

Codage des entiers positifs en base b

Thème 1 - Représentation de données - Types et valeurs de bases

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

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.

Le système décimal

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

center w:400px

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

center w:400px

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

!!! note inline end "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| |...|...|

!!! info
“Un
autre
exem-
ple, la
base
octale”
Pour
Homer
Simp-
son,
qui n’a
que 8
doigts,
il
aurait
été cer-
taine-
ment
plus
na-
turel
de
faire
des pa-
quets
de 8...
et
donc
de ma-
nipuler
seule-
ment 8
chiffres
: 0, 1,
2, 3, 4,
5, 6 et
7.
C’est
le **sys-
tème
octal**.

Ainsi,
«son»
10
signi-
fierait
1
paquet
de 8 et
0 unité,
soit
«notre»
8...
Et
donc
notre
10 (en
déci-
mal)
s'écirait
12 (en
octal).
Enfin,
2022
(en
déci-
mal)
s'écrit
3746
en
octal
puisque
:

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

Notion de base

!!! abstract “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

!!! note “Exemples”

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

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

Le système binaire (base 2)

!!! abstract “Le binaire”

- Le binaire, c’est le système de comptage des ordinateurs. Pourquoi le binaire et pas le décimal ?
- Un ordinateur est composé de circuits électroniques et de composants électriques, alimentés en courant continu.
- 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*).
- Un regroupement de 8 bits est appelé un **octet**.

??? video “Vidéo”

!!! exo “{ exercice() }”

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 ?

!!! info “Conversions” == “Binaire \rightarrow Décimal” Il est utile de connaître alors les puissances de 2 pour convertir rapidement de l’écriture binaire vers l’écriture décimale.

{: .center width=50%}

Par exemple $10110101_2 = 128 + 32 + 16 + 4 + 1 = 181_{10}$.

=== "Décimal → Binaire Méthode 1"

- On peut utiliser l'algorithme de soustraction: on soustrait du nombre la plus grande p

- On utilise l'algorithme de divisions: on effectue les divisions successives du nombre

{: .center}

=== "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	2^0	
128	64	32	16	8	4	2	1	
1	1	0	1	1	0	0	1	
217-128=89		89-64=25		25	25-16=9		9-8=1	1 1 1-1=0

=== "Décimal → Base quelconque"

Les deux algorithmes de soustraction et de divisions fonctionnent également pour la conv

!!! exemple “{{ exercice() }}” Écrire 96,196,119 et 168 en binaire avec cette méthode.

Le système hexadécimal (base 16)

!!! abstract “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 part

!!! info “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 r

{align=right}

Par exemple, la couleur rouge ci-contre a pour composantes (178, 26, 15), ce qui donne en he

??? tip “Convertisseur” Un convertisseur{:target="__blank"} bien utile.

Exercices

À la main !!! exemple “{{ exercice() }}" 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\$

!!! exemple “{{ exercice() }}" Convertir les écritures décimales vers la base mentionnée.

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

!!! exemple “{{ exercice() }}" Convertir du binaire vers le décimal.

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

!!! exemple “{{ exercice() }}" 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?

!!! exemple “{{ exercice() }}"

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

2. Convertir les écritures hexadécimales en décimal: 99, D7, 1B8, ABC.

3. Comment s'écrit la couleur blanche en code hexadécimal?

!!! exemple “{{ exercice() }}" Convertir les écritures décimales en hexadécimal.

1. 45
2. 72
3. 421

!!! exemple “{{ exercice() }}” C’est en 11110010000 qu’a été démontré le théorème fondamental de l’informatique. Exprimer ce nombre en base dix

!!! exemple “{{ exercice() }}” Convertir les nombres suivants en base 10 :

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

!!! exemple “{{ exercice() }}” === “Conversion Binaire \rightarrow 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, $(11111100101)_2 = \underbrace{0111}_7 \underbrace{1110}_{14} = \text{E}_{16}$

=== "Conversion Hexadécimal \rightarrow Binaire"
On fait la même chose dans l'autre sens.

=== "Application"

1. Convertir $(101101)_2$ puis $(1001101111)_2$ en base 16.
2. Convertir $(D)_{16}$ puis $(CA)_{16}$ en binaire.

!!! exemple “{{ exercice() }}”

=== “Conversion Binaire \rightarrow Hexadécimal” Convertir les nombres binaires suivants en hexadécimal

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

=== "Conversion Hexadécimal \rightarrow Binaire"
Convertir en binaire les nombres suivant :
1. $(FABE51)_{16}$
2. $(59A75)_{16}$

Avec Python ??? info “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.

```
```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.

Réciproquement, pour convertir l'écriture d'un nombre écrit en base b vers le décimal, on

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

!!! exemple “{{ exercice() }}” Utilisez ces fonctions pour vérifier vos résultats des exercices précédents.

{{ terminal() }}

## Integer Overflow

!!! objectif “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 !

!!! exo “{{ exercice() }}” 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.

!!! savoir “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).

!!! objectif “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

Exemple : comme  $2^8 = 256$  et  $2^9 = 512$ ,  $2^8 \leq 277 < 2^9$  donc 9 bits sont nécessaires

**\*\*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 générer



```

|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| $2^8 - 1 = 255$ |||

```

## Addition en binaire

Voyons maintenant comment on ajoute des nombres exprimés en base deux

```
{:.center width=50%}
```

!!! exo “{ { exercice() } }” Faites les additions suivantes :

```

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

```