ALGORITHMIQUE · CHAPITRE 6 · FONDATIONS

Écrire un nombre en bases 2, 8, 10 et 16

Comprenez la numération de position, convertissez un entier entre bases 2, 8, 10 et 16 et reliez chiffres binaires, octaux et hexadécimaux.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Distinguer une valeur de son écriture dans une base.
- Convertir un entier non négatif par poids de position ou divisions successives.
- Relier un chiffre octal à trois bits et un chiffre hexadécimal à quatre bits.

Prérequis. Connaître les entiers, les puissances, le quotient entier et le reste.

26 et 1A peuvent-ils désigner le même nombre ?

On peut compter vingt-six objets sans choisir leur écriture. Le nombre s'écrit 26 en décimal, 11010 en binaire, 32 en octal et 1A en hexadécimal. Une **base de numération** organise l'écriture ; elle ne change pas la quantité.

Comprendre la numération de position

Dans une base entière $b \geq 2$, chaque chiffre a une valeur de 0 à b - 1. En partant de la droite, les positions ont pour poids $b^0 = 1$, $b^1 = b$, b^2, puis les puissances suivantes. La valeur est la somme des chiffres multipliés par leurs poids. Nous étudions ici les entiers non négatifs.

Chiffres autorisés selon la base

Base	Nom	Chiffres	Écriture de vingt-six

Base	Nom	Chiffres	Écriture de vingt-six
2	Binaire	0, 1	11010_{2}
8	Octale	0 à 7	32_{8}
10	Décimale	0 à 9	26_{10}
16	Hexadécimale	0 à 9, A à F	$1A_{16}$

En hexadécimal, A vaut 10, B vaut 11, C vaut 12, D vaut 13, E vaut 14 et F vaut 15. Les indices $_{2}$, $_{8}$, $_{10}$, $_{16}$ indiquent la base dans le texte mathématique. `26₁₀ = 2 × 10 + 6` ; `1A₁₆ = 1 × 16 + 10` ; `11010₂ = 16 + 8 + 2` ; `32₈ = 3 × 8 + 2`. Le chiffre 8 n'est pas autorisé en base 8.

Convertir par divisions successives

Pour passer de 26 à la base 2, divisez par 2 et conservez les restes. Recommencez avec le quotient tant qu'il est non nul. Le premier reste donne le chiffre de droite, car `n = 2q + r`. Diviser q décale le raisonnement vers la position suivante.

Divisions de vingt-six par deux

Nombre	Quotient	Reste
26	13	0
13	6	1
6	3	0
3	1	1
1	0	1

Lisez les restes du bas vers le haut : 11010. La même méthode avec le diviseur 8 donne les restes 2 puis 3, donc 32_8 ; avec 16, les restes 10 (A) puis 1, donc $1A_{16}$. Pour zéro, l'écriture est simplement 0.

Relier binaire, octal et hexadécimal

Un **bit** est un chiffre binaire : 0 ou 1. Comme `8 = 2³`, un chiffre octal correspond à trois bits ; comme `16 = 2⁴`, un chiffre hexadécimal correspond à quatre bits. Groupez depuis la droite et complétez à gauche avec des zéros : `011 010₂ = 32₈` et `0001 1010₂ = 1A₁₆`. Ces zéros à gauche ne changent pas la valeur.

Comparer écriture et valeur

En Python, `int("1A", 16)` convertit le texte hexadécimal en entier. `bin`, `oct` et `hex` produisent des textes préfixés par `0b`, `0o` et `0x`. Les littéraux `0b11010`, `0o32` et `0x1A` ont tous la valeur entière 26. MATLAB utilise ici `base2dec` pour lire et `dec2base` pour produire le texte dans la base choisie ; les apostrophes délimitent le texte.

Pseudo-code

```
// example.pseudo
// positional_value
value <- 1 * 16 + 10
quotient <- floor_divide(value, 2)
remainder <- value MOD 2
PRINT value, quotient, remainder
```

Python

```python
# positional_value.py
value = int("1A", 16)
quotient = value // 2
remainder = value % 2
print(value, quotient, remainder)
print(bin(value), oct(value), hex(value))
```

MATLAB

```matlab
% positional_value.m
value = base2dec('1A', 16);
quotient = floor(value / 2);
remainder = mod(value, 2);
disp([value, quotient, remainder]);
disp(dec2base(value, 2));
disp(dec2base(value, 8));
disp(dec2base(value, 16));
```

Python affiche 26, 13, 0 puis `0b11010 0o32 0x1a`. MATLAB affiche les mêmes nombres, puis 11010, 32 et 1A sans préfixe. La casse des lettres ne change pas leur valeur.

Prévoir et vérifier une conversion

Avant de relancer, prévoyez les écritures de 15, 16 et 0 : F, 10 et 0 en base 16 ; 1111, 10000 et 0 en base 2. Convertissez puis reconstituez chaque valeur avec ses poids. 10_2, 10_8 et 10_{16} valent respectivement 2, 8 et 16 : annoncer la base évite une ambiguïté.

Changer de base sans changer de quantité

▼ **Que vaut 101101_2 ?**

32 + 8 + 4 + 1 = 45 en décimal. En regroupant les bits : 101 101 donne 55_8 ; 0010 1101 donne $2D_{16}$.

▼ Peut-on écrire 19 en base 8 ?

Le texte 19 est invalide dans cette base car 9 n'est pas un chiffre octal. La quantité dix-neuf s'écrit 23_8 : 2 × 8 + 3 = 19.

À retenir - fiche mémo

Base

En base b, les chiffres vont de 0 à b - 1 et les positions pèsent 1, b, b^2, etc.

Conversion

Divisez par la base et lisez les restes en sens inverse ; zéro s'écrit 0.

Regroupement

Un chiffre octal correspond à trois bits ; un chiffre hexadécimal à quatre bits.

Pseudo-code

```
// example.pseudo
// positional_value
value <- 1 * 16 + 10
PRINT value
```

Point de vigilance. Une écriture comme 10 exige d'annoncer sa base ; la valeur ne change pas lors d'une conversion.

Consulter la cheatsheet : Fondations · Télécharger la fiche du thème en PDF

Version web : https://www.clearcode.fr/courses/programming/algorithmic/foundations/number-bases