Culture informatique

Cours n° 2 :

Les bases de numération

Syst. Décimal

Le système décimal

– Les chiffres.

Les nombres que nous utilisons habituellement sont ceux de la base 10 (système décimal).

Nous disposons de dix chiffres différents de 0 à 9 pour écrire tous les nombres. La représentation des nombres est une succession de ces chiffres, comme par exemple

342 381.

Syst. Décimal

– Numération de position

On sait que dans 342 381, le chiffre 4 ne vaut pas 4 mais 40 000 . . .

De même suivant sa position, le chiffre 3 correspond soit à 300 000, soit à 300.

Suivant sa position dans l'écriture d'un nombre, un même chiffre, correspond à des valeurs bien différentes. Dans les calculs, la disposition des chiffres, alignés à partir de la droite, permet de tenir compte des différentes valeurs associées à des positions.

Exemple
les chiffres 4, 3 et 4 de la même colonne (celle des centaines) correspondent à 400, 300, 400.

$$\begin{array}{r} 2456 \\ +\ 7321 \\ +\ \ \ 409 \\ \hline 10186 \end{array}$$

Rang d'un chiffre : le chiffre le plus à droite a toujours le rang 0

Syst. Décimal

Prenons le nombre 2348

$$2348 = 2 \times 1000 + 3 \times 100 + 4 \times 10 + 8 \times 1$$

$$= 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 8 \times 10^0$$

On peut retrouver le même résultat par divisions successives de 2348 par 10.

$$\begin{array}{l|l} 2348 & 10 \\ \hline \end{array}$$

$$\begin{array}{l|l} & 10 \\ \hline \end{array}$$

$$\begin{array}{l|l} & 10 \\ \hline \end{array}$$

– Résumé.

10 représente la base et les puissances (ici de 0 à 3) le rang (ou la position) de chaque chiffre.

Le système de numération décimal est le système dit de base dix, les chiffres utilisés sont au nombre de dix, ce sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Suivant sa position le chiffre x vaut $x.10^0$, $x.10^1$, $x.10^2$, . . .

Syst. Décimal

Exercice 1 :

Décomposer, sous formes de puissances de 10, les nombres suivants :

$2134_{10} =$

$42381_{10} =$

Syst. Décimal
Syst. Binaire

Le système binaire

Dans les domaines de l'automatisme, de l'électronique et de l'informatique, nous utilisons la base 2. Tous les nombres s'écrivent avec deux chiffres uniquement (0 et 1). De même que nous utilisons le système décimal parce que nous avons commencé à compter avec nos dix doigts, nous utilisons le binaire car les systèmes technologiques ont souvent deux états stables. (cf histoire)

- Un interrupteur est ouvert ou fermé
- Une diode est allumée ou éteinte
- Une tension est présente ou absente

A chaque état du système technologique, on associe un état logique binaire. La présence d'une tension sera par exemple notée 1 et l'absence 0.

Le chiffre binaire qui peut prendre ces deux états est nommé "Bit" (qui vient de **Bi**nary digi**t**).

Avec un bit nous pouvons coder deux états

0
1

Avec deux bits nous pouvons coder quatre états

0	0
0	1
1	0
1	1

Avec trois bits nous pouvons coder huit états

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

A chaque nouveau bit, le nombre de combinaisons possibles est doublé.

Ce nombre est égal à 2 puissance N (N étant le nombre de bits).

Les règles sont les mêmes que pour le décimal.

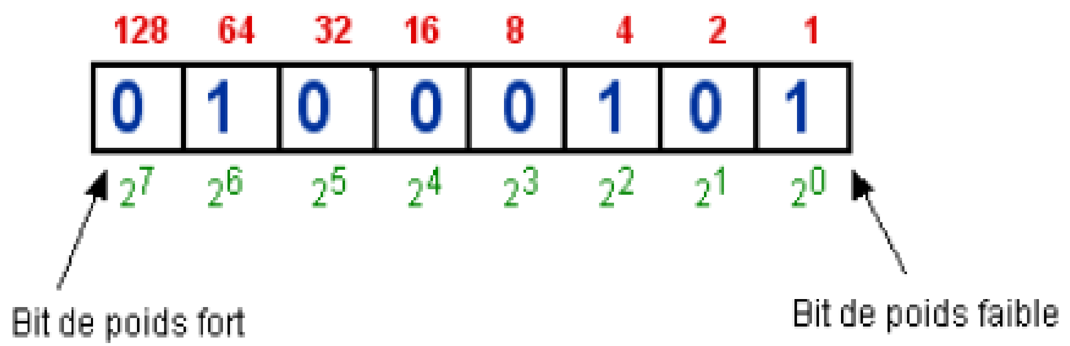
$$1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$= 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 11_{10}$$

Une suite de huit bits est nommé un octet (byte). Ex : 01000101

Une information codée sur 2 octets, soit 16 bits, est un mot (word).

1 octet :



- Le bit le plus à droite s'appelle le LSB (Less Significant Bit).
- Le bit le plus à gauche s'appelle le MSB (Most Significant Bit).

Avec un octet, nous pouvons écrire $2^8 = 256$ nombres binaires de 0 à 255.

Comparaison décimal - binaire

Base 10	Base 2
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
…	…

C'est déjà le dernier chiffre !

Remises à 0 en cascade !

C'est le nombre max avec 3 bits, le nombre suivant est 1000b = 2^3

En base B avec n chiffres, on peut compter de
0 à $B^n - 1$

Soit B^n valeurs différentes.

Un même nombre peut être représenté dans plusieurs bases

- 123 en base 10 (décimal)
- 1111011 en base 2 (binaire)

Il vaut mieux préciser la base pour éviter les ambiguïtés.

Notation en décimal :

123 ou 123_{10}

Notation en binaire :

1111011_{2} ou 1111011b ou #1111011 ou %1111011

Exercice 2 :

a) Donner la valeur du LSB et du MSB de l'octet : 01101111_2

b) Décomposer, sous formes de puissances de 2, les nombres binaires suivants :

$1010_2 =$

$11001_2 =$

c) Combien de bits comportent le nombre binaire : 10111010_2 ?

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire

Conversion de binaire à décimal

La méthode pour convertir un nombre binaire en un nombre décimal découle directement de la définition.

$$10011010_2 = 1\times2^7 + 0\times2^6 + 0\times2^5 + 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$$
$$= 128 + 16 + 8 + 2$$
$$= 154_{10}$$

Position du chiffre	8	7	6	5	4	3	2	1
Calcul	$2^{(8-1)}$	$2^{(7-1)}$	$2^{(6-1)}$	$2^{(5-1)}$	$2^{(4-1)}$	$2^{(3-1)}$	$2^{(2-1)}$	$2^{(1-1)}$
Valeur décimale	128	64	32	16	8	4	2	1
Opération	×	×	×	×	×	×	×	×
Valeur binaire à calculer	1	0	0	1	1	0	1	0
Valeur selon position	128	0	0	16	8	0	2	0

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire

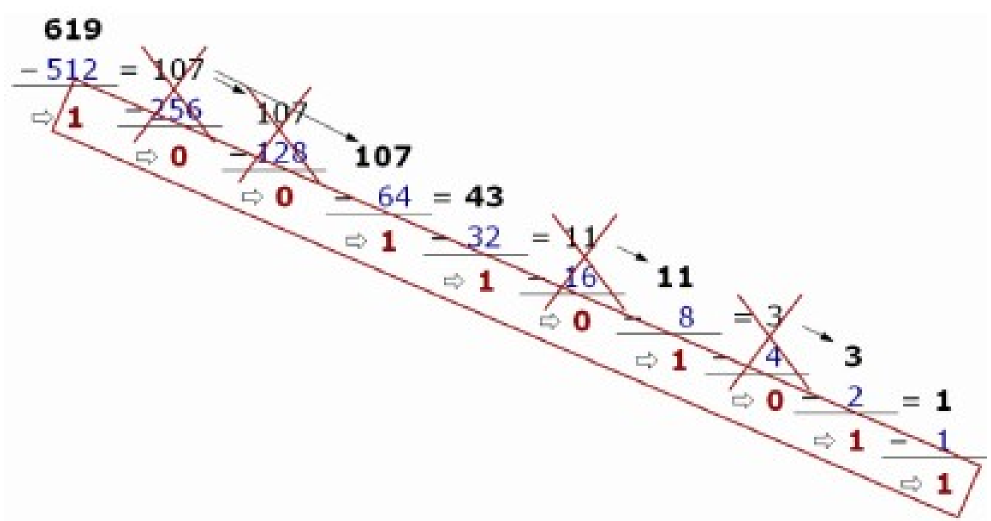
Conversion de décimal à binaire

Méthode 1 : par soustractions successives

n	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2^n	16383	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Exemple : convertir 619 en binaire

On soustrait la plus grande puissance de 2 possible. Si le résultat est négatif on note un 0 et on essaye de soustraire la puissance de 2 suivante. Si le résultat est positif on note 1, et on récupère le reste pour continuer avec les puissances de 2 suivantes.



Il suffit de lire la valeur binaire correspondante : 1001101011

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire

Méthode 2 : Méthode par divisions successives

Il faut réaliser des divisions successives par 2 et garder le reste.

Exemple : convertir 43 en binaire

Le premier reste donne le chiffre de poids faible.

Base

43 ÷ 2 = 21 reste 1

21 ÷ 2 = 10 reste 1

10 ÷ 2 = 5 reste 0

5 ÷ 2 = 2 reste 1

2 ÷ 2 = 1 reste 0

1 ÷ 2 = $\boxed{0}$ reste 1

le quotient = 0
la conversion
est finie

$\boxed{1\ 0\ 1\ 0\ 1\ 1}$

Il suffit de lire le résultat :
101011

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire

Exercice 3 :

a) Convertir les valeurs binaires en décimal :

0011 =

11001 =

10001001 =

b) Convertir les valeurs décimales en binaires :

57 =

193 =

255 =

chacun avec une méthode différente.

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa

Le système hexadécimal

La manipulation des nombres écrits en binaire est difficile pour l'être humain et la conversion en décimal n'est pas simple. C'est pourquoi nous utilisons de préférence le système hexadécimal (base 16).

Pour écrire les nombres en base 16 nous devons disposer de 16 chiffres, pour les dix premiers, nous utilisons les chiffres de la base 10, pour les suivant nous utiliserons des lettres de l'alphabet.

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Notation en hexadécimal :

$A3F_{16}$ ou A3Fh ou $A3F ou 0xA3F

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal

Ce système hexadécimal est aussi utilisé pour coder les couleurs (dans les pages HTML par exemple).

000000	0000FF	00FF00	FF0000	FFFF00	FFFFFF
noir	bleu	vert	rouge	jaune	blanc

Les règles sont ici aussi les mêmes que pour le décimal.

$$\begin{aligned} A3F_{16} &= A \times 16^2 + 3 \times 16^1 + F \times 16^0 \\ &= 10 \times 256 + 3 \times 16 + 15 \times 1 \\ &= 2560 + 48 + 15 \\ &= 2623_{10} \end{aligned}$$

Conversion hexadécimal vers décimal

Pour convertir un nombre hexadécimal en décimal, il suffit donc de se reporter à cette définition.

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa

Conversion décimal en hexadécimal

Les deux méthodes utilisées pour convertir le décimal en binaire s'appliquent également pour le décimal en hexadécimal.

Méthode 1 : par soustractions successives

On soustrait la plus grande puissance de 16 inférieure au nombre et on recommence avec le reste.

Exemple : 745 :

$16^2 = 256$ et $16^3 = 4096$ donc on garde 16^2.

$2 \times 16^2 = 512$ et $3 \times 16^2 = 768$, donc on a

$745 - 2\times16^2 = 233$.... on continue avec 233...

Du coup on peut écrire :

$745 = 2 \times 16^2 + 14 \times 16^1 + 9 \times 16^0$.

Ainsi $745_{10} = 2E9_{16}$

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa

Méthode 2 : par divisions successives

Reprenons le même exemple.

745	16
9	46

On divise le résultat, 46, encore par 16 jusqu'à obtenir une valeur inférieure à 16. 46 / 16 = 2 reste 14. Puis encore une fois, 2 / 16 = 0 reste 2. Il suffit de prendre les restes en ordre inverse. On retrouve bien notre résultat : $745_{10} = 2E9_{16}$

Exercice 4 :

a) Convertir les valeurs hexadécimal en décimal :

$10_{16} =$

$7_{16} =$

$234_{16} =$

b) Convertir les valeurs décimal en hexadécimal :

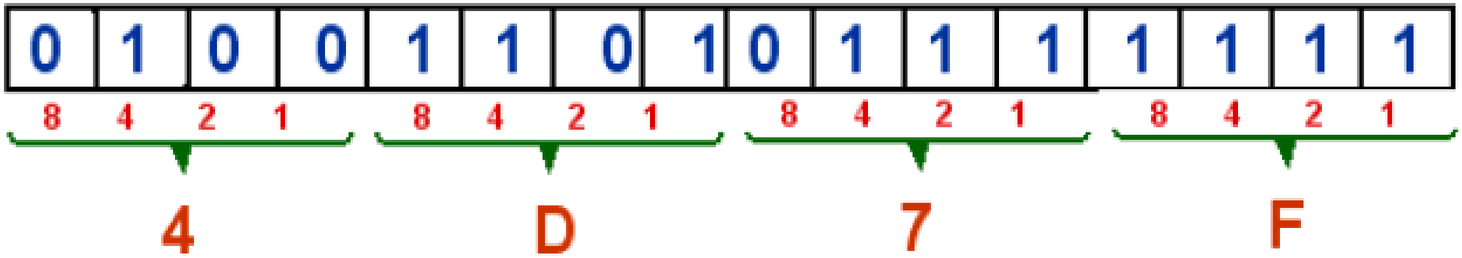
$456_{10} =$

$32_{10} =$

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa
binaire → hexa

Conversion binaire en hexadécimal

La conversion du binaire en hexadécimal est très simple, c'est d'ailleurs la raison pour laquelle nous utilisons cette base. Il suffit de faire correspondre un mot de quatre bits (quartet) à chaque chiffre hexadécimal.



$0100110101111111_2 = 4D7F_{16}$

On groupe les bits de 4 en 4 (en partant de la droite bien sûr) puis on converti simplement ces blocs de 4 bits en chiffres / lettres.

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa
binaire → hexa
hexa → binaire

Conversion hexadécimal en binaire

De la même manière pour convertir un nombre hexadécimal en binaire il suffit de convertir chacun des chiffres / lettres en bloc de 4 bits.

Attention de ne pas oublier les 0.

Exemple : AC74

A	C	7	4
⇩	⇩	⇩	⇩
1010	1100	0111	0100

$AC74_{16} = 1010110001110100_2$

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa
binaire → hexa
hexa → binaire

Exercice 5 :

a) Convertir les valeurs hexadécimal en binaire :

1A =

FF =

68 =

b) Convertir les valeurs binaire en hexadécimal :

10101101 =

01110010 =

10110011 =

c) Compléter le tableau :

Décimal	0	1	2	3	4	5	6	7
Binaire	0000	0001	0010					
Hexadécimal								
Décimal	8	9	10	11	12	13	14	15
Binaire								
Hexadécimal								

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa
binaire → hexa
hexa → binaire
ASCII

Le code ASCII

Afin de pouvoir utiliser en informatique les bases de numération qu'on a vu, il est encore nécessaire de définir des tables de codage permettant de traduire toutes les informations numériques ou alphabétiques sous forme de valeurs numériques exploitables par une machine.

Au début des années 1960, le code ASCII (American Standard Code for Information Interchange) voit le jour et est adopté comme standard. De base il permettait de représenter les caractères sur 7 bits et offrait donc la possibilité de coder 128 caractères différents. Ayant été inventé par des anglophones il ne comprend aucun caractère accentué.

Les codes 0 à 31	Caractères de contrôle (saut de page,...)
Les codes 33 à 47	Ponctuation
Les codes 48 à 57	Les chiffres 0 à 9
Les codes 58 à 64	Ponctuation
Les codes 65 à 90	Lettres majuscules
Les codes 91 à 96	Ponctuation
Les codes 97 à 122	Lettres minuscules
Les codes 123 à 127	Ponctuation et suppression

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa
binaire → hexa
hexa → binaire
ASCII

Le code ASCII étendu

Afin de répondre aux besoins internationaux, (caractères accentués et autres caractères propres à certaines langues), il a été mis au point un code ASCII étendu.

Afin de supporter tous les nouveaux caractères, ce code ASCII étendu utilise 8 bits offrant donc des valeurs allant de 0 à 255.

Exercice 6 :
Compléter le tableau suivant (avec l'aide de la table ASCII).

	Code ASCII en Hexa	Code ASCII en binaire
B		
o		
n		
j		
o		
u		
r		

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa
binaire → hexa
hexa → binaire
ASCII
Préfixes

Les préfixes

La notation kilo (k) est officiellement de 10^3. Indépendamment de l'unité : que ce soient des kilo-grammes (1000 g), des kilo-mètres, des kilo-Joules, des kilo-ampères, des kilo-hertz ou des kilo-octets.

1 kilo-octet = 1000 octets. 1 kilo-bits = 1000 bits.

Mais 1000 n'est pas un multiple de 2 aussi on a choisit le multiple de 2 le plus proche $2^{10} = 1024$.

De manière erronée, on a définit : 1 ko = 1024 octets = 1kB (comme Bytes).

Pour éviter les erreurs, en 1998 de nouveaux préfixes pour le binaire ont été créé comme le kibi.

1 kio = 1024 octets soit 1024x8 = 8192 bits.

Syst. Décimal
Syst. Binaire
binaire → décimal
décimal → binaire
Syst. Hexa
hexa → décimal
décimal → hexa
binaire → hexa
hexa → binaire
ASCII
Préfixes

Les préfixes

Multiples d'octets tels que définis par IEC 60027-2					
Préfixe SI			Préfixe binaire		
Nom	Symbole	Valeur	Nom	Symbole	Valeur
kilooctet	ko	10^3	kibioctet	Kio	2^{10}
mégaoctet	Mo	10^6	mébioctet	Mio	2^{20}
gigaoctet	Go	10^9	gibioctet	Gio	2^{30}
téraoctet	To	10^{12}	tébioctet	Tio	2^{40}
pétaoctet	Po	10^{15}	pébioctet	Pio	2^{50}
exaoctet	Eo	10^{18}	exbioctet	Eio	2^{60}
zettaoctet	Zo	10^{21}	zébioctet	Zio	2^{70}
yottaoctet	Yo	10^{24}	yobioctet	Yio	2^{80}

Encore de nos jours dans l'usage courant, on utilise le Ko à la place du Kio.

Attention donc de bien avoir la même convention que votre interlocuteur, pour éviter les erreurs.