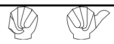
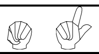
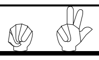
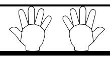
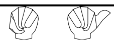
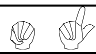
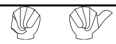
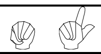
LA NUMEROTATION BINAIRE

Ce chapitre explique la manière utilisée en informatique pour représenter des chiffres dans les fils électriques. Il est indispensable pour qui veut comprendre le fonctionnement de la partie matérielle d'un ordinateur : la numérotation binaire est au micro-électronicien ce que la clé de 12 est au mécano, le micro au chanteur, la plume à l'écrivain, bref, ce que le sens de la chute était à Pierre Desproges…

COMPTONS SUR NOS DOIGTS

Nous avons dix doigts

Vous avez sûrement déjà compté sur vos doigts le nombre de personnes invitées à une fête. Le problème survient lorsqu'on dépasse le nombre de doigts disponibles : 10, pour la plupart d'entre nous. La tentation est grande de repartir du premier doigt, mais rien ne nous indique si c'est la première fois ou pas qu'on revient à ce premier doigt, d'où risque d'erreur.

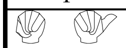
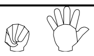
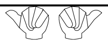
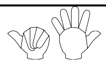
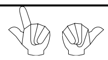
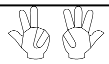
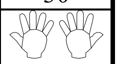
1	2	3	…	10	11	12	…	21	22	…
			…				…			…

Nos amis gauchers devront se soumettre, une fois de plus, à un petit exercice d'adaptation, qu'ils m'en pardonnent.

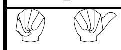
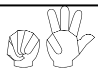
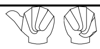
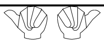
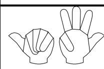
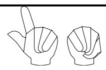
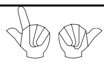
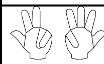
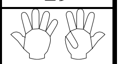
Les cas 1 et 2 sont identiques respectivement aux cas 11 et 12, puis aux cas 21 et 22. Ce système de comptage n'est donc pas fiable.

Mais nous avons aussi deux mains de cinq doigts

Une autre solution consiste à ne compter les personnes que sur la main droite, et ne se servir de la main gauche que pour compter le nombre de fois où on est revenus au premier doigt de la main droite. Cela donne :

1	…	5	6	…	10	11	…	19	…	30
										

On observera, en faisant l'exercice avec ses propres mains (et aussi ses mains propres) que lorsqu'on sort un doigt de la main gauche, on rentre en même temps les doigts de la main droite. La position "n doigts de la main gauche sortis, et 5 doigts de la main droite sortis" et la position "n+1 doigts de la main gauche sortis, et aucun doigt de la main droite sorti" sont donc parfaitement équivalentes, la "valeur" d'un doigt de la main gauche correspondant exactement à 5 doigts de la main droite. Cela revient à dire que la main gauche ne compte pas le nombre de fois où on est revenus au premier doigt (de la main droite), mais le nombre de fois où la main droite était entièrement ouverte. Le tableau ci-dessus devient donc :

1	…	4	5	6	…	9	10	11	…	19	…	29
												

Pourquoi seulement 29 ? Remarquez que :

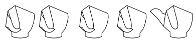
- dans le premier cas, le pouce de la main droite est toujours levé : on commence donc à compter à 1.
- dans le deuxième cas, l'auriculaire de la main droite n'est jamais levé : en effet, dès qu'on cherche à le lever, on lève irrésistiblement un doigt supplémentaire de la main gauche en refermant la main droite. En contrepartie, il arrive que cette dernière soit complètement fermée : on a inventé le zéro. Ce principe ne nous interdit donc pas la combinaison . On compte alors de 0 à 29, ce qui correspond bien à 30 cas.

Imaginons que nous ayons cinq mains de deux doigts

Nous allons appliquer le même principe que dans le cas précédent : chaque main compte le nombre de fois où la main immédiatement à droite est entièrement ouverte. On referme alors cette dernière.

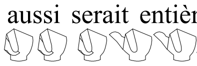
Partons de zéro :

Comptons un :



Si on ajoute 1, on devrait lever l'index de la main la plus à droite. Mais alors, vu qu'elle serait entièrement ouverte, on pratique comme précédemment : on lève le pouce de la main à sa gauche, et on peut refermer la main la plus à droite. Cela donne :

Ajoutons 1 : la main la plus à droite étant fermée, on peut lever son pouce. Cela donne :

Ajoutons 1 : on devrait lever l'index de la main la plus à droite. Mais alors, vu qu'elle serait entièrement ouverte, on devrait lever l'index de la main à sa gauche. Mais alors, vu que celle-ci aussi serait entièrement ouverte, on lève le pouce de la main à sa gauche. Ce dernier prenant à lui seul la valeur +1, on referme alors les deux mains de droite (comme lorsque le compteur kilométrique de la voiture passe de 099999 à 100000 km). Cela donne :

Et ainsi de suite… Comme dans le cas de deux mains à cinq doigts, on se rend compte que l'index ne sert finalement à rien (le saviez-vous seulement ?). Seule la position du pouce apporte de l'information. On notera alors par un 0 un pouce baissé, par un 1 un pouce levé. Dans notre cas de cinq mains à deux doigts, le comptage s'effectue alors comme suit :

0	00000
1	00001
2	00010
3	00011
4	00100
5	00101
6	00110
7	00111
8	01000
9	01001
10	01010
11	01011
12	01100
13	01101
14	01110
15	01111
16	10000
17	10001
18	10010
19	10011
20	10100
21	10101
22	10110
23	10111
24	11000
25	11001
26	11010
27	11011
28	11100
29	11101
30	11110
31	11111

Pour aller au-delà de 31, il suffit de rajouter un sixième chiffre à gauche, et ainsi de suite…

Le comptage présenté ici constitue le fameux système de *numérotation binaire*. Son intérêt particulier est qu'il est facilement transposable en grandeurs électriques : il suffit d'affecter un fil à chaque chiffre d'un nombre, et de définir par convention que le chiffre est 1 correspond à la présence d’une tension dans le fil définie arbitrairement (tension notée Vcc : cc pour *Courant Continu*), 0 correspond à une tension nulle (0 V).

GENERALISATION

Dans le cas de deux mains à cinq doigts, nous avons vu qu'un doigt de la main gauche avait la même valeur que 5 doigts de la main droite. Si nous avions eu une troisième main, un doigt de celle-ci aurait eu la même valeur que 5 doigts de la main gauche, soit que 5 × 5 = 25 doigts de la main droite.

De la même manière, dans le cas de cinq mains à deux doigts, un doigt de la deuxième main droite a la même valeur que 2 doigts de la main droite. Un doigt de la troisième main droite a aussi la même valeur que 2 doigts de la main à sa droite, soit 2 × 2 = 4 doigts de la main droite, et ainsi de suite. La valeur des doigts de chaque main est donc respectivement, en partant de la droite : 1, 2, 4, 8, 16, 32… (on double à chaque fois).

Le nombre de doigts de chaque main est appelé la *base* de comptage. Nous comptons communément en base 10, probablement parce que nous avons dix doigts. Dans le cas de deux mains à cinq doigts, nous avons compté en base 5, sur des nombres de 2 chiffres. Dans le cas de cinq mains à deux doigts, nous avons compté en base 2 (dite binaire), sur des nombres de 5 chiffres. De manière générale et d'après la théorie des nombres (non développable ici), la valeur d'un nombre entier de m+1 chiffres (noté $a_m a_{m-1} \ldots a_2 a_1 a_0$) est, en base N :

$$a_m N^m + a_{m-1} N^{m-1} + \ldots + a_2 N^2 + a_1 N^1 + a_0 N^0$$

Rappel : *N^i est dit "N puissance i", c'est-à-dire $N \times N \times N \times \ldots \times N$ (N apparaissant i fois).*
Par exemple, $10^4 = 10 \times 10 \times 10 \times 10 = 10\,000$, $2^5 = 2 \times 2 \times 2 \times 2 \times 2 = 32$.
Cas particuliers : quel que soit N, $N^1 = N$ et par convention $N^0 = 1$.

Les a_i sont des chiffres allant de 0 à N-1 (ce qui fait N chiffres). Par exemple, pour le nombre 2357 : a_3= 2, a_2= 3, a_1 = 5, a_0 = 7. Chaque N^i est appelé le *poids* du chiffre correspondant. Celui-ci dépendant de la position du chiffre dans l'écriture du nombre (2357 n'a pas la même valeur que 2537, bien qu'écrit avec les mêmes chiffres), cette dernière une base constitue un système de numérotation *positionnelle* (contrairement aux systèmes de numérotation *additionnelle*, qui se contentent d'additionner des valeurs indépendamment de leur position).

On peut légèrement simplifier :

$$a_m N^m + a_{m-1} N^{m-1} + \ldots + a_2 N^2 + a_1 N + a_0$$

Base 10 (pour repère)

En base dix, cela donne :

$$a_m \times 10^m + a_{m-1} \times 10^{m-1} + \ldots + a_2 \times 10^2 + a_1 \times 10 + a_0$$

Soit :

$$\ldots + a_5 \times 100\,000 + a_4 \times 10\,000 + a_3 \times 1\,000 + a_2 \times 100 + a_1 \times 10 + a_0$$

Avec les a_i étant des chiffres allant de 0 à 9.

Par exemple, 5684 n'est qu'une notation qui représente la valeur 5×1000 + 6×100 + 8×10 + 4. Nous sommes tellement habitués à la base 10 que nous en oublions la différence entre la notation (c'est-à-dire la représentation issue d'une convention d'écriture, propre à la base utilisée), et la valeur (indépendante de la base).

Base 2

En base deux, cela donne :

$$a_m \times 2^m + a_{m-1} \times 2^{m-1} + \ldots + a_2 \times 2^2 + a_1 \times 2 + a_0$$

Soit :

$$\ldots + a_5 \times 32 + a_4 \times 16 + a_3 \times 8 + a_2 \times 4 + a_1 \times 2 + a_0$$

Avec les a_i étant soit 0, soit 1. Chaque 0 ou 1 est appelé *bit*, à ne pas confondre avec le mot anglais *byte*, qui désigne un *octet*, c'est-à-dire un nombre de 8 bits (par exemple 10100110). Chaque bit est affecté à un poids respectif de 1, 2, 4, 8, 16, 32, 34, 128, etc… (à partir du chiffre le plus à droite). Par définition, ces poids sont les puissances successives de 2. Le bit le plus à gauche est dit *bit de poids fort*, celui le plus à droite est dit *bit de poids faible*. Les bits d'un octet sont souvent numérotés, depuis le bit0 (le plus à droite) jusqu'au bit7 (le plus à gauche). Ainsi, dans le nombre binaire 10100000, les bits 5 et 7 sont à 1, les autres à 0.

CONVERSIONS

Binaire → décimal

La formule ci-dessus donne directement la valeur décimale d'un nombre binaire. En pratique, on somme les poids des chiffres à 1. Par exemple, pour le nombre binaire 10100110 :

Binaire :	1	0	1	0	0	1	1	0
Rappel des poids	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*
Poids pris en compte	128		32			4	2	
Somme :	128+32+4+2 = 166							

Donc $10100110_{\text{binaire}} = 166_{\text{décimal}}$

Nota : pour compter ses invités à une fête, on remarquera qu'avec dix doigts, on a dix bits (doigt levé = 1, doigt baissé = 0). On peut donc compter depuis 0000000000 jusqu'à 1111111111, soit jusqu'à 1023 invités. Il suffit d'avoir beaucoup d'amis, et de bien assouplir ses articulations. En outre, nos amis menuisiers devront avoir moins d'invités…

Nota : un octet permet de compter de 00000000 à 11111111, soit de 0 à 255.
en jumelant deux octets (comptage 16 bits), on compte de 0 (00000000 00000000) jusqu'à 65 535 (11111111 11111111). L'octet de gauche a alors dans son ensemble un poids de 256 (valeur totale = octet de gauche × 256 + octet de droite).
en jumelant quatre octets (comptage 32 bits), on compte de 0 à 4 294 967 295.

Décimal → binaire

Enfilons maintenant une blouse de boucher (n'ayant pas encore investi dans une balance électronique). A l'aide d'une balance à 2 plateaux et de poids, nous devons peser un gigot posé sur un des deux plateaux :

- on pose le plus gros poids sur l'autre plateau : si ce dernier ne se baisse pas, on laisse le poids, sinon on l'enlève.
- puis on pose le poids immédiatement inférieur : si ce dernier ne se baisse pas, on laisse le poids, sinon on l'enlève.
- et ainsi de suite avec tous les poids… jusqu'à l'équilibre.

En fin de manipulation, le poids du gigot est la somme des poids placés sur l'autre plateau.

Nous procèderons de la même manière que le boucher, en nous posant la question : quelles sont les puissances de deux qui "pèsent" dans notre nombre ? En effet, ces puissances de deux correspondent aux chiffres 1 de sa notation binaire. Si on trouve une combinaison qui répond à la question, c'est la bonne, car elle est unique (précision pour les seuls puristes : la théorie des nombres indique qu'un chiffre décimal a deux représentations, mais l'une des deux est infiniment longue – par exemple 25 = 24,999999999999…-, et nous l'éliminons car nous travaillons sur les représentations finies). Il faut partir de la plus grosse puissance de 2 inférieure au nombre, car si elle est supérieure, elle ne peut forcément pas être comprise dans notre nombre.

Par exemple, pour 166 :

$2^8 = 256$ ne peut pas être compris dans 166. Prenons alors la puissance précédente : $2^7 = 128$ est comprise dans 166. Nous savons donc deux choses :

- notre nombre a 8 chiffres binaires (car le bit de poids 2^7 occupe la 8ème place en partant de la droite)
- c'est une lapalissade en notation binaire : le premier bit est un 1.

On peut donc présenter 166 sous la forme suivante :

Décimal	166 = 128 + A							
	128	A						
Binaire	1	?	?	?	?	?	?	?
Rappel des poids	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*

Donc A = 166-128 = 38. Nous allons donc appliquer le même raisonnement à A : la plus grosse puissance de 2 comprise dans A est $2^5 = 32$. Donc, le bit de poids 32 est 1 (les précédents étant 0 : ici, $2^6 = 64$ n'est pas compris dans 38, donc le bit correspondant est à 0). La représentation se précise :

Décimal	166 = 128 + A							
	128	A = 38 = 32 + B						
	128		32	B				
Binaire	1	0	1	?	?	?	?	?
Rappel des poids	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*

Par le même raisonnement sur B, on obtient successivement :

Décimal	166 = 128 + A							
	128	A = 38 = 32 + B						
	128		32	B = 6 = 4 + C				
	128		32			4	C	
Binaire	1	0	1	0	0	1	?	?
Rappel des poids	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*

Décimal	166 = 128 + A							
	128	A = 38 = 32 + B						
	128		32	B = 6 = 4 + C				
	128		32			4	C = 2 + D	
	128		32			4	2	D=0
Binaire	1	0	1	0	0	1	1	0
Rappel des poids	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*

On retrouve bien $166_{\text{décimal}} = 10100110_{\text{binaire}}$, ce qui ne manque pas de nous rassurer.

On remarquera que la valeur des poids du boucher ne sont pas quelconques : si le premier poids fait 1 kg, le second pèse 500g (soit la moitié du précédent), le troisième 250g (soit encore la moitié du précédent), puis 125g, 63g, etc… (le « etc » me permettant de passer sous silence les approximations décimales de la suite). Comme monsieur Jourdain, le boucher compte en binaire sans le savoir. Ironie du sort : les balances électroniques les plus avancées, qui traitent l'information sous forme binaire, ne font également que compter des poids, comme nous venons de le faire lors de notre conversion, et comme le faisait le boucher…

LA NUMEROTATION HEXADECIMALE

La notation hexadécimale consiste à compter non pas en base 2 ou 10, mais en base 16 ! Cela nécessite d'avoir 16 chiffres (dans le sens littéral : symbole de représentation d'une valeur numérique). On peut déjà utiliser les chiffres usuels de 0 à 9, représentant leurs valeurs habituelles. Il en reste donc 6 à trouver : on aurait pu utiliser par exemple 🕭 pour la valeur 10, ✂ pour la valeur 11, 🖂 pour la valeur 12, ⌛ pour la valeur 13, ⌘ pour la valeur 14 et ✹ pour la valeur 15. Le nombre hexadécimal 1🖂3✹ aurait eu pour valeur :

$$1\times16^3 + \text{🖂}\times16^2 + 3\times16 + \text{✹} = 1\times16^3 + 12\times16^2 + 3\times16 + 15 = 7231_{\text{décimal}}$$

Il a été décidé d'utiliser des symboles plus familiers : les six premières lettres de l'alphabet, en majuscule. Les chiffres utilisés et leurs valeurs respectives sont donc les suivants :

Chiffre hexadécimal	Valeur en décimal	Valeur en binaire
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Le nombre 1⌧3✸ précédent s'écrit en fait : 1C3F.

La notation hexadécimale revêt deux aspects intéressants :

- 16 étant une puissance de deux (en l'occurrence 2^4), un chiffre hexadécimal correspond exactement à quatre chiffres binaires (et inversement). Ainsi, les conversions entre les bases 16 et 2 sont directes et réciproques. Par exemple, pour le nombre binaire 10100110 :

Binaire	1010 0110		
	1010	0110	Conversion directe pour chaque chiffre hexadécimal
Hexadécimal	A	6	
	A 6		

De la même manière, 11000110111010101 = 0001 1000 1101 1101 0101
= 1 8 D D 5
= 18DD5

- De nombreuses données informatiques se présentent sous la forme d'octets. Un octet étant un nombre binaire à huit chiffres (00000000 à 11111111), c'est aussi un nombre hexadécimal à deux chiffres, d'après ce que nous venons de voir (00 à FF). La notation hexadécimale permet donc de représenter la plupart des données informatiques sous une forme à la fois compacte et invariante de deux chiffres hexadécimaux.

Pour information, les bases 4 et 8 (2^2 et 2^3) présentent la même caractéristique de conversion avec le binaire, mais permettent une compression d'écriture moindre que l'hexadécimal. De plus, la base 8 (dite octale) est inadaptée à l'écriture d'un octet (elle groupe les bits par 3, or il y 8 bits à écrire).

NOTATION

Enfin, et afin d'éviter tout risque d'erreur entre bases, il est recommandé d'écrire un "b" à la fin d'un nombre binaire, un "d" à la fin d'un nombre décimal, et un "h" à la fin d'un nombre hexadécimal (toujours en minuscules, afin de ne pas confondre avec les "chiffres" hexadécimaux B et D). Par exemple :

- 10100110b = 166d = A6h

Mais aussi (et la vérification constituera un bon exercice) :

- 10100110d = 100110100001110110001110b = 9A1D8Eh
- 10100110h = 10000000100000000000100010000b = 269484304d

D'où l'intérêt de cette notation, lorsqu'on jongle entre les bases !

LE BINAIRE REFLECHI (OU CODE DE GRAY)

Définition

Le codage binaire décrit ci-dessus présente un inconvénient : pour passer d'une valeur à la suivante, plusieurs bits peuvent changer de valeur en même temps. Par exemple, pour passer de 31d à 32d en binaire, tous les bits changent de valeur : de 11111b à 100000b. Or, il est fréquent d'utiliser dans les circuits intégrés des suites de nombres qui se suivent. Cela entraîne alors de nombreux basculement de transistors, donc de l'échauffement (qui oblige, comme nous le verrons, à limiter la rapidité) et de la consommation de courant.

Un autre codage binaire a donc été mis au point afin de pallier cet inconvénient : le *binaire réfléchi*. Le but étant de ne changer qu'un bit en passant d'une valeur à la valeur immédiatement supérieure (par exemple de 4 à 5), et donc inférieure aussi, son principe de construction est le suivant (exemple sur 4 bits) :

- Passage de 0d à 1d : dans la notation binaire normale, seul le bit0 change. On conserve donc 0d = 0000b, 1d = 0001b.

Décimal	Binaire	Binaire réfléchi			
		bit3	bit2	bit1	bit0
0	0000	0	0	0	0
1	0001	0	0	0	1

- Pour les deux chiffres suivants (2d et 3d) : on inscrit 1 dans bit1. Pour bit0, on reprend les valeurs déjà prises dans les valeurs précédentes, mais à l'envers, comme si le trait gras était un miroir réfléchissant la colonne (d'où le terme binaire *réfléchi*).

Décimal	Binaire	Binaire réfléchi			
		bit3	bit2	bit1	bit0
0	0000	0	0	0	0
1	0001	0	0	0	1
2	0010	0	0	1	1
3	0011	0	0	1	0

- Pour les quatre chiffres suivants (de 4d à 7d) : on inscrit 1 dans bit2. Pour bit0 et bit1, on reprend les valeurs déjà prises dans les valeurs précédentes, mais à l'envers, de la même manière que précédemment.

Décimal	Binaire	Binaire réfléchi			
		bit3	bit2	bit1	bit0
0	0000	0	0	0	0
1	0001	0	0	0	1
2	0010	0	0	1	1
3	0011	0	0	1	0
4	0100	0	1	1	0
5	0101	0	1	1	1
6	0110	0	1	0	1
7	0111	0	1	0	0

- Pour les huit chiffres suivants (de 8d à 15d) : on inscrit 1 dans bit3. Pour bit0, bit1 et bit2, on reprend les valeurs déjà prises dans les valeurs précédentes, mais à l'envers, de la même manière que précédemment.

Décimal	Binaire	Binaire réfléchi			
		bit3	bit2	bit1	bit0
0	0000	0	0	0	0
1	0001	0	0	0	1
2	0010	0	0	1	1
3	0011	0	0	1	0
4	0100	0	1	1	0
5	0101	0	1	1	1
6	0110	0	1	0	1
7	0111	0	1	0	0
8	1000	1	1	0	0
9	1001	1	1	0	1
10	1010	1	1	1	1
11	1011	1	1	1	0
12	1100	1	0	1	0
13	1101	1	0	1	1
14	1110	1	0	0	1
15	1111	1	0	0	0

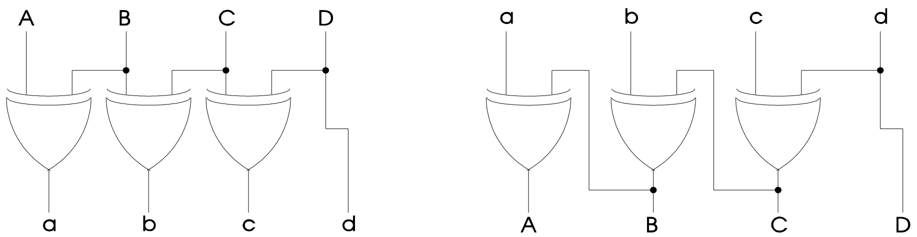
- et ainsi de suite pour aller au-delà de 15d, et même aussi loin que l'on veut… C'est un véritable acharnement.

Le concept de la réflexion repose sur deux idées :

- lorsqu'on passe le bit de gauche de 0 à 1, il n'y a plus qu'une solution pour n'avoir qu'un bit qui change de valeur : reprendre telles quelles les valeurs des autres bits du nombre précédent.
- puisqu'au-dessus du "miroir", il n'y a qu'un bit qui change à la fois, ce sera la même chose par réflexion en dessous du miroir (dans l'ordre inverse, mais peu importe).

Conversions

La conversion d'un nombre binaire noté ABCD en un nombre binaire réfléchi noté abcd, et inversement, se réalise comme suit (les montages sont extensibles au nombre de bits désirés) :



Remarque

Attention : le code de Gray n'est pas à proprement parler une base ! En effet, la valeur du nombre écrit ne se déduit pas par le calcul de :

$$a_mN^m + a_{m-1}N^{m-1} + \ldots + a_2N^2 + a_1N^1 + a_0N^0$$

Dans le même ordre d'idée, voici une petite question :

ATTENTION AU PIEGE (DIGRESSION)

Compte-t-on en base :

- 12, quand un anglais compte 1 shilling au lieu de 12 pence ?
- 24 lorsqu'on compte les heures de 0h à 23h ?

- 60 lorsqu'on compte les minutes de 0' à 59', les secondes de 0'' à 59'' ?

Comme le titre de ce sous-chapitre peut le laisser présager, la réponse est négative : en effet, lorsqu'on compte les heures :

- on utilise toujours bien 10 chiffres (de 0 à 9), et non 24. Or, le fondement d'une base N est d'utiliser N chiffres.
- lorsqu'on atteint la 24ème heure, on repart de 0, et non de 10.

On compterait les heures en base 24 si :

- on les comptait avec 24 chiffres, par exemple : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, I, J, K, L, M, N.
- on notait la 24ème heure : 10. Puis 11 pour la 25ème, 12 pour la 26ème, …1N pour la 47ème, 20 pour la 48ème…

On peut seulement dire que le décompte se fait « modulo 24 » pour les heures (ce qui signifie qu'on repart de 0 quand on atteint 24), « modulo 60 » pour les minutes et secondes, etc… mais toujours en base 10.

Cependant, l'utilisation quotidienne de bases autres que 10 est tout à fait possible. Certaines civilisations ne s'en sont pas privées :

- De nombreuses civilisations ont utilisé la base 4, issue du décompte des quatre doigts de la main sans le pouce. En a découlé une base 8, utilisant les deux mains. Coïncidence ? De nos jours, 16 (qui n'est jamais que 2×8) est en français le dernier nombre ayant sa propre dénomination (seize), les nombres suivants étant composés : dix-sept, dix-huit…
- Les celtes comptaient en base 12 (système *duodécimal*), en pointant les phalanges des quatre doigts d'une main avec son pouce. Ce système était loin d'être inefficace, car 12 est divisible par 2, 3, 4 et 6 (ce qui simplifie les calculs de division par 2, 3, 4 ou 6 et leurs multiples), alors que 10 n'est divisible que par 2 et 5. Il nous en reste les douzaines d'œufs, d'huîtres… ainsi que les dénomination propres des nombres jusqu'à douze dans d'autres langues (allemand = de « eins » à « zwölf », puis « dreizehn », « vierzehn »…, anglais = de « one » à « twelve », puis « thirteen », « fourteen »…) et de nombreuses quantités pour les anglais.
- Les mayas avaient leur base 20 (système *vicésimal*), avec les chiffres suivants :

0 (coquillage)	1 ●	2 ● ●	3 ● ● ●	4 ● ● ● ●
5 —	6 ● —	7 ● ● —	8 ● ● ● —	9 ● ● ● ● —
10 — —	11 ● — —	12 ● ● — —	13 ● ● ● — —	14 ● ● ● ● — —
15 — — —	16 ● — — —	17 ● ● — — —	18 ● ● ● — — —	19 ● ● ● ● — — —

Il est remarquable que même le dessin des chiffres est « construit », et non issu d'une simple convention comme nos 0, 1, 2, 3… A tel point qu'on pourrait <u>presque</u> dire qu'il y a trois chiffres : (le zéro, de valeur nulle), ● (l'unité) et ——— (de valeur 5 unités). Le « presque » se justifie par le fait que cette notation des chiffres n'est pas une notation positionnelle, mais additionnelle (la valeur du chiffre est déduite de la somme du poids des points et des barres, indépendamment de leur position qui n'est que convention), et ne constitue donc pas une base.

- Les chaldéens (de la Chaldée, nom donné à la région de Babylone) comptaient en base 60 (système *sexagésimal*) ! Pour eux, $568_{chaldéen} = 5×60^2 + 6×60 + 8 = 18\ 368_{décimal}$. Comme ceux des mayas, leurs chiffres étaient construits, <u>presque</u> comme une base, « ▼ » représentant l'unité et « < » dix unités (notation *cunéïforme* : représentation à l'aide de dessins de « clous »).

18 368$_{décimal}$ s'écrivait alors ▼▼▼ ▼▼ ▼▼▼ ▼▼▼ ▼▼▼▼ ▼▼▼▼

5112 $_{décimal}$ (= 1×60² + 25×60 + 12) s'écrivait alors ▼ <<▼▼▼ ▼▼ <▼▼

Dès qu'on change de base, on a tellement l'impression de perdre tout repère qu'il semble impossible d'appréhender la nouvelle base : estimer au simple jugé, sans calcul, ce que représente 568$_{chaldéen}$. Or, ce n'est en fait pas si déroutant que cela : pour se convaincre qu'il ne s'agit que d'une question d'habitude, il suffit de constater que 568$_{chaldéen}$ = 5×60² + 6×60 + 8, est comparable pour nous à 5 heures 6 minutes 8 secondes. On se rend alors parfaitement compte de l'ordre de grandeur que représentait 568$_{chaldéen}$ pour les chaldéens (5 heures 6 minutes 8 secondes par rapport à 1 seconde pour nous).
A partir de l'observation des astres, ils avaient évalué l'année (définie par une rotation complète de la terre autour du soleil) à 6 périodes de 60 jours, d'où découlent aujourd'hui les conventions suivantes :

- le degré est la 6×60ème = 360ème partie du cercle, c'est-à-dire l'angle décrit par la terre autour du soleil en 1 jour.
- le degré est divisé en 60 minutes, elles-mêmes constituées de 60 secondes.
- la détermination de la longitude d'après la position du soleil nécessitant une fine maîtrise du temps (qui passe), la transposition en minutes et secondes temporelles devenait alors naturelle.
- par ailleurs, on remarque que 60 a en français un statut particulier, puisqu'au delà on compte différemment : soixante-dix, quatre-vingts, quatre-vingt-dix.

Parmi les autres curiosités (mais n'obtenant pas le statut de « bases ») :

- Les grecs et les égyptiens utilisaient une unité, le *ell*, valant 6 mains de 4 doigts, origine du 24 que l'on retrouve dans nos modulos (les anciennes horloges italiennes avaient 24 heures), à moins que 24 ne provienne simplement de deux douzaines celtes (une pour le matin, une pour l'après-midi) ?
- Notre langage contemporain dénote la réminiscence d'une référence au nombre 20 dans le français médiéval, issue de la notation romaine (qui était tout sauf une base : comment poser le moindre calcul en notation romaine, par exemple « LXXII×XIV = MVIII », pas même plus pratique que de poser l'opération « soixante-douze×quatorze =mille-huit » ?). En effet, 80 était noté IIII$_{XX}$ (et par exemple 93 noté III$_{XX}$XIII), d'où, par exemple :
  - notre « quatre-vingts » au lieu d'« octante », comme on pourrait dire « quatre-dix » plutôt que « quarante ».
  - l'« Hôpital des XV-XX » (lire « quinze-vingts ») à Paris, issu de la « Méson des tras cens aveugles » fondée à la moitié du XIII^{ème} (les habitudes sont tenaces !) siècle par le roi Saint-Louis et destinée à héberger 300 (= 15 × 20) pensionnaires aveugles.
- Le Hindi (Inde) utilise la base 10, mais tous les nombres jusqu'à 100 ont un nom !
- L'Egypte ancienne utilisait une notation additionnelle, avec notamment I pour l'unité, ∩ pour les dizaines et 9 pour les centaines. Ainsi, 235 pouvait s'écrire 99∩∩∩IIIII, ou bien IIIII99∩∩∩, ou encore II9∩∩I9I∩I.

Preuve en est que, malgré les apparences et nos certitudes, si notre numérotation décimale est mathématiquement construite, il n'en va pas de même de sa retranscription littéraire, qui ne s'est pas tout à fait épurée du poids de l'histoire.

RECREATION (ENFIN…)

Voici un petit jeu avec lequel vous pourrez coller pas mal d'amis. Il est très simple à utiliser, bien moins à comprendre pour un non-initié. Vous, cher lecteur, disposez cependant pour cela d'un atout considérable : le fait de découvrir ce jeu dans un chapitre consacré à la numérotation binaire est une indication déterminante pour deviner son fonctionnement, aussi pourrez-vous être indulgents envers vos "victimes" si elles ne trouvent pas la clé de ce petit tour. C'est pourquoi je vais vous le dévoiler par étapes : vous verrez à partir de quel indice vous aurez compris son fonctionnement (c'est-à-dire qu'il faut comprendre comment retrouver le nombre, mais également comment les grilles sont construites et pourquoi).

1er indice

Ce premier indice est tout simplement la règle du jeu : vous et votre victime (puisque c'est bien de cela dont il s'agit) disposez des grilles ci-dessous. La victime doit choisir un nombre entre 1 et 100 compris, bien sûr sans vous le dévoiler. Elle doit par contre vous indiquer les grilles sur lesquelles il apparaît. Vous êtes alors capable, très rapidement, d'annoncer le nombre choisi.

Par exemple, elle a choisi le nombre 54, et vous annonce qu'elle le voit dans les 2ème, 3ème, 5ème et 6ème grilles, ce qui vous suffit pour lui annoncer sous ses yeux ébahis qu'elle pense au nombre 54.

Voici les grilles :

1	3	5	7	9	11	13	15	17	19
21	23	25	27	29	31	33	35	37	39
41	43	45	47	49	51	53	55	57	59
61	63	65	67	69	71	73	75	77	79
81	83	85	87	89	91	93	95	97	99

2	3	6	7	10	11	14	15	18	19
22	23	26	27	30	31	34	35	38	39
42	43	46	47	50	51	54	55	58	59
62	63	66	67	70	71	74	75	78	79
82	83	86	87	90	91	94	95	98	99

4	5	6	7	12	13	14	15	20	21
22	23	28	29	30	31	36	37	38	39
44	45	46	47	52	53	54	55	60	61
62	63	68	69	70	71	76	77	78	79
84	85	86	87	92	93	94	95	100	

8	9	10	11	12	13	14	15	24	25
26	27	28	29	30	31	40	41	42	43
44	45	46	47	56	57	58	59	60	61
62	63	72	73	74	75	76	77	78	79
88	89	90	91	92	93	94	95		

16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	48	49	50	51
52	53	54	55	56	57	58	59	60	61
62	63	80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95		

32	33	34	35	36	37	38	39	40	41
42	43	44	45	46	47	48	49	50	51
52	53	54	55	56	57	58	59	60	61
62	63	96	97	98	99	100			

64	65	66	67	68	69	70	71	72	73
74	75	76	77	78	79	80	81	82	83
84	85	86	87	88	89	90	91	92	93
94	95	96	97	98	99	100			

2ème indice

Observez les nombres en haut à gauche de chaque grille : ceux-ci ne vous rappellent-ils rien ?

3ème indice

Le nombre à deviner est calculé en faisant le total des nombres en haut à gauche des grilles indiquées par la victime. Par exemple, la victime a choisi le nombre 54, qui figure dans les 2ème, 3ème, 5ème et 6ème grilles, celles-ci ayant comme nombre en haut à gauche respectivement 2, 4, 16 et 32. Le nombre à rechercher est donc 2+4+16+32 = 54.

Néanmoins, cela n'explique pas encore comment sont construites ces grilles.

Solution

J'espère que ce paragraphe ne servira que de vérification d'une solution déjà trouvée !

Un nombre entre 0 et 100d s'écrit en binaire sur 7 bits. Chaque grille correspond à un de ces bits, et liste tous les chiffres entre 0 et 100d pour lesquels ce bit est à 1. Comme ils sont triés par ordre croissant, le premier (celui en haut à gauche de la grille) est le plus petit de la liste, donc celui ayant le bit considéré à 1 et tous les autres à 0 : il indique donc le poids du bit pris à 1 dans la grille.

Lorsque la victime indique les grilles où elle retrouve son nombre, elle indique en fait les bits à 1 dans celui-ci en numérotation binaire. Il suffit donc de sommer les poids de ces bits (soit le nombre en haut à gauche de la grille) pour reconstituer le nombre recherché.

Celui qui ne connaît pas la notation binaire aura bien du mal à trouver la clé du mystère. Or, peu de monde la connaît. Vous aurez également des difficultés à lui expliquer simplement. Aussi l'exercice suivant devrait vous y aider.

FINI DE JOUER !

Petit exercice

On commençait à se refroidir… : comment construire le jeu précédent, en utilisant la base 10 au lieu de 2 (pour des raisons de commodité, on se limitera aux nombre de 0 à 99 compris) ?

Solution

Chaque grille ne contient que 10 nombres, et regroupe les nombres ayant en commun un même chiffre à la même position (unité ou dizaine), ce qui fait 20 grilles : 10 pour les chiffres des unités, 10 pour ceux des dizaines (en fait, les deux grilles de poids 0 sont inutiles : elles ne participent pas au calcul) :

0	10	20	30	40	50	60	70	80	90
1	11	21	31	41	51	61	71	81	91
2	12	22	32	42	52	62	72	82	92
3	13	23	33	43	53	63	73	83	93
4	14	24	34	44	54	64	74	84	94
5	15	25	35	45	55	65	75	85	95
6	16	26	36	46	56	66	76	86	96
7	17	27	37	47	57	67	77	87	97
8	18	28	38	48	58	68	78	88	98
9	19	29	39	49	59	69	79	89	99
0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

RESUME

Tout nombre peut être représenté par une suite de 0 et de 1, en le convertissant en base 2, dite numérotation binaire.
En associant, dans un fil électrique, une tension de référence au chiffre 1, et tension nulle au chiffre 0, il est possible de coder n'importe quel nombre binaire par un groupement suffisant de fils électriques (un par chiffre).
La compréhension de la notation binaire est indispensable pour la suite.