

# Les Principaux Systèmes de Numération en Informatique

Dans le domaine de l'informatique, plusieurs systèmes de numération sont utilisés pour représenter et manipuler les données. Les plus courants sont les systèmes décimal, binaire, hexadécimal et octal. Chacun de ces systèmes a une base différente, adaptée à des besoins spécifiques dans le contexte du traitement des données numériques. Dans cet article, nous allons explorer ces quatre systèmes de numération, en mettant en lumière leurs caractéristiques, leur utilité, et comment passer de l'un à l'autre.

## Le Système Décimal (Base 10)

Comme discuté précédemment, le système décimal est basé sur la base 10 et utilise les chiffres de 0 à 9. Ce système est le plus familier car il est utilisé dans la vie quotidienne pour la plupart des calculs. Cependant, en informatique, d'autres systèmes sont souvent plus efficaces pour certaines tâches.

### Exemple de Représentation

Le nombre 245 en système décimal est simplement écrit comme :

$$245 = 2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

## Le Système Binaire (Base 2)

Le système binaire est fondamental en informatique. Il est basé sur la base 2 et n'utilise que deux symboles : 0 et 1. Les ordinateurs fonctionnent principalement avec des signaux électriques, qui peuvent être facilement représentés par ces deux états (marche/arrêt, vrai/faux).

### Décomposition d'un Nombre Binaire

Prenons le nombre binaire 1011 :

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

Ce qui se traduit par :

$$1011_2 = 8 + 0 + 2 + 1 = 11_{10}$$

Ainsi, le nombre binaire 1011 équivaut au nombre décimal 11.

### Utilité en Informatique

Les systèmes binaires sont utilisés pour toutes les opérations de base dans les ordinateurs, car les circuits électroniques ne peuvent interpréter que deux états différents.

Toutes les données, y compris les images, le texte et les instructions de programme, sont finalement converties en binaire pour être traitées par un ordinateur.

## Le Système Hexadécimal (Base 16)

Le système hexadécimal est basé sur la **base 16**. Il utilise 16 symboles : les **chiffres 0 à 9** et les **lettres A à F** (où A représente 10, B représente 11, etc.). Ce système est couramment utilisé en informatique pour représenter les valeurs binaires de manière plus concise.

Le système hexadécimal utilise les symboles suivants : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**.

Le système hexadécimal, souvent abrégé en "**hexa**", est largement utilisé en informatique en raison de sa capacité à **représenter de grandes quantités de données** de manière compacte et lisible et pour sa **correspondance directe avec le binaire** : comme **chaque chiffre hexadécimal correspond exactement à une séquence de 4 bits**, il est facile de convertir entre le binaire et l'hexadécimal.

## Décomposition d'un Nombre Hexadécimal

Prenons le nombre hexadécimal 2F3 :

$$2F3_{16} = 2 \times 16^2 + F \times 16^1 + 3 \times 16^0$$

Comme F équivaut à 15 en décimal, cela donne :

$$2F3_{16} = 2 \times 256 + 15 \times 16 + 3 = 512 + 240 + 3 = 755_{10}$$

## Utilité en Informatique

Le système hexadécimal est souvent utilisé pour représenter les adresses mémoire et les couleurs dans les applications graphiques. Il est plus compact que le binaire, ce qui le rend plus facile à lire pour les humains tout en restant proche de la représentation binaire.

## Le Système Octal (Base 8)

Le système octal est basé sur la base 8 et utilise les chiffres de 0 à 7. Ce système était plus couramment utilisé dans les premiers jours de l'informatique, mais il est moins fréquent aujourd'hui. Cependant, il reste utile dans certains contextes, notamment dans la représentation compacte des nombres binaires.

## Décomposition d'un Nombre Octal

Prenons le nombre octal 345 :

$$345_8 = 3 \times 8^2 + 4 \times 8^1 + 5 \times 8^0$$

Ce qui donne :

$$345_8 = 192 + 32 + 5 = 229_{10}$$

## Utilité en Informatique

Le système octal est parfois utilisé dans les systèmes d'exploitation Unix pour représenter les permissions des fichiers, où chaque chiffre octal correspond à un ensemble de bits indiquant les permissions.

## Conversion entre les Systèmes

Si cela te paraît compliqué, garde en tête qu'il n'existe en fait que **3 formules** à retenir:

- Conversion base 10 --> une autre base: **somme des exposants**
- Conversion d'une autre base --> la base 10: **suite de divisions entières**
- Conversion binaire <----> l'hexadécimal: **groupes de 4**

## Conversion Binaire -> Décimal

Pour convertir un nombre binaire en décimal, on **multiplie chaque chiffre par la puissance de 2 correspondante** et on additionne les résultats. Par exemple, pour le nombre binaire 1010 :

$$1010_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 8 + 0 + 2 + 0 = 10_{10}$$

Cela me paraît plus simple de commencer par la droite ( $2^0$ )

## Conversion Décimal -> Binaire

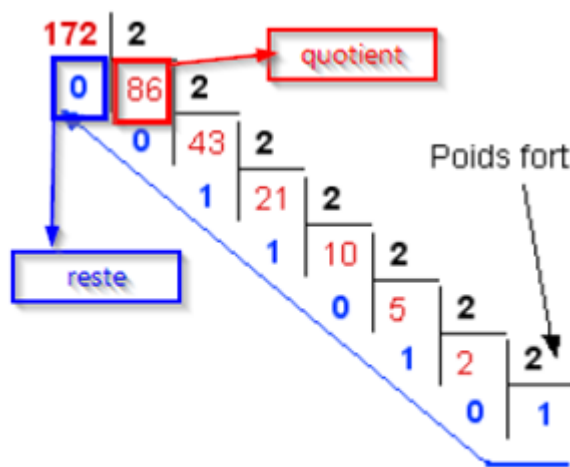
Pour convertir un nombre décimal en binaire, on divise le nombre par 2 et on note le reste. Il s'agit d'une **division entière**, comme celle que l'on a appris en primaire. Autrement dit, on divise par 2 et on ne garde que la partie entière (ex:  $7 / 2 = 3.5 \rightarrow 3$ )

On répète l'opération avec le quotient jusqu'à ce que le quotient soit 0. On note ensuite le résultat **en partant du bas: on note le dernier quotient et tous les restes**.

Par exemple, pour convertir 172 en binaire :

- $172 / 2 = 86$ , reste 0
- $86 / 2 = 43$ , reste 0
- $43 / 2 = 21$ , reste 1
- $21 / 2 = 10$ , reste 1
- $10 / 2 = 5$ , reste 0
- $5 / 2 = 2$ , reste 1
- $2 / 2 = 1$ , reste 0

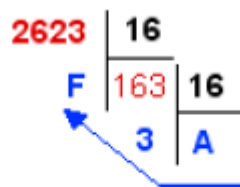
Le résultat en binaire est donc 101011100.



$$172 / 2 = 86, \text{ il reste } 0 \dots$$

## Conversion Décimal -> Hexa

La conversion du décimal -> hexadécimal fonctionne sur le même principe: une succession de divisions entières et de restes... sauf que dans ce cas, on divise par 16.



$$2623 / 16 = 163, \text{ il reste } 15 \dots$$

## Conversion Hexadécimal -> Binaire

Chaque chiffre hexadécimal peut être **directement converti en une séquence de 4 bits**. Par exemple, le nombre hexadécimal 3A se convertit en binaire :

- 3 -> 0011
- A -> 1010 (car A = 10 en décimal)

Ainsi, 3A en hexadécimal devient 00111010 en binaire.

Tu peux utiliser la table de correspondance suivante :

Binaire	Hexadécimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F



Il existe une astuce pour construire cette table rapidement et sans se tromper. Ecoute bien en classe!

## Conversion Binaire -> Hexadécimal

La conversion d'un nombre binaire en hexadécimal est un processus simple et direct, car il existe une correspondance étroite entre ces deux systèmes de numération. **Chaque chiffre hexadécimal représente exactement quatre bits** (une séquence de quatre chiffres binaires). Pour convertir un nombre binaire en hexadécimal, suivez ces étapes :

1. **Diviser le nombre binaire en groupes de 4 bits** : Commencez par diviser le nombre binaire en groupes de **4 bits, en partant de la droite**. Si le nombre de bits n'est pas un multiple de 4, ajoutez des zéros à gauche pour compléter le dernier groupe.

Par exemple, pour convertir le nombre binaire 1011101010 en hexadécimal, divisez-le en groupes de 4 bits :

10 1110 1010

En ajoutant des zéros pour compléter les groupes, cela devient :

0010 1110 1010

2. **Convertir chaque groupe de 4 bits en hexadécimal** : Chaque groupe de 4 bits est ensuite converti en son équivalent hexadécimal, en utilisant la table de correspondance du paragraphe précédent.

Pour notre exemple :

- 0010 devient 2
- 1110 devient E
- 1010 devient A

3. **Assembler les chiffres hexadécimaux** : Assemblez les chiffres hexadécimaux obtenus pour former le nombre hexadécimal final. Dans notre exemple, **1011101010** en binaire se convertit en **2EA** en hexadécimal.

Cette méthode est rapide et efficace, car elle tire parti de la correspondance directe entre les bits binaires et les chiffres hexadécimaux. C'est pourquoi la conversion binaire vers hexadécimal est couramment utilisée pour représenter les valeurs binaires sous une forme plus compacte et lisible en informatique.

## Conclusion

Les systèmes de numération binaire, octal et hexadécimal sont essentiels en informatique, chacun ayant sa propre utilité et domaine d'application. Le système binaire est fondamental car il correspond directement à la manière dont les ordinateurs traitent l'information. Le système hexadécimal est utilisé pour sa compacité et sa proximité avec le binaire, tandis que le système octal est utilisé dans des contextes spécifiques comme la gestion des permissions dans les systèmes Unix. Comprendre ces systèmes de numération permet de mieux appréhender le fonctionnement interne des ordinateurs et d'améliorer ses compétences en programmation et en manipulation de données.

## Quelques exercices

Pour vous aider à maîtriser les conversions entre les systèmes de numération binaire, décimal, et hexadécimal, voici cinq exercices progressifs pour chaque type de conversion. Ces exercices te permettront de pratiquer et de renforcer ta compréhension de ces systèmes fondamentaux.

### Exercice 1 : Conversion de binaire en décimal

1. Convertis le nombre binaire **1011** en décimal.
2. Convertis le nombre binaire **1101** en décimal.
3. Convertis le nombre binaire **100100** en décimal.
4. Convertis le nombre binaire **1110** en décimal.
5. Convertis le nombre binaire **101010** en décimal.

### Exercice 2 : Conversion de décimal en binaire

1. Convertis le nombre décimal **13** en binaire.
2. Convertis le nombre décimal **9** en binaire.

3. Convertis le nombre décimal 31 en binaire.
4. Convertis le nombre décimal 20 en binaire.
5. Convertis le nombre décimal 45 en binaire.

### Exercice 3 : Conversion de hexadécimal en décimal

1. Convertis le nombre hexadécimal 1A en décimal.
2. Convertis le nombre hexadécimal 2F en décimal.
3. Convertis le nombre hexadécimal 10 en décimal.
4. Convertis le nombre hexadécimal B4 en décimal.
5. Convertis le nombre hexadécimal 7E en décimal.

### Exercice 4 : Conversion de décimal en hexadécimal

1. Convertis le nombre décimal 26 en hexadécimal.
2. Convertis le nombre décimal 47 en hexadécimal.
3. Convertis le nombre décimal 16 en hexadécimal.
4. Convertis le nombre décimal 180 en hexadécimal.
5. Convertis le nombre décimal 126 en hexadécimal.

### Exercice 5 : Conversion de binaire en hexadécimal et vice-versa

1. Convertis le nombre binaire 101110 en hexadécimal.
2. Convertis le nombre binaire 1111 en hexadécimal.
3. Convertis le nombre hexadécimal 1C en binaire.
4. Convertis le nombre hexadécimal 3A en binaire.
5. Convertis le nombre binaire 10011011 en hexadécimal.