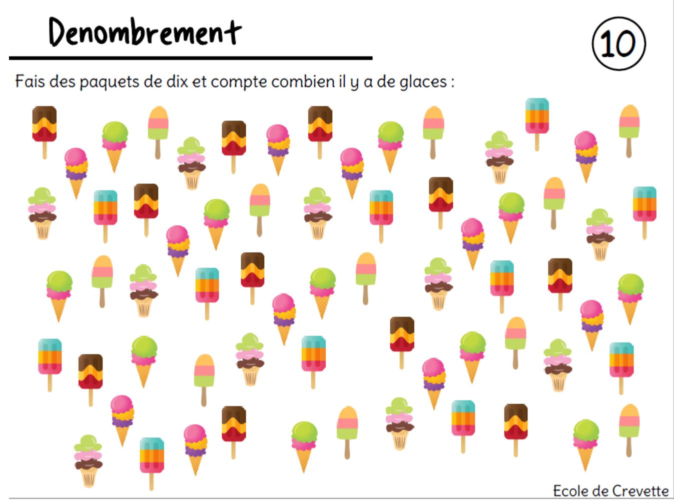
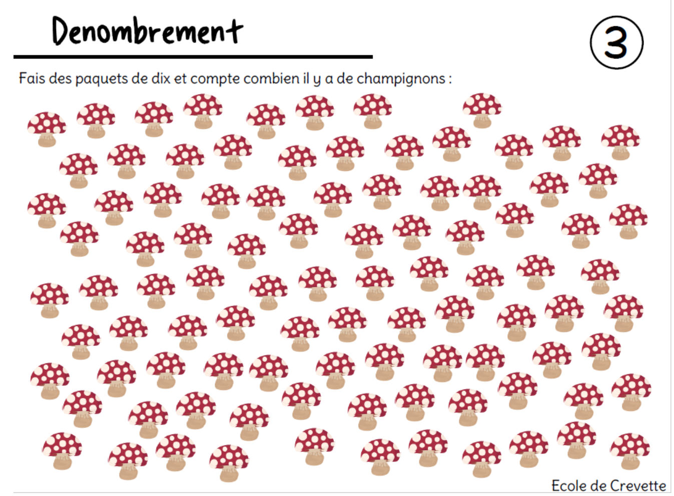
NSI – 1ere	COURS Séquence 2-AB : Écriture d'entiers positifs en base b	LFV

I. Représentation des entiers en base 2 (binaire)

1. Retour vers le passé

Voici ci-contre deux exercices classiques que vous avez tous fait dans votre enfance : dénombrer les objets d'une collection (source : http://www.ecoledecrevette.fr, Licence CC – BY – NC – ND).

<table>
<tr>
<td>


</td>
<td>


</td>
</tr>
<tr>
<td>On trouve : glaces.</td>
<td>On trouve : champignons.</td>
</tr>
</table>

2. Compter en base 10

Dans les deux exercices précédents vous avez utilisé une technique élémentaire pour dénombrer les objets. En base 10 on compte en effet en faisant des paquets de 10, des paquets de 100, des paquets de 1000 etc. On obtient ainsi l'écriture d'un nombre entier positif en base 10. Ainsi vous avez également tous effectué des exercices du type suivant :

Complète les pointillés :

$$3\,495 = 3 \times 1000 + 4 \times 100 + 9 \times 10 + 5 \times 1$$

$$3\,495 = 3 \times 10^3 + 4 \times 10^2 + 9 \times 10^1 + 5 \times 10^0$$

Cette façon de compter est tellement ancrée dans votre système de pensée mathématique qu'il y a une chose à expliquer. **Pourquoi fait-on des paquets de $10^0, 10^1, 10^2, 10^3$ *etc* ? Parce qu'on a que dix symboles à disposition : les chiffres de 0 à 9** donc on fait des paquets de puissances de dix.
En effet, si on cherche par exemple le nombre de combinaisons possibles obtenues avec 3 chiffres, on a 10 possibilités pour chacun des 3 chiffres, soit $10 \times 10 \times 10 = 10^3 = 1\,000$ combinaisons possibles, de 0 à 999. Donc au-delà de 999 il faut … nécessairement rajouter un quatrième chiffre : le chiffre des milliers !

3. Paquets utilisés en base 2

Compter en base 2 (c'est-à-dire en binaire), c'est compter en utilisant seulement deux symboles : typiquement les chiffres 0 et 1. On doit donc faire des paquets de $2^0 = 1,\ 2^1 = 2,\ 2^2 = 4,\ 2^3 = 8\ etc.$
Autant le dire tout de suite : les premières puissances de 2 doivent idéalement être apprises par cœur. Ou plutôt, vous devez être capables de les retrouver très rapidement de tête. Typiquement si on vous demande 2^7 vous devez très rapidement répondre 128.

Pour cela on dispose du tableau ci-dessous :

k	0	1	2	3	4	5	6	7	8	9	10	11
2^k	1	2	4	8	16	32	64	128	256	512	1024	2048

Les valeurs en gras doivent **vraiment** être apprises par cœur. Voici pourquoi :

- $2^4 = 16$ est utile car en informatique on travaille en octets (8 bits), qui sont souvent "découpés" en deux blocs de 4 bits représentés en héxadécimal, c'est-à-dire en base 16.

- $2^8 = 256$ est utile car l'utilisation des octets sur (8 bits) conduit de nombreuses grandeurs en informatique à être représentées par des entiers entre 0 et 255.

- $2^{10} = 1024$ est utile car il permet d'évaluer approximativement de tête une grande puissance de 2. Ainsi $2^{35} = (2^{10})^3 \times 2^5 = (1024)^3 \times 2^5 \approx 1000^3 \times 32 \approx 32$ milliards.

Si vous connaissez ces trois puissances de deux par cœur, vous retrouverez rapidement les autres en multipliant ou divisant par 2 ces puissances.

4. Conversions de la base 2 vers la base 10

Prenons un exemple : on cherche à représenter 167 en base 2. On va donc l'écrire en utilisant que des puissances de 2 présentes dans le tableau ci-dessus.

$167 = 128 + 39$

$167 = 128 + 32 + 7$

$167 = 128 + 32 + 4 + 3$

$167 = 128 + 32 + 4 + 2 + 1$

$167 = 2^7 + 2^5 + 2^2 + 2^1 + 2^0$

$167 = \mathbf{1} \times 2^7 + \mathbf{0} \times 2^6 + \mathbf{1} \times 2^5 + \mathbf{0} \times 2^4 \quad + \quad \mathbf{0} \times 2^3 + \mathbf{1} \times 2^2 + \mathbf{1} \times 2^1 + \mathbf{1} \times 2^0$

$167 = \mathbf{1010\ 0111_2}$

Remarque : On pourrait facilement confondre les deux écritures suivantes ;

- 1010 0111 qui est la représentation de cent soixante sept en base 2
- 10 100 111 qui est la représentation de dix millions cent-mille cent onze en base 10

C'est pourquoi lorsqu'on manipule des bases différentes on indique la base en indice :

$$167_{10} = 1010\ 0111_2$$

Exercice de cours

Représenter les nombres suivants en base 2. On laissera apparentes les écritures intermédiaires.

$43_{10} = 32 + 8 + 2 + 1$ $43_{10} = 101011_2$	$411_{10} = 256 + 128 + 16 + 8 + 2 + 1$ $411_{10} = 110011011_2$	$1023_{10} = 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$ $1023_{10} = 1111111111_2$

Exercice de cours

Représenter les nombres suivants en base 10. On laissera apparentes les écritures intermédiaires.

$0110\ 1101_2 = 0 + 64 + 32 + 0 + 8 + 4 + 0 + 1$ $0110\ 1101_2 = 109_{10}$	$10\ 1000\ 1001_2 = 512 + 0 + 128 + 0 + 0 + 0 + 8 + 0 + 0 + 1$ $10\ 1000\ 1001_2 = 649_{10}$	$1111\ 1111_2 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$ $1111\ 1111_2 = 255_{10}$

II. Représentation des entiers en base 16 (hexadécimal)

1. Conversions de la base 2 à la base 16

En base 16 nous allons faire des paquets de :

$$16^0 = 1, \quad 16^1 = 16, \quad 16^2 = 256, \quad 16^3 = 4096 \ldots$$

Nous allons par ailleurs être confrontés au nombre de symboles utilisables :

- En base 2 nous utilisons 2 symboles : les deux chiffres 0 et 1.
- En base 10 nous utilisons 10 symboles : les dix chiffres $0, 1, 2, 3, 4, 5, 6, 7, 8$ et 9.
- En base 16 nous utilisons 16 symboles : les dix chiffres de 0 à 9 et les six lettres A, B, C, D, E, F.

Ainsi en base 16, on a l'utilisation des seize *symboles* suivants :

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Avant de s'attaquer à la conversion de base 2 vers base 16, remarquons qu'en partant de la base 10 on décompose facilement un nombre en base $mille =$ base 10^3 en regroupant les chiffres par 3. Avec des notations abusives on a par exemple :

$$954\,324\,546\,765_{10} = 954_{10}\ Milliards + 324_{10}\ Millions + 546_{10}\ Mille + 765_{10}$$

$$954\,324\,546\,765_{10} = 954_{10}\ mille^3 + 324_{10}\ mille^2 + 546_{10}\ mille^1 + 765_{10}\ mille^0$$

De façon similaire on passe facilement de la base 2 à la base $seize = 2^4$ en regroupant les bits par 4.

Si on part d'un nombre représenté en base 2, on obtient facilement sa représentation en base $seize = 2^4$ en regroupant les bits par 4 :

$$1110\ 1011\ 0101\ 1001_2 = 1110_2\ seize^3 + 1011_2\ seize^2 + 0101_2\ seize^1 + 1001_2\ seize^0$$

$$1110\ 1011\ 0101\ 1001_2 = 14_{10}\ seize^3 + 11_{10}\ seize^2 + 5_{10}\ seize^1 + 9_{10}\ seize^0$$

$$1110\ 1011\ 0101\ 1001_2 = E_{16}\ seize^3 + B_{16}\ seize^2 + 5_{16}\ seize^1 + 9_{16}\ seize^0$$

$$1110\ 1011\ 0101\ 1001_2 = EB59_{16}$$

Ainsi chaque groupe de 4 bits en base 2 correspond à un symbole entre 0 et 15 en base 16

Exercice de cours

Représenter les nombres suivants en base 16. On laissera apparentes les écritures intermédiaires.

$1000\ 0110_2 = 86_{16}$	$0110\ 1100\ 1000_2 = 6C8_{16}$	$11\ 0110\ 1011\ 0110_2 = 36B6_{16}$

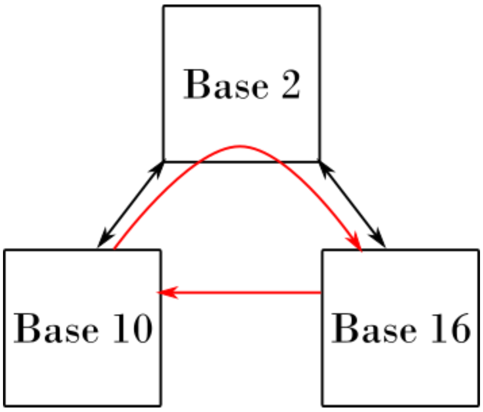
Exercice de cours

Représenter les nombres suivants en base 2. On laissera apparentes les écritures intermédiaires.

$A4_{16} = 1010\ 0100_2$	$B0B_{16} = 1011\ 0000\ 1011_2$	$1011_{16} =$ $0001\ 0000\ 0001\ 0001_2$

2. Conversions de la base 16 vers la base 10

On peut passer directement et rapidement de la base 16 à la base 10. En revanche pour passer directement de la base 10 à la base 16 il est plus rapide de passer par la base 2. Ce qui se résume par le schéma ci-contre.



Exercice de cours

Représenter les nombres suivants en base 16. On laissera apparentes les écritures intermédiaires.

642_{10} = 10 1000 0010_2 642_{10} = 282_{16}	176_{10} = 1011 0000_2 176_{10} = $B0_{16}$	255_{10} = 1111 1111_2 255_{10} = FF_{16}

Exercice de cours

Représenter les nombres suivants en base 10. On laissera apparentes les écritures intermédiaires.

$D7_{16}$ = 1101 0111_2 $D7_{16}$ = 215_{10} ou $D7_{16}$ = 13*16 + 7*1	$6C_{16}$ = 0110 1100_2 $6C_{16}$ = 108_{10} ou $6C_{16}$ = 6*16 + 12*1	$7C3_{16}$ = 0111 1100 0011_2 $7C3_{16}$ = 1987_{10} ou $7C3_{16}$ = $7*16^2$ + 12*16 + 3*1

III. Représentation en base b – Algorithme des divisions successives

1. Passer de la base b à la base 10

Il suffit de se rappeler que le $i^{ème}$ chiffre en partant de droite représente le nombre de paquet de taille b^{i-1}.

Exemple :

$342_7 = 3 \times 7^2 + 4 \times 7^1 + 2 \times 7^0$
$342_7 = 3 \times 49 + 4 \times 7 + 2 \times 1$
$342_7 = 147 + 28 + 2$
$342_7 = 177_{10}$

2. Passer de la base 10 à la base b

Nous ne démontrerons pas la méthode suivante. **Il s'agit de la méthode des divisions successives qui fonctionne pour toutes les bases b.** Il s'agit, comme son nom l'indique, d'effectuer des divisions successives par b et de noter les restes successifs. Le mieux est de voir sur deux exemples :

Exemple 1 : convertir 4579_{10} en base 7

```
4579 % 7 = 1  et  4579 // 7 = 654
 654 % 7 = 3  et   654 // 7 = 93
  93 % 7 = 2  et    93 // 7 = 13
  13 % 7 = 6  et    13 // 7 = 1
   1 % 7 = 1  et     1 // 7 = 0
```

On en déduit en regardant les restes successifs que $4579_{10} = 16231_1$
On vérifie aisément que cela fonctionne : $1 \times 7^4 + 6 \times 7^3 + 2 \times 7^2 + 3 \times 7^1 + 1 \times 7^0 = 4579$.

Exemple 2 : convertir 23564_{10} en base 9

On trouve 35282_9

Exemple 3 : Avec cette méthode convertir 777_{10} en base 2 puis en base 16

On trouve $11\ 0000\ 1001_2$ puis 309_{16}

Quelle est la meilleure méthode pour passer de base 10 à base 2 ou 16 ?

Nous avons vu deux méthodes :

- Celle des I et II de ce cours qui se base sur la décomposition en paquets de taille 2^k et permet donc de comprendre ce qu'est réellement l'écriture binaire. De plus elle fait le lien entre groupe de 4 bits et hexadécimal. Cette méthode apporte donc de la compréhension.
  En revanche elle nécessite de connaître les différentes puissances de 2 et ne fonctionne que pour la base 2, extensible à la base 16.

- Celle des divisions successives qui a le mérite de fonctionner pour toutes les bases et qui – surtout – peut facilement se programmer. En revanche on ne comprend que difficilement pourquoi cela fonctionne.

Connaître la décomposition en paquets de puissances de 2 est de toute façon obligatoire pour faire la conversion dans l'autre sens. La première méthode doit donc être connue.

Savoir passer de la base 10 à une autre base b dans le cas général est au programme, la seconde méthode doit donc également être connue.

C'est donc à vous de voir quelle méthode vous convient le plus lorsque vous devez passer de la base 10 (décimal) à la base 2 (binaire) ou 16 (hexadécimal).

IV. Exercices

Exercice 1 : écriture en base 2 des puissances de 2 : 2^k

Écrire sur 8 bits les entiers suivants :

$$128_{10} = 1000\ 0000_2$$

$$64_{10} = 0100\ 0000_2$$

$$32_{10} = 0010\ 0000_2$$

$$16_{10} = 0001\ 0000_2$$

$$8_{10} = 0000\ 1000_2$$

$$4_{10} = 0000\ 0100_2$$

$$2_{10} = 0000\ 0010_2$$

$$1_{10} = 0000\ 0001_2$$

Exercice 2 : écriture en base 2 des $2^k - 1$

Écrire sur 8 bits les entiers suivants :

$$127_{10} = 0111\ 1111_2$$

$$63_{10} = 0011\ 1111_2$$

$$31_{10} = 0001\ 1111_2$$

$$15_{10} = 0000\ 1111_2$$

$$7_{10} = 0000\ 0111_2$$

$$3_{10} = 0000\ 0011_2$$

$$1_{10} = 0000\ 0001_2$$

Exercice 3 : conversion vers base 2

Représenter les nombres suivants en base 2. On laissera apparentes les écritures intermédiaires.

43_{10} 101011_2	411_{10} $1\ 1001\ 1011_2$	$1AC_{16}$ $1\ 1010\ 1100_2$
14_{10} 1110_2	178_{10} $1011\ 0010_2$	$F841_{16}$ $1111\ 1000\ 0100\ 0001_2$

Exercice 4 : conversion vers base 16

Représenter les nombres suivants en base 16. On laissera apparentes les écritures intermédiaires.

$1001\ 0011\ 1010_2$ $93A_{16}$	999_{10} $3E7_{16}$	4095_{10} FFF_{16}

$1111\ 0001\ 1000_2$ F18$_{16}$	$1001\ 0000\ 1100_2$ 90C$_{16}$	145_2 pas possible !!!

Exercice 5 : conversion vers base 10

Représenter les nombres suivants en base 10. On laissera apparentes les écritures intermédiaires.

$11\ 1111\ 0011_2$ 1011$_{10}$	$B52_{16}$ 2898$_{10}$	1010_{16} 4112$_{10}$
1010_2 10$_{10}$	FF_{16} 255$_{10}$	$1111\ 1111_2$ 255$_{10}$

Exercice 6 : algorithme des divisions successives

- Représenter 179_{10} en base 3 On obtient 20122_3
- Représenter 587_{10} en base 5 On obtient 4322_5
- Représenter 178962_{10} en base 20 (système *vicésimal*, pour les amateurs d'Histoire, se renseigner sur l'Hôpital des Quinze-Vingt). On obtient 12782_{20}