pratique. En effet le traitement spécifique du signe coûte cher en circuits électroniques et en temps de calcul. C'est une version améliorée qui est utilisée dans la plupart des calculateurs : elle se nomme le complément à deux.

2ème solution : codage sur 16 bits :

Si on utilise des mots de 16 bits, cette fois-ci les nombres 1000 0000 0000 0000 et 0000 0000 0000 0000 représente 2 nombres distincts donc on peut représenter les entiers relatifs compris entre -2^{15} = – 32 768 et $2^{15}-1$ =32 767. L'idée est de transporter les entiers strictement négatifs dans le monde des entiers naturels et de laisser intact les entiers positifs (voir ci-dessous) :



Avec cette méthode, tous les entiers relatifs sont donc représentés par un entier naturel.

13) Comment est alors représenté l'entier négatif -1?

Cette manière de représenter les entiers relatifs s'appelle le notation en **complément à deux**.

14) Quels entiers relatifs peut-on représenter avec des mots de 8 bits? Combien sont-ils?

15) Même question pour 32 et 64 bits.

La représentation binaire sur n bits d'un entier relatif donné en décimal est :

- Si l'entier relatif x est positif ou nul, on le représente comme l'entier naturel x.
- S'il est strictement négatif, on le représente comme l'entier naturel $x + 2^n$.

Sur 8 bits :
L'entier relatif 0 est représenté comme l'entier naturel 0 soit 0000 0000.
L'entier relatif -128 est représenté comme l'entier naturel $-128 + 2^8 = -128 + 256 = 128$ soit 1000 0000.

16) Trouver la représentation binaire sur 8 bits des entiers relatifs 127 et -127.

La représentation décimale d'un entier relatif donné en binaire sur n bits :
Si cet entier relatif est donné par le mot m, on commence par calculer l'entier naturel p représenté par ce mot.

- Si p est strictement inférieur à 2^n-1, c'est l'entier relatif représenté,
- S'il est supérieur ou égal à 2^n-1, l'entier relatif représenté est $p - 2^n$.

Sur 8 bits :
Le mot 0000 0000 représente l'entier naturel 0 et donc l'entier relatif 0.
Le mot 1000 0000 représente l'entier naturel $128 = 2^7$ et donc l'entier relatif $128 - 2^8 = 128 - 256 = -128$.

17) Trouver la représentation décimale des entiers relatifs dont la représentation binaire sur huit bits est 0111 1111 et 1000 0001.

4 Représentation des nombres à virgule :

En notation décimale, on utilise la notation scientifique pour représenter un nombre décimal : $a \times 10^p$ avec $1 \leq a < 10$ et $p \in Z$.
Par exemple : $52980480 = 5,2980480 \times 10^7$
De manière analogue, nous allons utiliser cette représentation en base 2 :
$s\ m\ 2^n$
où s est le signe du nombre : + ou –,
n son exposant : $n \in Z$
et m sa mantisse : $1 \leq m < 2$
Quand on utilise 64 bits pour représenter un nombre à virgule, on utilise 1 bit pour le signe, 11 bits pour l'exposant et 52 bits pour la mantisse. Le signe + est représenté par 0 et le signe - par 1.

18) Est-il nécessaire d'utiliser les 52 bits pour coder toute la mantisse ? Justifier

Le nombre représenté est donc $-\dfrac{206727}{131072}\times 2^{71} = -3{,}724\ldots\ldots\times 10^{21}$

19) Trouver le nombre à virgule représenté par le mot :
0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.

20) Comment est représenté le nombre entier 7 ? Et le nombre à virgule 7.0 ?

21) Quel est le plus grand nombre à virgule que l'on peut représenter en binaire ?

22) Quel est le plus petit nombre à virgule, donc négatif, que l'on peut représenter en binaire ?

23) Quel est le plus petit nombre à virgule strictement positif que l'on peut représenter en binaire?