Table of Contents

1 Introduction

Qu'est-ce un nombre ?

Une question qu'on ne se demande pas souvent est par rapport aux nombres et comment on les écrit. Auparavant, on avait un système des chiffres « romains ». Dans ce système, on écrivait :

I	un
II	deux
III	trois
IV	quatre
V	cinq
VI	six
VII	sept
VIII	huit
IX	neuf
X	dix
XI	onze
XII	douze

XIX	dix-neuf
XX	vingt
XXI	vingt et un
XXV	vingt-cinq
XXX	trente
XL	quarante
L	cinquante
LX	soixante
LXX	soixante-dix
LXXX	quatre-vingts
XC	quatre-vingt-dix
C	cent
CX	cent dix
CXX	cent vingt
CL	cent cinquante
CC	deux cent
CCC	trois cent
D	cinq cent
M	mille

Dans ce système, chaque chiffre a une valeur différente. On les rassemble plusieurs pour exprimer une valeur. On rassemble de plus grand au plus petit, donc CCXVI veut dire « deux cent seize ». Au lieu d'écrire VIIII pour neuf, on écrit IX, c'est-à-dire « dix moins un ». Donc pour écrire « deux cent quatre-vingt-dix-neuf », on écrit CCXCIX (« deux cent, puis cent moins dix, puis dix moins un »). Nous voyons que ce système, alors que c'est très différent de ce qu'on connaît, fonctionne. Tout nombre qu'on peut écrire avec notre système (ex: 3942) peut s'écrire aussi avec les chiffres romains (ex: MMMCMXLII). Mais les opérations sont difficiles. (Combien font MMMCMXLII divisé par IX?)

Un nombre est quelque chose de pur, d'abstract. Cinq bananes sont différent de cinq voitures, mais les deux sont « cinq ». Mais pour écrire ces nombres, il faut un système. Plusieurs systèmes sont possibles. Le système qu'on utilise dans la vie quotidienne s'appelle décimal. Nous avons déjà vu le système de **chiffres romains**. En 3e, on étudie les autres systèmes propre à l'informatique : binaire, hexadécimal, octal.

2 Décimal

Quel est la différence entre 4392 et 4329 ?

Alors que les romains anciens utilisaient les chiffres romains, vers le 14e siècle on a commence à emprunter les chiffres arabes – le système qu'on utilise aujourd'hui. Alors que les romains utilisaient les chiffres différents pour dire "mille" et "cent", les arabes utilisaient un système des **places**. Dans ce système, il n'y a que dix chiffres :

1	un
2	deux
3	trois
4	quatre
5	cinq
6	six
7	sept
8	huit
9	neuf
0	zéro

N.B. On a introduit un nouveau chiffre pour « zéro ». Cela était une grande invention dans le monde de mathématiques ! Pourquoi avoir un nombre pour quelque chose qu'il n'y a pas ? Mais ce petit zéro nous aide beaucoup, comme on verra à tout à l'heure.

Dans le système arabe, chaque chiffre ne peut être qu'une des dix valeurs-ci. Mais on peut utiliser plusieurs chiffres à la fois. Chaque chiffre prend aussi une valeur de sa **place**. Donc, si je vois 275, je sais qu'il y a deux **cent**, soixante-quinze. Le « 2 » ici ne joue pas le même que le « 2 » simple ! Ici nous voyons que le chiffre « 2 » veut dire toujours « deux »... mais ici il se trouve à la place des « cent ». Ce qui donne « deux cent » !

Et voilà maintenant la puissance de zéro. Le zéro nous permet de vider une place. C'est-à-dire le zéro fait la différence entre « deux » (2), « vingt » (20), « deux cent » (200), et « deux mille » (2000).

Nous somme très familiers avec ce système. En générale, on voit qu'un nombre composés des chiffres comme ceci :

2519

veut dire

2	5	1	9
1000	100	10	1

c'est-à-dire deux mille, cinq cent, dix, neuf.

Dans le langage mathématiques, on dirais qu'un nombre

$a_3a_2a_1a_0$

correspond à :

$a_3 \times 10^3 + a_2 \times 10^2 + a_1 \times 10^1 + a_0 \times 10^0$

ou également

$a_3 \times 1000 + a_2 \times 100 + a_1 \times 10 + a_0 + 1$

Nous voyons que chaque place a sa valeur, qui joue avec la valeur du chiffre. Les valeurs de places montent chaque fois en se multipliant par 10 – qui est exactement le nombre de chiffres qu'il y a dans le système ! C'est pour cela qu'on appelle ce système **décimal** ou **base dix**.

Base dix nous semble le plus compréhensible parce que c'est notre habitude. Peut-être c'est parce que nous avons dix doigts qu'on utilise un système base dix.

Comprenons bien comment ce système marche ! C'est à partir de système qu'on va comprendre les autres systèmes de numération.

Révision

- Si j'ajoute cent a 3250, j'aurais combien ? Si j'ajoute dix ? Si j'ajoute un ?
- Si je prends le numéro 1776 et je diminue jusqu'à 1676, j'ai diminué par combien ? Si je diminue jusqu'à 1766 ? Si je diminue à 776 ?
- Quel est le nombre qui suit 179 ? Pourquoi ? le nombre qui suit 199 ? Pourquoi ? le nombre qui suit 999 ? Pourquoi ?

3 Binaire

L'ordinateur n'a pas de doigts. Pour compter, il ne peut qu'utiliser l'électricité. Considérons un interrupteur. On peut l'allumer, ou l'éteindre. Pour l'ordinateur, il n'y a que deux chiffres : 0, qui veut dire « éteint », et 1, qui veut dire « allumé ». Celui qui a créé l'ordinateur était donc obligé de trouver un système par lequel l'ordinateur peut traiter les nombres !

Le système que l'ordinateur utilise s'appelle « binaire » ou « base 2 » (« bi- » est une racine qui veut dire « deux », comme il n'y a que deux chiffres). Dans ce système, les valeurs des **places** sont différentes. Au lieu d'être les places 1, 10, 100, 1000 (les puissances de dix), ils sont les places de 1, 2, 4, 8 (les puissances de deux).

Voilà compter en binaire :

0	zéro
1	un
10	deux
11	trois
100	quatre
101	cinq

110	six
111	sept
1000	huit
1001	neuf
1010	dix
1011	onze
1100	douze
1101	treize
1110	quatorze
1111	quinze

Cela peut sembler un peu bizarre, mais le binaire suit les mêmes principes que le décimal. Chaque chiffre ne peut que être 0 ou 1, et après on a fini les chiffres, il faut ajouter un chiffre, qui donne 10.

Regardons un nombre comme 1101 (binaire).

1	1	0	1

Les valeurs des places sont :

1	1	0	1
8	4	2	1

Alors, ce numéro égale 1 x 8 + 1 x 4 + 0 x 2 + 1 x 1 = 8 + 4 + 0 + 1 = 13.

Pour distinguer 1000 binaire (huit) et 1000 décimal (mille) on va écrire un petit chiffre 2 pour les nombres binaires, et un petit 10 pour les nombres décimaux. Donc, $1000_2 = 8_{10}$.

Nous constatons que les puissances des deux se succèdent en multipliant par deux – alors, après 8 sera 16, 32, 64, etc.

Donc le nombre 101011_2 peut se traduire en décimal :

1	0	1	0	1	1
32	16	8	4	2	1

1 x 32 + 0 x 16 + 1 x 8 + 0 x 4 + 1 x 2 + 1 x 1 = 32 + 0 + 8 + 0 + 1 + 1 = 43_{10}.

Révision

- Traduisez en décimal : 10000_2, 00110_2, 111010_2, 1111111_2.
- Traduisez 1001_2 et 10010_2 en décimal. Quelle est leur relation? Pourquoi ?
- Quel est le nombre qui suit 1111111_2 ? Pourquoi ?

Maintenant on comprend un peu traduire les nombres binaires en décimal. Mais comment faire l'opération inverse ?

Sachons qu'un numéro doit convenir aux places données la-dessus. C'est-à-dire, on doit avoir un nombre des 32aines, de 16aines, et ainsi de suite. Alors, si on vois le numéro 14, on se demande : comment former 14 avec les chiffres 8, 4, 2, et 1 ?

$14 = 8 + 4 + 2$. Alors, $14_{10} = 1110_2$, parce que

1	1	1	0
8	4	2	1

donne toujours 14.

Par le même principe, s'il me faut mettre le numéro 76_{10} en binaire :

64	32	16	8	4	2	1

J'arrange d'abord mes places avec leurs valeurs. Je me demande, quelles dois-je ? Je vois que c'est un nombre grand, alors je vais mettre 64. J'avais à mettre 76. Si je mets 64, il me manque 12.

1						
64	32	16	8	4	2	1

Quelles places peuvent faire 12 ? Je vois que 32 et 16 sont trop grand, mais je vais mettre 8.

1	0	0	1			
64	32	16	8	4	2	1

Je voulais mettre 12, mais je n'ai que mettre 8. Il me manque toujours 4. Alors, je mets 4.

1	0	0	1	1	0	0
64	32	16	8	4	2	1

Et voilà : $76_{10} = 1001100_2$.

La méthode ici n'est pas difficile. La seule chose qu'il faut rappeler ici est que je prends toujours le nombre le plus grand. Si je n'avais pas pris 64, il me sera impossible de faire 76.

Révision

- Convertir les nombres 17_{10}, 45_{10}, 69_{10}, et 100_{10} en binaire. Vérifiez que vos réponses sont correctes en retraduisant en décimal.
- Si je ne prends pas 64 mais seulement les chiffres plus petits de 64, quel est le plus grand nombre que je peux faire avec les chiffres plus petits ? Comment ?
- Si un nombre est impair, on connaît quoi sur sa forme binaire ?

3.1 Moyenne alternative.

S'il ne me plaît pas de deviner les chiffres binaires, il existe une autre moyenne pour convertir les chiffres en binaire. Ceci utilise une série des divisions.

Exemple : regardons 14_{10}. Divisons par deux.

14 : 2 = 7 reste 0.

Comme ce nombre est pair, il n'est composé de chiffres pairs, donc la place pour 1 sera 0. Divisons encore par deux.

7 : 2 = 3 reste 1.

En divisant par deux et encore par deux, on a vraiment divisé par quatre. Le reste ici signifie que 2 doit être sur 14. Divisons encore par 2, et encore.

3 : 2 = 1 reste 1.

1 : 2 = 0 reste 1.

Quand on regarde les restes, nous voyons qu'ils sont : 0 1 1 1. Revenons sur 14_{10}, en haut, et nous voyons que en binaire, $14_{10} = 1110_2$. Alors, les restes, si on prend de bas en haut nous donne les chiffres pour sa forme binaire.

Essayons avec un autre exemple : 29_{10}.

29 : 2 = 14 reste 1.

14 : 2 = 7 reste 0.

7 : 2 = 3 reste 1.

3 : 2 = 1 reste 1.

1 : 2 = 0 reste 1.

Donc, $29_{10} = 11101_2$. On peut vérifier facilement :

1	1	1	0	1
16	8	4	2	1

16 x 1 + 8 x 1 + 4 x 1 + 2 x 0 + 1 x 1 = 16 + 8 + 4 + 0 + 1 = 29_{10}.

Révision

- Convertir les nombres 17_{10}, 45_{10}, 69_{10}, et 100_{10} en binaire avec cette méthode. Vérifiez vos réponses en comparant contre vos réponses de la dernière section.

3.2 Nombres fractionnaires en binaire

On voit qu'en décimal, parfois il arrive certains nombres qui sont entre deux nombres. Par exemple, s'il y a 7 cerises, et il me faut partager parmi deux amis, combien est-ce que chacun va recevoir ? On dit 7 : 2 = 3,5. Le « ,5 » (virgule cinq) est une manière de dire que chacun doit recevoir un « demi ». Chacun va recevoir plus que 3, mais personne ne doit avoir 4.

En décimal, on utilise la virgule pour dire, « ces chiffres auront une valeur très petite ». En fait, les places après la virgule ont leurs valeurs, exactement comme les autres places. Voilà :

1	2	,	4	6
10	1		0,1	0,01
10^1	10^0		10^{-1}	10^{-2}

En passant à gauche, on voit que les valeurs des places s'augmentent – chacune est dix fois l'ancienne. Et en passant à droite, les valeurs se diminuent – chacune est l'ancienne **divisée par dix**.

En binaire, c'est toujours la même chose :

1	0	,	1	0	1
2	1		0,5	0,25	0,125
2^1	2^0		2^{-1}	2^{-2}	2^{-3}

Donc, $10{,}101_2 = 1 \times 2 + 0 \times 1 + 1 \times 0{,}5 + 0 \times 0{,}25 + 1 \times 0{,}125 = 2 + 0 + 0{,}5 + 0 + 0{,}125 = 2{,}625_{10}$.

On peut également convertir les nombres fractionnaires décimaux en binaire, soit en prenant les valeurs, ou soit par une série de **multiplications** par 2. Au lieu de prendre les restes, comme on a fait pour les divisions, on prend les parties entières.

Exemple : $0{,}8125_{10}$.

0,8125 x 2 = 1,625. On prend 1.

0,625 x 2 = 1,25. On prend 1.

0.25 x 2 = 0,5. On prend 0.

0,5 x 2 = 1,0. On prend 1.

Les restes, 1101, donne $0{,}1101_2$. Donc, $0{,}8125_{10} = 0{,}1101_2$.

Donc, si on voit un nombre fractionnaire, on peut le convertir en transformant d'abord la partie entière, et puis en transformant la partie fractionnaire.

Révision

- Convertir en décimal. $0{,}01_2$; $0{,}1011_2$; $0{,}00001_2$; $1101{,}01_2$.
- Convertir les nombres $0{,}5_{10}$, $0{,}0625_{10}$, et $5{,}375_{10}$ en binaire avec n'importe quelle méthode. Vérifiez vos réponses en traduisant encore en décimal.

4 Opérations

L'ordinateur est chargé de gérer les informations et les numéros, et il le fait en traduisant en binaire. Mais l'ordinateur doit pouvoir aussi **calculer** avec ces informations. Comment fait-il ?

Il arrive que les opérations en binaire sont presque mêmes qu'en décimal, mais **simplifié** parce qu'il y a moins de possibilités.

4.1 Addition

En décimal, on fait :

$$\begin{array}{rcccc} & 4 & 3 & 8 \\ + & 5 & 8 & 1 \\ \hline 1 & 0 & 1 & 9 \end{array}$$

On regarde les colonnes de chiffres de droit à gauche. 8 + 1 = 9. 3 + 8 = 11, mais on ne peut pas mettre 11 dans une colonne ! Plutôt on met 1, et on apporte 1 à la prochaine colonne. Donc on aura 1 + 4 + 5 = 10. On met 0, et on apporte 1 dans la dernière colonne, où il n'y a rien, donc on peut mettre simplement en bas.

En binaire, c'est presque même :

$$\begin{array}{rccccc} & 1 & 1 & 0 & 1 & 0 \\ + & & 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 1 & 0 \end{array}$$

Nous procédons de droit à gauche. 0 + 0 = 0 (bien sûr). 1 + 0 = 1, et 0 + 1 = 1 aussi. Dans la quatrième colonne nous avons un problème. 1 + 1 = 2, mais un binaire il n'y a pas de chiffre pour écrire 2. Plutôt on met 10. Mais on ne peut pas mettre 10 dans une colonne ! On met 0, et on apporte le 1 à la prochaine colonne. Donc ici on aura 1 + 1 + 1, qui est 3; mais en binaire on écrit 11. On met 1 et on apporte 1 dans la dernière colonne, où il n'y a rien, donc on peut mettre simplement en bas. Donc 11010_2 + 11100_2 = 110110_2.

Révision

- Vérifiez le résultat de cette addition en traduisant les numéros en décimal avant de faire l'addition. Est-ce que ça donne la même chose ?
- Faites : 11010_2 + 111_2, 100011_2 + 10011_2, 1111111_2 + 1_2. Vérifiez vos réponses.

4.2 Soustraction

En décimal, on fait :

$$\begin{array}{rcccc} & 8 & 0 & 1 & 8 \\ - & 1 & 4 & 8 & 3 \\ \hline & 6 & 5 & 3 & 5 \end{array}$$

On commence toujours à droit. On voit 8 - 3 = 5. Puis on a 1 - 8, mais on ne peut pas soustraire 8 de 1 parce que 8 est trop grand. Donc on emprunte de la colonne à gauche. Mais là il n'y a que 0, et on ne peut pas emprunter ce qu'il n'y a pas. Donc on emprunte de 8 (il reste 7) pour donner 10 à 0. On emprunte 1 de 10 (il reste 9) pour donner 10 à 1. Maintenant on a 11 - 3 = 8, et 9 (qui restait) - 4 = 5, et finalement 7 (qui restait) - 1 = 6.

```
    7  9  11
    8  0  1  8
-   1  4  8  3
-------------
    6  5  3  5
```

En binaire, on fait à peu près la même chose :

```
    1  0  0  0  1  1
-        1  1  1  0
-------------------
       1  0  1  0  1
```

On commence toujours à droit. On voit 1 - 0 = 1, et 1 - 0 = 0. Puis on a 0 - 1, mais on ne peut pas soustraire 1 de 0 parce que 1 est trop grand. Donc on emprunte de la colonne à gauche. Mais là il n'y a que 0, et on ne peut pas emprunter ce qu'il n'y a pas. Donc on emprunte encore à gauche, mais c'est toujours 0 ! On continue jusqu'à arriver à 1. On emprunte pour mettre 10 à droite, mais là il faut encore emprunter (10 - 1 = 1 reste). On emprunte et emprunte et emprunte. Finalement on aura :

```
    0  1  1  10
    1  0  0   0  1  1
-        1   1  1  0
-------------------
       1  0   1  0  1
```

10 - 1 = 1, 1 - 1 = 0, et 1 - 0 = 1. Donc 100011_2 - 1110_2 = 10101_2.

Révision

- Vérifiez le résultat de cette soustraction en traduisant les numéros en décimal avant de faire la soustraction. Est-ce que ça donne la même chose ?
- Faites : 100010_2 - 10011_2, 11010_2 - 111_2, 1000000_2 - 1_2. Vérifiez vos réponses.

4.3 Multiplication

En décimal :

```
       1  4  3  2
    x     3  7  1
-------------------
       1  4  3  2
 1  0  0  2  4
 4  2  9  6
-------------------
 5  3  1  2  7  2
```

D'abord, on regarde 1432 x 1. On peut le faire chiffre par chiffre : 2 x 1 = 2, 3 x 1 = 3, 4 x 1 = 4, et 1 x 1 = 1. On écrit cette série en bas. Puis, on regarde 1432 x 7, et on fait la même chose, mais à une place à

gauche. 2 x 7 = 14, on écrit 4 et garde 1. 3 x 7 = 21, on ajoute le 1 qu'on a gardé, qui donne 22; on écrit 2 et on garde 2. 4 x 7 = 28, plus le 2 qu'on a gardé, qui donne 30; on écrit 0 et on garde 3. 1 x 7 = 7 plus le 3 qu'on a gardé, qui donne 10. En somme, ça donne 10024. On fait la même chose avec 3, qui donne 4296, mais deux places à gauche. Finalement on additionne toutes les trois choses, qui donnent 531272. Donc, 1432_{10} x 371_{10} = 531272_{10}.

En binaire, c'est même plus facile, parce qu'il n'y a pas besoin de « garder » des chiffres, ni de mémoriser un grand tableau de multiplication.

```
        1 0 1 1
      x   1 1 0
-------------------
        0 0 0 0
      1 0 1 1
    1 0 1 1
-------------------
1 0 0 0 0 1 0
```

D'abord, on regarde 1011 x 0. Ça donne tout simplement 0. Puis, on regarde 1011 x 1, qui donne 1011; on écrit une place à gauche. Encore, on regarde 1011 x 1, qui donne encore 1011, qu'on écrit deux places à gauche. Finalement on les somme, qui donne 1000010. Donc, 1011_2 x 110_2 = 1000010_2.

Révision

- Vérifiez le résultat de cette multiplication en traduisant les numéros en décimal avant de faire la multiplication. Est-ce que ça donne la même chose ?
- Faites : 11010_2 x 11_2, 11010_2 x 101_2, 1101_2 x 10000_2. Vérifiez vos réponses.

4.4 Division

En décimal, la division se fait :

```
  6 2 7 | 3
- 6 ↓ ↓ | 2 0 9
-------  -----
  0 2 ↓
  - 0 ↓
    2 7
  - 2 7
-------
      0
```

On regarde le chiffre à droit. On trouve le nombre de fois que le diviseur peut se trouver là. on voit que pour le nombre 6, 3 peut se mettre 2 fois. Donc on met 2 sous 3 et on écrit le résultat de leur multiplication (3 x 2 = 6, alors on soustrait 6). Puis on descend le prochain chiffre, qui est 2. 3 ne peut pas se mettre en 2, donc on met 0 fois. 3 x 0 = 0, donc on soustrait 0. Puis on descend le prochain

chiffre, qui est 7. Ça fait 27. 3 peut se mettre 9 fois là, qui donne 27, donc on soustrait 27. Il reste 0, et il n'y a plus de chiffres, donc on a fini : $627_{10} : 3_{10} = 209_{10}$.

En binaire, c'est même plus facile, parce que les chiffres du quotient ne peut que être 0 ou 1. Donc :

```
 1  0  0  0  0  0  1  |  1  1  0  1
 -  1  1  0  1  ↓  ↓  |  1  0  1
 ---------------------
       0  1  1  0  ↓
    -  0  0  0  0  ↓
    ---------------
       0  1  1  0  1
       -  1  1  0  1
       ---------------
                   0
```

On commence en regardant les premiers chiffres. 1101 ne peut pas entrer dans 1000, alors on augmente encore un chiffre. 1101 ne peut que être dans 10000 une fois, donc on met 1 en bas de 1101 et on soustrait 1101 x 1 = 1101. 10000-1101 donne 11. Puis on descend le prochain chiffre 0, que donne 110. 1101 ne peut pas entrer dans 110, donc on met 0 et on soustraire 1101 x 0 = 0000. Puis on descend le prochain chiffre 1, qui donne 1101. 1101 peut entrer dans 1101 une seule fois, donc on met 1 et on soustrait 1101 x 1 = 1101. Il reste 0 et il n'y a plus de chiffres, donc on a fini. $1000001_2 : 1101_2 = 101_2$.

Remarque : N'oubliez pas de mettre 0 s'il faut descendre encore un chiffre !

Révision

- Vérifiez le résultat de cette division en traduisant les numéros en décimal avant de faire la division. Est-ce que ça donne la même chose ?
- Faites : $100011_2 : 101_2$, $110010_2 : 1010_2$, $1101000_2 : 100_2$. Vérifiez vos réponses.

5 Bit, octet, baud

5.1 Le bit

L'ordinateur utilise dans sa mémoire une série de **transistors**, qui fonctionnent comme les interrupteurs qu'on a vus à tout à l'heure. Pour enregistrer un numéro, il faut beaucoup de transistors. Plus grand le numéro, plus de transistors. Pour discuter la taille d'un numéro, on parle du nombre de **chiffres** qu'il a. Ces chiffres sont les chiffres du numéro quand il est converti en binaire. Donc, on regarde le nombre de **chiffres binaire**.

Le numéro $14_{10} = 1110_2$. On dit que ce numéro à quatre chiffres binaires, ou qu'il faut quatre chiffres binaires pour enregistrer ce numéro.

Un chiffre binaire s'appelle aussi **bit**. Ce mot vient de l'anglais **binary digit**, c'est à dire **chiffre binaire**. Le numéro 1110_2 compte 4 bits. Le numéro 11110_2 compte 5 bits.

Si je regarde les possibilités pour un bit, je vois qu'il y a deux : ce bit peut être 0, ou 1.

Si je regarde les possibilités pour deux bits, je vois qu'il y a quatre : ces bits peuvent être 00, 01, 10, ou 11.

Si je regarde les possibilités pour trois bits, je vois qu'il y a **huit**. Pourquoi? Parce que, pour chaque possibilité avec deux bits, je peux prendre cette possibilité avec 0, ou avec 1. Alors, les possibilités sont 000, 001, 010, 011, 100, 101, 110, et 111. Ces nombres correspondent aux nombres binaires de 0 à 8_{10}.

Révision

- Combien de bits y a-t-il dans le nombre 1110100_2 ? 11101010111_2 ? 17_{10} ? 45_{10} ? 52_{10} ?
- Combien de possibilités y a-t-il avec quatre bits ? Ils correspondent auxquels nombres ? Quel est le plus grand nombre qu'on peut faire avec ?
- Combien de possibilités y a-t-il avec huit bits ? Quel est le plus grand nombre qu'on peut faire avec ?

5.2 Octets

On voit que, pour **x** bits, il y a 2^x possibilités, et leurs valeurs sont de 0 jusqu'à 2^x - 1.

Les bits sont souvent regroupés en groupe de 8, ce qu'on appelle un octet. Donc, 8 bits font un octet, et un octet fait 8 bits. Un octet est l'unité de la taille de l'information numérique. En anglais on dit aussi « byte » (qui se prononce comme le mot anglais « bite », avec la même voyelle que le français « bail »).

Un octet, étant petit, se mesure avec le même système de préfixes que le Système Internationale, qui est utilisé dans la science. C'est à dire :

1 kilooctet = 1000 octets = 1000 octets = 10^3 octets

1 mégaoctet = 1000 kilooctets = 1000000 octets = 10^6 octets

1 gigaoctet = 1000 mégaoctets = 1000000000 octets = 10^9 octets

1 téraoctet = 1000 gigaoctets = 1000000000000 octets = 10^{12} octets

1 pétaoctet = 1000 téraoctets = 1000000000000000 octets = 10^{15} octets

Cependant, en informatique il est traditionnel d'utiliser un système de puissances de 2 et non de 10. (C'est plus facile pour l'ordinateur si on divise par les puissances de 2.) Donc, certains disent que 1 kilooctet = 1024 octets, 1 mégaoctet = 1024 kilooctets = 1048576 octets, etc. Pour être strictement correct, on devrait appeler ces quantités avec les préfixes kibioctets, mébioctets, et ainsi de suite, avec « bi » pour « binaire ». Néanmoins c'est rare de voir ces unités, et certains utilisent « kilooctets » pour dire « 1024 octets ». Alors, faites attention pour savoir si votre professeur préfère les systèmes de 1000 ou de 1024 !

C'est pour cela que si on achète un disque dur 300 Go (= 300 000 000 000 octets), l'ordinateur dit que c'est 279.4 Go (gibioctets, proprement écrit Gio, 1 Gio = 1 073 741 824 octets).

Un octet peut facilement enregistrer une lettre (voir 7. ASCII). Pour enregistrer les autres qualités d'information, il faut beaucoup plus de bits. Une image de haute résolution peut être vers 1 ou 2 mégaoctets. Une chanson compressée (un fichier MP3) peut être vers 1 ou 2 mégaoctets par minute. Une vidéo haute qualité peut être 10 ou 20 mégaoctets par minute !

On garde ces informations dans les périphériques de stockage. Un CD peut contenir jusqu'à 700 Mo. Un DVD 4,7 Go. Aujourd'hui on sort les clés USB de 1, 2, 4, 8, et 16 Go.

- Combien de bits y a-t-il dans 4 octets ? Quel est donc le nombre le plus grand qu'on peut enregistrer avec 4 octets ?
- J'achète une clé USB de 4043284480 octets. Combien de gigaoctets a-t-elle ? Combien de gibioctets ?
- J'ai un fichier musique de 4,8 mébioctets. Combien d'octets fait-elle ? J'ai un disque dur de 120 Mo. Combien de telles musiques peux-je y enregistrer ?
- J'ai une clé USB de 32 Mo. Est-ce que c'est grand ? Pourquoi ?

5.3 Transfert de données

Dans un réseau, surtout un grand réseau comme Internet, on transfère les informations. Le transfert d'information est limité par ce qu'on appelle la **bande passante** – c'est la vitesse, ou **débit**, d'un lien réseau. La bande passante définit la quantité **maximum** d'information transmissible dans une unité de temps. S'il y a interférence, la bande passante ne doit pas être remplie.

Une unité de B.P. doit avoir unité d'information par unité temps, par exemple, kilooctets par seconde. On utilise souvent les unités de base **bits par second** (**bps**), s'écrit aussi baud.

Voilà la bande passante pour certaines qualités de connexion.

Connexion	B.P.
clé Camtel	167 kbps
ADSL	512 kbps
réseau Ethernet	1000 mbps

La plus élevée la bande passante, le plus vite le transfert de fichiers.

Travailler avec les vitesses-ci peut être difficile parce que les fichiers se mesurent en **octets** mais les bandes passantes en **bits**. Faites attention comment convertir – rappelons-nous que 8 bits = 1 octet.

Exemple.

J'ai un fichier musique de 4 Mo, et je veux l'envoyer à mon ami à travers une clé Camtel de BP 167 kbps.

Le fichier est 4 Mo = 4000 ko. (Chaque Mo = 1000 ko.)

4000 ko = 32000 kb. (Chaque octet = 8 bits.)

32000 kb : 167 kbps = 191.616766467 sec, ou arrondi 192 secondes.

60 secs donne une minute, donc 192 secondes. donne 192 : 60 = 3.2 minutes.

Il y a plusieurs manières de faire ce genre de calcul. Demandez à un prof de physique de vous montrer comment faire les calculs avec les unités pour une autre moyenne.

Rappelez-vous que D = Q/T (débit = taille divisé par temps), Q=DxT (taille = débit fois temps), et T=Q/D (temps = taille divisé par débit).

Révision

- J'ai un fichier de 2,7 Mo à télécharger avec ma connexion ADSL. Il me faudra combien de temps pour le faire ?
- Je copie des fichiers sur ma carte mémoire. Je transfère 150 Mo en 6 minutes. Quel est le débit de ce transfert, en kbps ?
- Je veux graver un CD avec une vitesse de 150 kbps. Combien de mégaoctets peux-je graver en 10 minutes ?

6 Hexadécimal

Le système de numération binaire est bien facile pour l'ordinateur, mais on voit que l'être humain souffre un peu à cause des nombres très longs. Un nombre de trois chiffres comme 300_{10} devient neuf chiffres en binaire ! Comment mieux gérer ces nombres ?

Pour cela, on a développé un système de numération qui s'appelle hexadécimal, ou « base 16 » (« hexa- » est une racine qui veut dire « six », et « déc- » est une racine qui veut dire « dix », comme dix + six = seize).

Dans base 16, il y a 16 chiffres. On a 0 jusqu'à 9, comme d'habitude, mais on a aussi six autres. Par convention, on les appelle : A, B, C, D, E, et F. Ceci peut être un peu bizarre – comment les lettres peuvent être les chiffres ? – mais on a besoin des chiffres, et pourquoi créer les nouveaux si on peut utiliser les lettres ?

Voilà compter en hexadécimal :

0	zéro	0	0
1	un	1	1
2	deux	2	10
3	trois	3	11
4	quatre	4	100
5	cinq	5	101
6	six	6	110
7	sept	7	111
8	huit	8	1000
9	neuf	9	1001
a	dix	10	1010
b	onze	11	1011
c	douze	12	1100
d	treize	13	1101

e	quatorze	14	1110
f	quinze	15	1111
10	seize	16	10000
11	dix-sept	17	10001
12	dix-huit	18	10010
1f	trente et un	31	11111
20	trente deux	32	100000
9f	cent cinquante-neuf	159	10011111
a0	cent soixante	160	10100000
a1	cent soixante et un	161	10100001
a9	cent soixante-neuf	169	10101001
aa	cent soixante-dix	170	10101010
af	cent soixante-quinze	175	10101111
b0	cent soixante-seize	176	10110000
bf	cent quatre-vingt-onze	191	10111111
ff	deux cent cinquante cinq	255	11111111
100	deux cent cinquante six	256	100000000

Ce tableau montre certains nombres en hexadécimal et aussi en décimal et binaire. Nous voyons que les chiffres a, b, c, d, e, et f fonctionnent presque comme les autres chiffres plus familiers à nous. On compte jusqu'à ne plus avoir de chiffres, et puis on utilise deux chiffres. On compte jusqu'à ne plus avoir de chiffres, et puis on utilise trois chiffres. Et ainsi de suite.

Convertir **de** hexadécimal et comme convertir de binaire. D'abord, on cherche les valeurs des places :

2	a	7
256	16	1

Nous voyons que les valeurs des places sont données par les puissances de 16. Avec ces valeurs, on fait un calcul familier: 2 x 256 + a x 16 + 7 x 1 = 512 + 160 + 7 = 679_{10}. (Rappelons que a suit 9 et a la valeur 10.)

Convertir **en** hexadécimal et comme convertir en binaire. Il y a toujours deux moyennes. La première est de mettre les places vides et deviner les chiffres qu'il faudra, et l'autre est de faire plusieurs divisions (par 16). Faisons les deux avec le nombre 199_{10}.

256	16	1

Je vois que 256 est trop grand pour mettre même 1. Mais combien de 16 peut-on avoir? 16 x 10 = 160. 16 x 11 = 176. 16 x 12 = 192. 16 x 13 = 208. Donc, 208 est trop grand. Il faut 12 fois. Pour écrire 12 en base seize, on utilise le chiffre c. On aura:

0	c	
256	16	1

On a déjà trouvé 192, mais on avait a trouvé 199. On voit qu'il reste 7. Pour compter 7, on met simplement 7. Donc:

0	c	7
256	16	1

$199_{10} = 0c7_{16} = c7_{16}$. (On met un petit 16 pour dire « base 16 ».)

Également, on peut diviser plusieurs fois par 16.

199 : 16 = 12 reste 7.

12 : 16 = 0 reste 12.

On utilise encore les restes inversé. 12 est toujours c. Donc, on aura $c7_{16}$.

Révision

- Convertir en décimal : 40_{16}, $7d_{16}$, dd_{16}.
- Convertir en hexadécimal : 50_{10}, 99_{10}, 221_{10}.
- Apprenez à utiliser la calculatrice pour convertir entre décimal et hexadécimal, et utilisez-la pour vérifier vos réponses.
- Quel est le plus grand nombre qu'on peut avoir avec trois chiffres hexadécimaux ? Quatre chiffres ?

6.1 Relation entre hexadécimal et binaire.

On a dit que hexadécimal est utilisé parce qu'il réduit le nombre de chiffres qu'on devrait utiliser en binaire. On voit déjà que hexadécimal utilise moins de chiffres que décimal. Cependant, convertir semble beaucoup complexe. Comment faire ?

Bien, on a bien fait de choisir hexadécimal, parce qu'il y a une relation spéciale entre hexadécimal et binaire, et cette relation fait que convertir entre les deux est très facile.

Pour comprendre cette relation, regardez bien le tableau au-dessus. Nous voyons que jusqu'à quatre chiffres binaires sont toujours traduisibles par un chiffre hexadécimal. Avec deux chiffres hexadécimaux, on peut aller jusqu'à huit chiffres binaires. Et ainsi de suite. On voit que **chaque groupe de quatre chiffres binaires vaut à un chiffre hexadécimal.**

Voyons un exemple. Je veux traduire 158_{16} en binaire. Je vais le faire en passant par décimal.

1	5	8
256	16	1

$158_{16} = 1 \times 256 + 5 \times 16 + 8 \times 1 = 256 + 80 + 8 = 344_{10}$. Maintenant, on convertit en binaire :

2048	1024	512	256	128	64	32	16	8	4	2	1

Je vois qu'il faut 256. Avec 256 il reste 344 - 256 = 88. Pour 88 il faut 64 et il reste 24. Pour 24 il faut 16 + 8.

			1	0	1	0	1	1	0	0	0
2048	1024	512	256	128	64	32	16	8	4	2	1

Donc $158_{16} = 344_{10} = 101011000_2$. Arrangeons un peu :

base 16	1	5	8
base 2	0001	0101	1000

Nous voyons qu'on peut prendre un raccourci. Au lieu de passer par décimal, qui est pénible, on peut traduire chiffre par chiffre, directement ! $1_{16} = 0001_2$, $5_{16} = 0101_2$, $8_{16} = 1000_2$.

Un autre exemple : traduire $cd7_{16}$ en binaire.

$c_{16} = 12_{10}$, et pour cela il faut 8 + 4. Donc $c_{16} = 1100_2$.

$d_{16} = 13_{10}$, et pour cela il faut 8+4+1. Donc $d_{16} = 1101_2$.

$7_{16} = 7_{10}$, et pour cela il faut 4+2+1. Donc $7_{16} = 0111_2$.

En joignant, on voit : $cd7_{16} = 110011010111_2$.

Également, si on veut traduire de binaire en hexadécimal, on peut le faire facilement en prenant les groupes de quatre chiffres, et les transformant un par un. Exemple : 101100101001_2.

1011	0010	1001
8+2+1	2	8+1
11	2	9
b	2	9

$= b29_{16}$.

S'il n'y a pas suffisamment de chiffres binaires qu'on divise en quatre, on ajoute les chiffres 0 à gauche. Exemple: 100011111_2 a neuf chiffres, donc il faut ajouter trois chiffres à gauche pour qu'il soit douze: 000100011111_2. Maintenant on divise en groupes de quatre chiffres, et on voit 0001, 0001, et 1111, qui donnent 1, 1, et f. Donc $100011111_2 = 11f_{16}$.

Nous voyons que un octet compte 8 bits, qui veut dire que la valeur d'un octet peut s'écrire en deux chiffres hexadécimaux.

Révision

- Convertir de hexadécimal en binaire, en passant par décimal : $1d_{16}$, $5f2_{16}$, 842_{16}. Convertir aussi avec la technique qu'on a vu dans ce chapitre. Quelle moyenne préférez-vous ?
- Convertir de binaire en hexadécimal, en passant par décimal : 10011111_2, 101111001111_2, 1110011001_2. Convertir aussi avec la technique qu'on a vue dans ce chapitre. Quelle moyenne rendez-vous ?
- Marthe dit qu'elle a fait la conversion de $2f8_{16}$ en binaire en passant par décimal. Elle a vu qu'il y avait trois chiffres hexadécimaux, et comme chaque chiffre hexadécimal donne quatre chiffres binaires, elle s'est attendu 12 chiffres binaires. Mais à la fin, elle n'a que eu 10. Faites la conversion et expliquer qu'est qui s'est passé.

7 ASCII

On a vu que l'ordinateur utilise les chiffres binaires pour garder et manipuler les numéros. Mais l'ordinateur doit aussi pouvoir manipuler le texte. Comment fait-il ?

À la base, l'ordinateur est une machine **numérique**. Ceci veut dire que l'ordinateur peut seulement comprendre les nombres. Chaque chose qu'on veut traiter avec l'ordinateur doit être d'abord traduite en nombres. On dit qu'on a **codé** les informations. Développer ces codes s'appelle **codification**.

Pour le texte, il y a un code qui permet l'ordinateur de traduire les lettres en nombres, et en manipuler. On appelle ce code ASCII (American Standard Code for Information Interchange). Ce code donne un nombre pour chaque lettre ou caractère qu'on veut enregistrer dans l'ordinateur. Ce code compte 128 lettres différentes. Voila le code complet :

Déc	Hex	Char	Déc	Hex	Char
0	00	NUL '\0'	64	40	@
1	01	SOH (start of heading)	65	41	A
2	02	STX (start of text)	66	42	B
3	03	ETX (end of text)	67	43	C
4	04	EOT (end of transmission)	68	44	D
5	05	ENQ (enquiry)	69	45	E
6	06	ACK (acknowledge)	70	46	F
7	07	BEL '\a' (bell)	71	47	G
8	08	BS '\b' (backspace)	72	48	H
9	09	HT '\t' (horizontal tab)	73	49	I

Déc	Hex	Char	Déc	Hex	Char
10	0A	LF '\n' (new line)	74	4A	J
11	0B	VT '\v' (vertical tab)	75	4B	K
12	0C	FF '\f' (form feed)	76	4C	L
13	0D	CR '\r' (carriage ret)	77	4D	M
14	0E	SO (shift out)	78	4E	N
15	0F	SI (shift in)	79	4F	O
16	10	DLE (data link escape)	80	50	P
17	11	DC1 (device control 1)	81	51	Q
18	12	DC2 (device control 2)	82	52	R
19	13	DC3 (device control 3)	83	53	S
20	14	DC4 (device control 4)	84	54	T
21	15	NAK (negative ack.)	85	55	U
22	16	SYN (synchronous idle)	86	56	V
23	17	ETB (end of trans. blk)	87	57	W
24	18	CAN (cancel)	88	58	X
25	19	EM (end of medium)	89	59	Y
26	1A	SUB (substitute)	90	5A	Z
27	1B	ESC (escape)	91	5B	[
28	1C	FS (file separator)	92	5C	\
29	1D	GS (group separator)	93	5D	]
30	1E	RS (record separator)	94	5E	^
31	1F	US (unit separator)	95	5F	_
32	20	SPACE	96	60	`
33	21	!	97	61	a
34	22	"	98	62	b
35	23	#	99	63	c
36	24	$	100	64	d
37	25	%	101	65	e

Déc	Hex	Char	Déc	Hex	Char
38	26	&	102	66	f
39	27	´	103	67	g
40	28	(	104	68	h
41	29	)	105	69	i
42	2A	*	106	6A	j
43	2B	+	107	6B	k
44	2C	,	108	6C	l
45	2D	-	109	6D	m
46	2E	.	110	6E	n
47	2F	/	111	6F	o
48	30	0	112	70	p
49	31	1	113	71	q
50	32	2	114	72	r
51	33	3	115	73	s
52	34	4	116	74	t
53	35	5	117	75	u
54	36	6	118	76	v
55	37	7	119	77	w
56	38	8	120	78	x
57	39	9	121	79	y
58	3A	:	122	7A	z
59	3B	;	123	7B	{
60	3C	<	124	7C	|
61	3D	=	125	7D	}
62	3E	>	126	7E	~
63	3F	?	127	7F	DEL

Nous voyons que chaque nombre de 0 jusqu'à 127 peut se représenter avec 7 bits, ou deux chiffres hexadécimaux. Donc, chaque lettre a un code ASCII qui est compris de deux chiffres hexadécimaux. Exemple, la lettre « A » (majuscule) correspond au code 41_{16}. Ces codes ont aussi les valeurs

décimales, donc « A » correspond aussi à 65_{10}. Ça veut dire que une phrase peut se coder en ASCII soit avec les valeurs en hexadécimal ou soit en décimal (mais hexadécimal est beaucoup plus souvent utilisé).

Nous voyons aussi que les lettres minuscules ont les codes un peu différents que les lettres majuscules – « a » minuscule est 61_{16}, ou 97_{10}. En effet, la valeur de chaque lettre minuscule est la valeur de sa forme majuscule, plus 20_{16} (en hexadécimal) ou 32_{10} (en décimal). Constatons aussi que l'espace a aussi un code – 20_{16} (32_{10}). Chaque chose qu'il faut que l'ordinateur enregistre doit pouvoir se coder.

Alors, quand on tape du texte sur le clavier, l'ordinateur ne comprend pas les lettres. Il comprend plutôt les numéros des lettres selon ce code. Quand je tape « Salut ! », l'ordinateur voit les chiffres (en hexadécimal) : 53 61 6c 75 74 20 21.

En fait, il y a plusieurs codes; ASCII a remplacé un code plus ancien qui s'appelle EBCDIC, et on peut dire que ASCII a été remplacé par le code ISO 8859-1, qui contient toutes les lettres ASCII et ajoute aussi les autres, par exemple les lettres avec accents. De plus en plus on utilise Unicode, qui a pour but le codification de chaque symbole écrite dans toutes les langues humaines.

Révision

- Coder le texte en ASCII, en hexadécimal : Bonjour M. Ethan
- Coder ce texte aussi en utilisant les codes décimaux.
- Décodez ce texte ASCII : 56 6f 69 63 69 20 6d 6f 6e 20 54 45 58 54 45.
- Décodez ce texte ASCII : 74 101 32 115 117 105 115 32 105 110 102 111 114 109 97 116 105 99 105 101 110.
- Allez au cybercafé et cherchez un peu le code de texte qu'on appelle ISO 8859-1. Utilisez ce code pour coder le texte : Allé à l'hôpital

8 Octal

Le dernier système de numération que nous allons voir est le système octal, ou base 8 (« oct- » veut dire huit). Dans ce système, il n'y a que 8 chiffres, qui sont de 0 à 7.

Exemple: 541_8

5	4	1
64	8	1

$541_8 = 5 \times 64 + 4 \times 8 + 1 \times 1 = 353_{10}$.

Et également, convertir est bien comme en hexadécimal ou décimal.

Exemple : 106_{10}

64	8	1

Combien de 64 ? 1 x 64 = 64, mais 2 x 64 = 128. Donc il faut un 64. Il reste 106 - 64 = 42. Combien de 8 ? 5 x 8 = 42, donc il faut cinq 8. Il reste 42 - 40 = 2. Il faut deux 1.

1	5	2
64	8	1

Donc $106_{10} = 152_8$.

Ou, avec les divisions :

106 : 8 = 13 reste 2. 13 : 8 = 1 reste 5. 1 : 8 = 0 reste 1.

Donc, $106_{10} = 152_8$.

Révision

- Convertir en décimal : 64_8, 721_8, 1000_8.
- Convertir en octal : 69_{10}, 91_{10}, 520_{10}.

Comme hexadécimal, octal est un système développe pour avoir une relation avec binaire. Mais alors que hexadécimal prend quatre bits pour chaque chiffre, en octal ce n'est que trois bits.

Donc, pour traduire d'octal en binaire, on prend chaque chiffre et traduis en trois bits.

Exemple : convertir 572_8 en binaire.

$5_8 = 101_2$.

$7_8 = 111_2$.

$2_8 = 010_2$.

Donc, $572_8 = 101111010_2$.

Et au contraire : convertissons 10110111_2 en octal.

On divise en groupes de trois, en ajoutant un 0 au début pour avoir neuf chiffres.

$010_2 = 2_8$.

$110_2 = 6_8$.

$111_2 = 7_8$.

Donc, $10110111_2 = 267_8$.

Révision

- Convertir en binaire : 152_8, 7421_8, 512213_8.
- Convertir en octal : 101111_2, 1011011_2, 111110110110_2.

9 Résumé

Nous avons vu tous les systèmes de numération qu'on utilise en informatique, et comment on les applique pour faire les choses que l'ordinateur fait : les opérations mathématiques, les débits de transferts, et coder du texte avec ASCII. L'idée fondamentale est qu'un nombre est une chose, mais

comment écrire ce nombre est une chose différente, et il peut avoir plusieurs manières pour écrire un seul nombre. Chaque manière peut soutenir certaines opérations, et bloquer d'autres.

Remarques :

Les élèves confondent parfois comment convertir de décimal et comment convertir en décimal. Quand on convertit **en** décimal, il faut cherchez les valeurs des places existantes. Quant on convertit **de** décimal, il faut créer plutôt les places avec les valeurs – les valeurs des places décimales ne sont pas très utiles. Faites attention à cette différence !

Même si vous n'avez pas du tableau du code ASCII, rappelez-vous toujours que la lettre A majuscule est $41_{16} = 65_{10}$, et que les autres lettres suivent ($B = 42_{16}$, $C = 43_{16}$..), et que les minuscules sont les majuscules + 20_{16}. Avec ça on peut facilement récréer le tableau. N'oubliez jamais les espaces (code 20_{16}).

Les tableaux de ce livre sont venus du « man-page » intitulé « ascii(7) ». Voila un autre tableau des codes ASCII, résumé :

	2	**3**	**4**	**5**	**6**	**7**
0:		0	@	P	`	p
1:	!	1	A	Q	a	q
2:	"	2	B	R	b	r
3:	#	3	C	S	c	s
4:	$	4	D	T	d	t
5:	%	5	E	U	e	u
6:	&	6	F	V	f	v
7:	´	7	G	W	g	w
8:	(	8	H	X	h	x
9:	)	9	I	Y	i	y
A:	*	:	J	Z	j	z
B:	+	;	K	[	k	{
C:	,	<	L	\	l	
D:	-	=	M	]	m	}
E:	.	>	N	^	n	~
F:	/	?	O	_	o	DEL

Nous voyons que chaque carré du tableau a une valeur donne en hexa par l'en-tête de colonne et le rang. Par exemple, « A » se situe dans la colonne de 4 et le rang 1, donc il a la valeur 41_{16}.