

Ecriture d'un entier positif dans une base $b \geq 2$

Programme 1ère

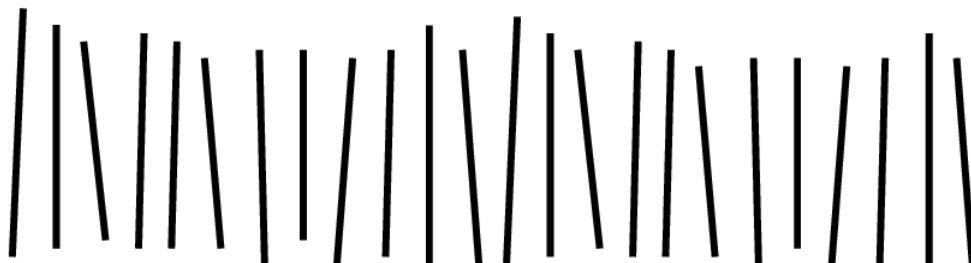
|Contenus|Capacités attendues|Commentaires| |---:|---:|---:| |Ecriture d'un entier positif dans une base $b \geq 2$ |
Passer de la représentation d'une base à l'autre|Les bases 2, 10 et 16 sont privilégiées|

1. Écriture d'un entier dans une base

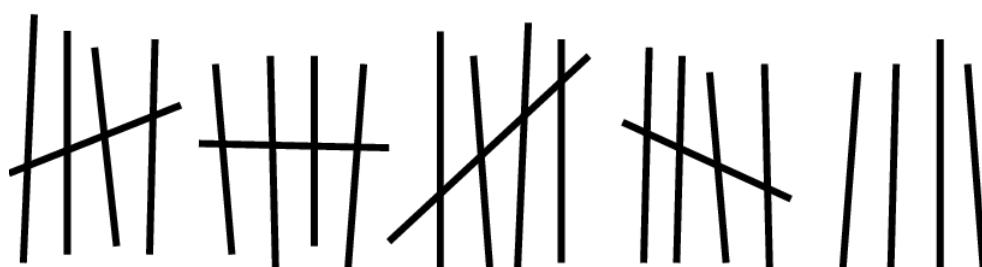
Le monde se divise en 2 catégories : ceux qui comprennent le binaire et ceux qui ne le comprennent pas.

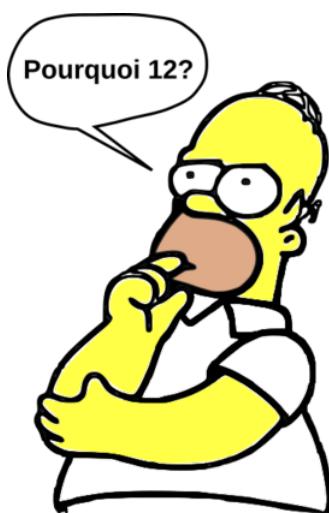
1.1. Le système décimal

Depuis la préhistoire, l'Homme a besoin de compter. Sa première idée:



Pas très pratique... Heureusement, il en eut rapidement une autre, beaucoup plus efficace: regrouper les bâtons en paquets.





Des paquets de 5 et surtout de 10. Pourquoi 10 ?

Pour écrire les nombres, regroupés en paquets de 10, il a donc fallu inventer 10 chiffres pour compter les unités (en-dessous du paquet): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Et le nombre 10 ne signifie ni plus ni moins que 1 paquet et 0 unité.

Ensuite, on fait un paquet de paquets, la centaine (100, c'est à dire $10 \times 10 = 10^2$), puis un paquet de centaines, le millier (1000, c'est-à-dire $10 \times 10 \times 10 = 10^3$), et ainsi de suite.

Au final, si on écrit le nombre 2022 on sait que c'est:

- **2** mille, c'est-à-dire 2×10^3 ;
- **0** centaine, c'est-à-dire 0×10^2 ;
- **2** dizaines, c'est-à-dire 2×10^1 ;
- **2** unité, c'est-à-dire 2×10^0 .

Et bien entendu, $2022 = 2 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 2 \times 10^0$

La position de chaque chiffre dans l'écriture du nombre correspond à la puissance de 10 par laquelle on le multiplie.

 Homer compte

Octal	Décimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
10	8
11	9
12	10
13	11
14	12
15	13
16	14
17	15
20	16
21	17
...	...

Un autre exemple, la base octale

Pour Homer Simpson, qui n'a que 8 doigts, il aurait été certainement plus naturel de faire des paquets de 8... et donc de manipuler seulement 8 chiffres : 0, 1, 2, 3, 4, 5, 6 et 7.

C'est le **système octal**.

Ainsi, «son» 10 signifierait 1 paquet de 8 et 0 unité, soit «notre» 8... Et donc notre 10 (en décimal) s'écrirait 12 (en octal).

Enfin, 2022 (en décimal) s'écrit 3746 en octal puisque :

$$\begin{aligned} 3746_8 &= 3 \times 8^3 + 7 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 \\ &= 8 \times 512 + 7 \times 64 + 4 \times 8 + 6 \times 1 \\ &= 1536 + 448 + 32 + 6 \\ &= 2022 \end{aligned}$$

2. ---

2.1. Notion de base

Écriture d'un entier en base b

- Choisir une base, c'est choisir la «taille» du paquet, et donc le nombre de chiffres dont on aura besoin: en base b , il y a b chiffres.
- Dans notre système décimal, la base est $b = 10$.
- La position de chaque chiffre correspond à la puissance de b par laquelle on le multiplie.
- On précisera la base (si différente de 10) en indice en fin du nombre. Par exemple, le nombre 312_5 est écrit en base 5.

Exemples

$$\begin{aligned}312_5 &= 3 \times 5^2 + 1 \times 5^1 + 2 \times 5^0 \\&= 3 \times 25 + 1 \times 5 + 2 \times 1 \\&= 82\end{aligned}$$

$$\begin{aligned}25072_8 &= 2 \times 8^4 + 5 \times 8^3 + 0 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 \\&= 2 \times 4096 + 5 \times 512 + 7 \times 8 + 2 \times 1 \\&= 10810\end{aligned}$$

2.2. Le système binaire (base 2)

Le binaire

- Le binaire, c'est le système de comptage des ordinateurs. Pourquoi le binaire et pas le décimal comme les humains ?
- Un ordinateur est composé de circuits électroniques et de composants électriques, alimentés par un signal électrique : 1, il y a du courant, 0, il n'y en a pas. Le plus simple pour compter est donc d'utiliser un système en base 2 (le binaire) car on peut représenter ses deux valeurs possibles (0 et 1). (voir vidéo suivante).
- Le système binaire est le système de numération de base $b = 2$.
- Il utilise donc seulement deux chiffres, 0 et 1, appelés **bits** (contraction de l'anglais binary digit). Le nombre 10101 occupe 5 rangs. On dit qu'il s'écrit sur 5 bits
- Un regroupement de 8 bits est appelé un **octet**.

 **Vidéo**

INTRODUCTION AU LANGAGE BINAIRES

 **Exercice 19**

1. Combien de valeurs peut-on coder avec 1 bit ?
2. Combien de valeurs peut-on coder avec 2 bits ?
3. Combien de valeurs peut-on coder avec 3 bits ?
4. Combien de valeurs peut-on coder avec n bits ?

i Conversions

Binaire → Décimal

Il est utile de connaître alors les puissances de 2 pour convertir rapidement de l'écriture binaire vers l'écriture décimale.

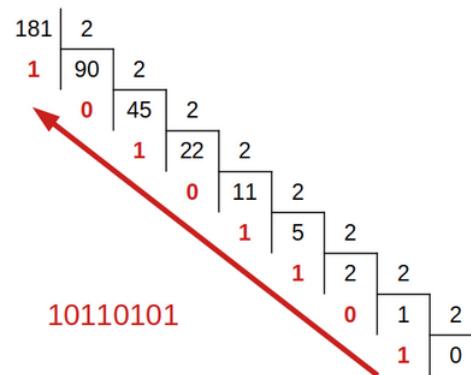
puissances de 2 : 128 64 32 16 8 4 2 1

bits :	1	0	1	1	0	1	0	1
--------	---	---	---	---	---	---	---	---

Par exemple $10110101_2 = 128 + 32 + 16 + 4 + 1 = 181$.

Décimal → Binaire Méthode 1

- On peut utiliser l'algorithme de soustraction: on soustrait du nombre la plus grande puissance de 2 possible, et on recommence...
- On utilise l'algorithme de divisions: on effectue les divisions successives du nombre par 2. L'écriture en binaire est donnée par les restes lus de bas en haut.



Décimal → Binaire Méthode 2

- On peut utiliser un tableau. Par exemple pour 217

2^7	2^6	2^5	2^4	2^3	2^2	2^1
128	64	32	16	8	4	2
1	1	0	1	1	0	0
217-128=89	89-64=25	25	25-16=9	9-8=1	1	1

Décimal → Base quelconque

Les deux algorithmes de soustraction et de divisions fonctionnent également pour la conversion du décimal vers n'importe quelle autre base.

Exercice 20

Écrire 96, 196, 119 et 168 en binaire avec cette méthode.

2.3. Le système hexadécimal (base 16)

L'hexadécimal

- Le système hexadécimal est le système de numération de base $b = 16$.
- Il utilise donc 16 chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
- Bien qu'il nécessite plus de chiffres, ce système donne des écritures plus courtes et particulièrement pratique du fait que 16 est une puissance de 2 : la conversion entre les systèmes binaire et hexadécimal se fait sans calcul, uniquement par regroupement de 4 bits.

Couleurs

Le système hexadécimal est en particulier un mode de code informatique des couleurs.

En effet une couleur (codage RGB) est composée de 3 valeurs correspondant aux composantes rouge, verte et bleue, et chacune de ces valeurs est un entier compris entre 0 et 255, c'est-à-dire entre 0 et FF en hexadécimal: on peut les écrire avec seulement deux chiffres.

Par exemple, la couleur rouge ci-contre a pour composantes (178, 26, 15), ce qui donne en hexadécimal : (B2, 1A, 0F), qu'on écrit en général de façon concaténée B21A0F et souvent précédée d'un #.



 **Convertisseur**

Un [convertisseur](#) bien utile.

2.4. Exercices

2.4.1. À la main

 **Exercice 21**

Convertir de la base mentionnée vers la base 10 (système décimal).

Attention, il y a un piège...

1. 131_6
2. 420_5
3. 1515_8
4. 716_4
5. 321_4

 **Exercice 22**

Convertir les écritures décimales vers la base mentionnée.

1. 47 en base 5
2. 92 en base 4

 Exercice 23

Convertir du binaire vers le décimal.

1. 1010
2. 101110
3. 10001101
4. 11111111
5. 101011001

 Exercice 24

Convertir du décimal vers le binaire.

1. 17
2. 34
3. 68
4. 100
5. 200

Que remarquez-vous quand on passe de l'écriture binaire d'un nombre à celle de son double?

 Exercice 25

1. Vérifier la conversion de la couleur rouge donnée en exemple à la section précédente.

 Algo

2. Convertir les écritures hexadécimales en décimal: 99, D7, 1B8, ABC.
3. Comment s'écrit la couleur blanche en code hexadécimal?

 **Exercice 26**

Convertir les écritures décimales en hexadécimal.

1. 45
2. 72
3. 421

 **Exercice 27**

C'est en 11110010000 qu'a été démontré le théorème fondamental de l'informatique. Exprimer ce nombre en base dix

 **Exercice 28**

Convertir les nombres suivants en base 10 :

1. $(1B7)_{16}$
2. $(ABE)_{16}$
3. $(A320)_{16}$

Exercice 29

Conversion Binaire → Hexadécimal

- On regroupe les bits par paquets de 4 (quitte à ajouter des 0 pour compléter).
- On convertit chaque paquet en chiffre hexadécimal.
- On concatène les chiffres

Par exemple, $11111100101_2 = \underbrace{0111}_7 \underbrace{1110}_{14=E_{16}} \underbrace{0101}_5 = 7E5_{16}$

Conversion Hexadécimal → Binaire

On fait la même chose dans l'autre sens.

Application

1. Convertir 101101_2 puis 10011011111_2 en base 16.
2. Convertir $8D_{16}$ puis $1CA7_{16}$ en binaire.

Exercice 30

Conversion Binaire → Hexadécimal

Convertir les nombres binaires suivants en hexadécimal

1. $(01001101)_2$
2. $(100110000111)_2$
3. $(101110101100)_2$

Conversion Hexadécimal → Binaire

Convertir en binaire les nombres suivant :

1. $(FABE51)_{16}$
2. $(59A75)_{16}$

2.4.2. Avec Python

Fonctions de conversion

Pour convertir un entier écrit en décimal vers le binaire ou l'hexadécimal, on peut utiliser les fonctions `bin` et `hex`, qui renvoient des chaînes de caractères.

Script Python

```
>>> bin(2021)
'0b11111100101'
>>> hex(2021)
'0x7e5'
>>>
```

Les préfixes `0b` et `0x` indiquent que les écritures sont respectivement en base 2 et en base 16.

Réiproquement, pour convertir l'écriture d'un nombre écrit en base b vers le décimal, on utilise la fonction `int` en passant en arguments l'écriture en chaîne de caractères et la base.

Script Python

```
>>> int('11111100101', 2)
2021
>>> int('7e5', 16)
2021
>>>
```

Exercice 31

Utilisez ces fonctions pour vérifier vos résultats des exercices précédents.

```
>>>
```

3. Integer Overflow

 **Enoncé du problème**

Dans une usine, sur une machine qui fabrique en moyenne 200 pièces par jour, un compteur stocké sur un octet affiche le nombre de pièces produites. quand un technicien récupère les pièces fabriquées, il vérifie que le nombre est conforme, puis appuie sur un bouton pour remettre le compteur à zéro. Un jour le compteur affiche 21 alors que 277 pièces ont été produites. Il appelle le numéro d'assistance pour signaler que le compteur ne fonction plus. On lui répond que c'est normal et qu'une mise à jour sera effectuée prochainement pour corriger ce défaut. Essayons d'expliquer ce qui est arrivé !

 **Exercice 32**

1. Donner l'écriture binaire de 277_{10} .
2. Sachant qu'un octet est composé de 8 bits, compléter un octet avec le nombre obtenu ? Que remarquez-vous ?
3. Expliquer alors le résultat 21 qui apparaît sur le compteur.

 **A retenir**

Si le nombre de bits obtenu est inférieur à n, on rajoute des 0 pour parvenir à n bits. S'il n'y a pas assez de bits disponibles, on dit que la capacité d'écriture est insuffisante (on parle d'overflow).


Enoncé du problème

Combien de bits faut-il pour représenter un nombre entier en binaire ?

Pour évaluer le nombre de bits minimum nécessaires à l'écriture en base 2 d'un entier positif, il faut trouver la plus petite puissance de 2 qui soit strictement supérieure à l'entier à écrire.

Exemple : comme $2^8 = 256$ et $2^9 = 512$, $2^8 \leq 277 < 2^9$ donc 9 bits sont nécessaires pour écrire l'entier 277.

1. Combien de bits minimum sont nécessaires pour stocker les entiers 1047 et 65 512 ?
2. Remplir le tableau suivant avec les tailles fréquentes des entiers en binaire et généralisation.

Nombre d'octets utilisés	1	2	4	8	n
Nombre de bits utilisés	8				
Nombre d'entiers positifs qu'on peut stocker	$2^8 = 256$				
Plus grand entier positif que l'on peut stocker	$2^8 - 1 = 255$				

4. Addition en binaire

Voyons maintenant comment on ajoute des nombres exprimés en base deux

$$\begin{array}{cccc}
 & 1 & 1 & 0 \\
 & 1 & 0 & 1 & \rightarrow & 5_{10} \\
 + & 1 & 1 & 1 & \rightarrow & 7_{10} \\
 \hline
 & 1 & 1 & 0 & 0 & \rightarrow & 12_{10}
 \end{array}$$

 Exercice 33

Faites les additions suivantes :

- 1.** A=1101+10
- 2.** B=101001+101
- 3.** C=10101010+1011
- 4.** D=1110101+10101