



Fondements de l'Informatique

Guermah Bassma ---- bassma.guermah@uir.ac.ma

Année scolaire : 2023/2024

Chapitre 1

Codage et Traitement de l'Information



Rappel : Chapitre 1 - Ce qu'il faudra retenir

➤ **Qu'est-ce que l'informatique ? Son évolution, ses perspectives.**

➤ **Traitement et codage des données :**

- Traitement de l'information
- Codage en général : le pourquoi
- Codage des nombres entiers
- Codage des nombres réels
- Codage des caractères : code ASCII
- Codage des caractères : code Unicode

➤ **Numération :**

- Les bases décimale, binaire et hexadécimale
- Conversion base quelconque -> Décimal
- Conversion Décimal -> Binaire
- Conversion Binaire -> Décimal
- Conversion Binaire -> Hexadécimal
- Conversion Hexadécimal -> Binaire



Partie 2

Traitement et codage des données

Partie 2 - Traitement et codage des données

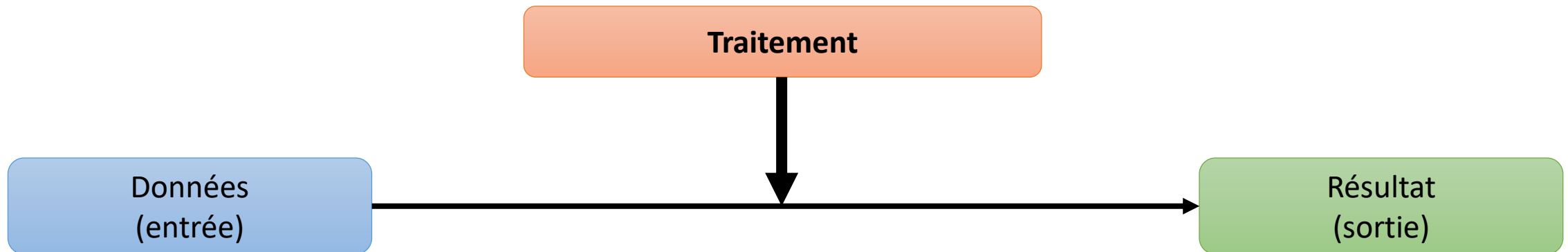
- Traitement de l'information
- Codage en général : le pourquoi
- Codage des nombres entiers naturels et relatifs
- Codage des nombres réels
- Codage des caractères : code ASCII
- Codage des caractères : code Unicode

Traitement de l'Information

Rappel :

- ❑ L'information est le support formel d'un élément de connaissance humaine susceptible d'être représentée à l'aide de conventions (codages) afin d'être conservée, traitée ou communiquée.
- ❑ Une donnée est la représentation d'une information sous une forme conventionnelle (codée) destinée à faciliter son traitement.

Schéma simplifié du traitement de l'information :



Traitement de l'Information – Définitions

- ❑ Une série de processus qui permettent d'extraire de l'information, de produire du savoir et des résultats à partir de données brutes.
- ❑ Permet de décrire parfaitement les données d'entrée et les résultats de sortie.
- ❑ Permet de décomposer le passage des données d'entrée vers des résultats en une suite d'opérations élémentaires.

Codage en général : le pourquoi

Langage Machine

Toutes les informations sont codées sous forme d'une suite de '0' et de '1' (langage binaire)



Traduire

Langage humain

Codage

Opération établissant une bijection entre une information (écrite en langage humain) et une suite de '0' et de '1' qui sont représentable en machine.

Vocabulaire

- ❑ Le terme bit signifie « binary digit », c'est-à-dire 0 ou 1.
- ❑ Le bit c'est la plus petite unité d'information manipulable par une machine numérique.
- ❑ L'octet « Byte » est une unité d'information composée de 8 bits.
- ❑ Une unité d'information composée de 16 bits est généralement appelée mot (en anglais word).
- ❑ Une unité d'information de 32 bits de longueur est appelée mot double (en anglais double word).
- ❑ les unités standardisées :

Un kilooctet (**ko**) = 10^3 octets

Un mégaoctet (**Mo**) = 10^6 octets

Un gigaoctet (**Go**) = 10^9 octets

Un téraoctet (**To**) = 10^{12} octets

Un pétaoctet (**Po**) = 10^{15} octets

Un exaoctet (**Eo**) = 10^{18} octets

Un zettaoctet (**Zo**) = 10^{21} octets

Un yottaoctet (**Yo**) = 10^{24} octets

Codage des nombres entiers

Soit $(x_n x_{n-1} \dots x_0)_b$ un nombre entier **X** écrit en base b avec $n+1$ chiffres :

$$X = x_n \times b^n + x_{n-1} \times b^{n-1} + \dots + x_0 \times b^0$$

- « x_k » est le chiffre associé à la puissance « b^k ».
- « x_k » $\in \{0, 1, \dots, b-1\}$

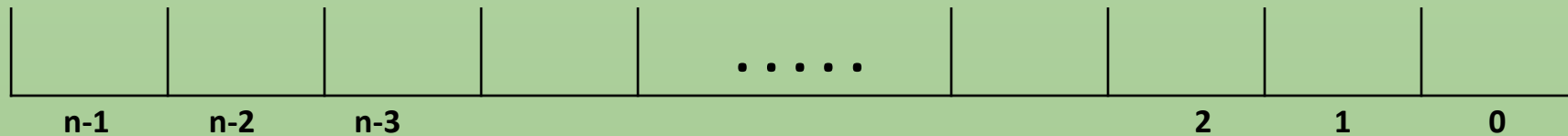
Exemple :

Soit le chiffre 2149 en base décimal (base 10), la position des chiffres 2, 1, 4, 9 indique la puissance de 10 à laquelle ils sont associés :

2	Associée	10^3
1		10^2
4		10^1
9		10^0

Codage des nombres entiers – base binaire

- En codage binaire, chaque chiffre du nombre X , « x_k » $\in \{0, 1\}$.
- Les nombres binaires sont écrits avec des 0 et des 1, qui sont représentés par des bits dans la machine.
- Soit le schéma d'une mémoire à n bits :



- Les cases du schéma représentent les bits,
- Le chiffre marqué représente le rang du bit.
- Le bit de rang 0 est appelé le bit de **poids faible**.
- Le bit de rang $n-1$ est appelé le bit de **poids fort**.

Codage des nombres entiers naturels

- Un entier naturel est un nombre entier positif ou nul.
- Le nombre de bits à utiliser dépend de la fourchette des nombres que l'on désire utiliser.
- Un codage sur n bits pourra permettre de représenter des nombres entiers naturels compris entre 0 et $2^n - 1$, soit l'intervalle $[0, 2^n - 1]$.

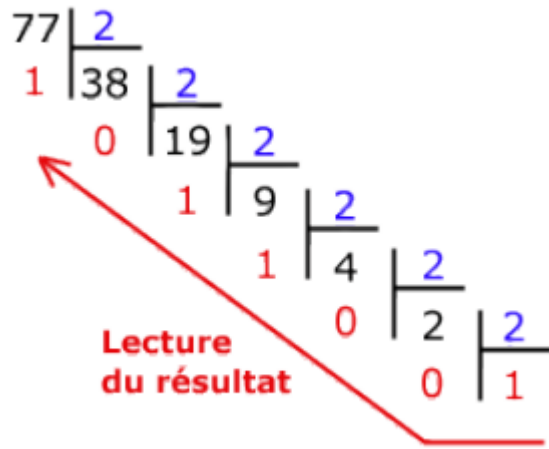
Exercice :

Combien de bits on aura besoin pour coder des nombres entiers naturels compris entre 0 et 255

Conversion base Décimal -> Binaire

Cas : entiers naturels

- La conversion d'un entier naturel écrit en décimal vers le binaire se fait par la division **Euclidienne** sur **2**.
- Exemple :**



77 s'écrit donc en base 2 : 1001101

Codage des nombres entiers naturels

- Exemples :

$$\begin{aligned}9 &= 1001_2 \\128 &= 10000000_2 \\77 &= 1001101_2\end{aligned}$$

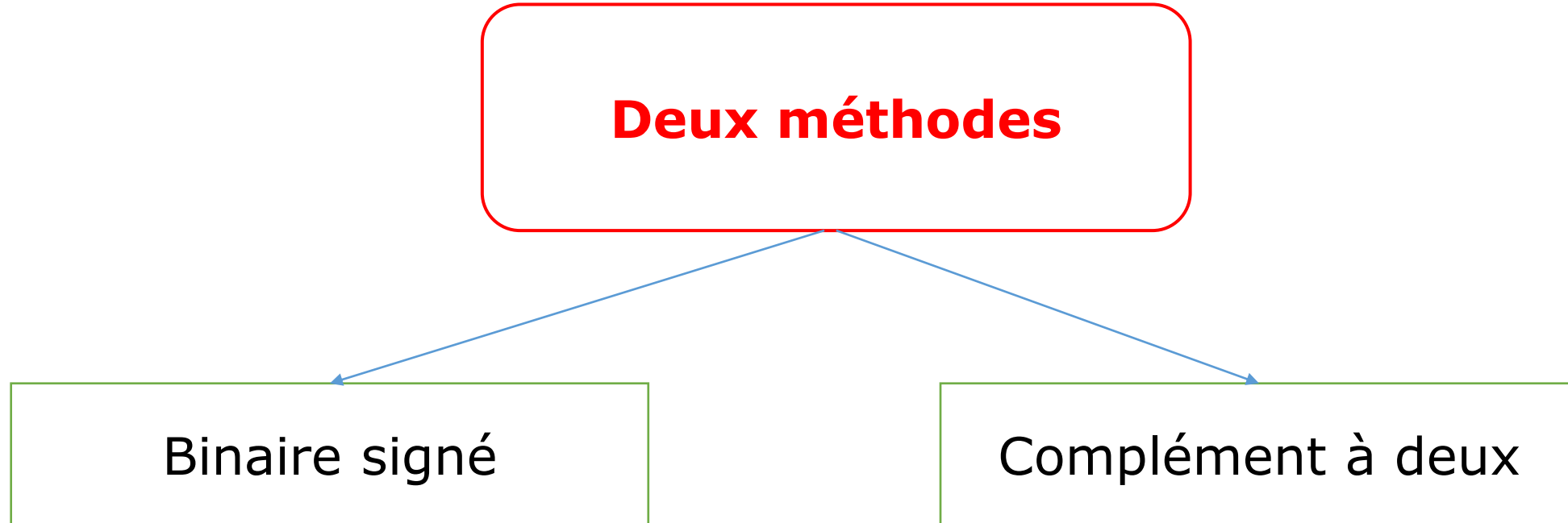
- Exercices :

Présenter les nombres entiers naturels suivants en binaire :

6, 100, 15, 255

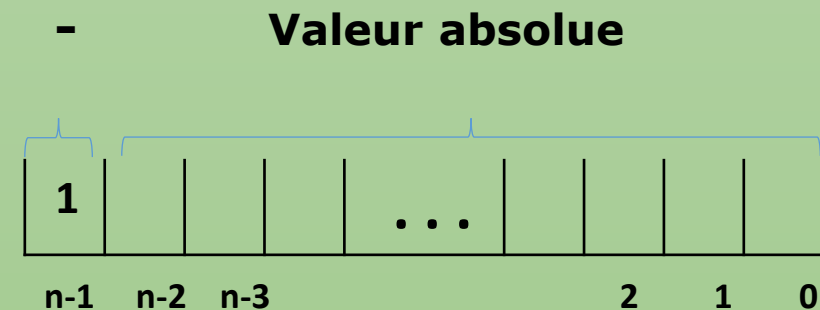
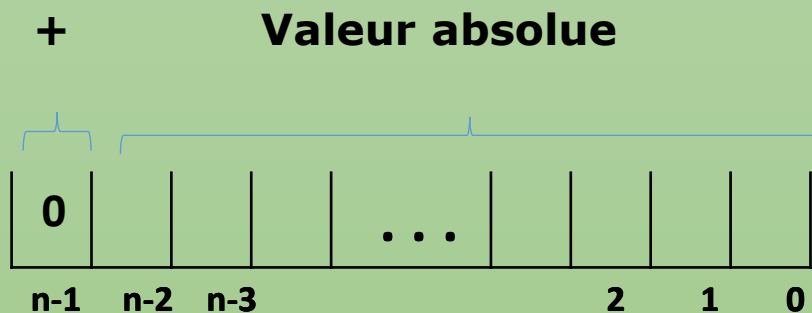
Codage des nombres entiers relatifs

-- Un entier relatif est un entier pouvant être négatif.



Codage des nombres entiers relatifs – binaire signé

- Dans la représentation des entiers relatifs en binaire signé, le bit de poids fort (*bit de rang $n-1$ associé à 2^{n-1}*) sert à représenter le signe (**0** pour un entier positif et **1** pour un entier négatif), les **$n-1$** autres bits représentent la valeur absolue du nombre en binaire.



Exemple : (le codage de 14 et -14)

14 = + 14 est représenté par : **0**000 ... 01110

- 14 est représenté par : **1**000 ... 01110

Codage des nombres entiers relatifs – binaire signé

- Un codage sur n bits pourra permettre de représenter des nombres entiers relatifs compris entre -2^{n-1} et $2^{n-1} - 1$, soit l'intervalle $[-2^{n-1}, 2^{n-1} - 1]$.

Exemples :

- ☐ Soit pour $n = 8$ bits (1 octet), tous les entiers de l'intervalle $[-128, 127]$.
- ☐ Soit pour $n = 16$ bits (2 octets), tous les entiers de l'intervalle $[-32768, 32767]$.
- ☐ Soit pour $n = 32$ bits (4 octets), tous les entiers de l'intervalle $[-2147483648, 2147483647]$.

Codage des nombres entiers

Exercice :

Question 1 : Quel est le plus grand entier que l'on peut coder sur 8 bits ? 32 bits ?

Question 2 : Codez les entiers relatifs suivants sur 8 bits (16 si nécessaire) en utilisant la méthode de binaire signé :

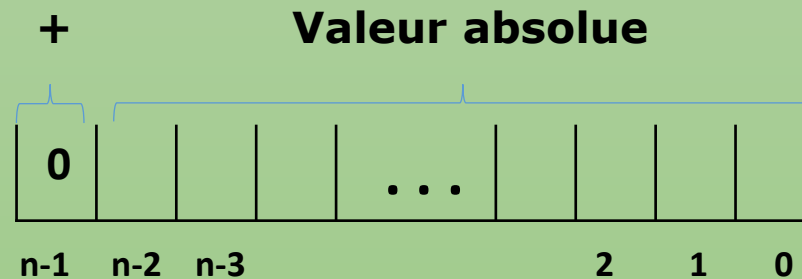
-19, 456, 1, -6, 72, 600, -187

Codage des nombres entiers relatifs – Complément à deux

Supposons que la mémoire soit à n bits, soit x un entier relatif à représenter :

- **Cas 1** : $x > 0$

La convention précédente s'applique (le bit de signe vaut 0, les $n-1$ bits restants codent le nombre), soit le schéma général suivant :



Codage des nombres entiers relatifs – Complément à deux

▪ Cas 2 : $x < 0$

- Etape 1 : Écrire la valeur absolue du nombre en base 2. **Le bit de poids fort doit être égal à 0**
- Etape 2 : Inverser les bits : les 0 deviennent des 1 et vice versa. On fait ce qu'on appelle le **complément à un**.
- Etape 3 : On ajoute 1 au résultat (addition binaire).

Exemple : Obtenir -7 => codage sur 4 bits

Etape 1 : 0111 ,

Etape 2 : 1000 (complément à 2) ,

Etape 3 : 1001 (ajoutant 1 au complément à 2).

Codage des nombres entiers relatifs – Complément à deux

- **Astuce :**

Pour transformer un nombre binaire en son complément à deux, on parcourt le nombre de droite à gauche en laissant inchangés les bits jusqu'au premier 1 (compris), puis on inverse tous les bits suivants.

Exemple : le nombre 20

Etape 1 : le nombre 20 s'écrit en binaire (sur 8 bits) : 00010100

Etape 2 : On garde la partie à droite telle quelle : 00010**100**

Etape 3 : On inverse la partie de gauche après le premier un : **1110**1100

Etape 4 : Et voici -20 : 11101100

Codage des nombres entiers

Exercice à rendre :

Question 1 : Codez les mêmes entiers relatifs de l'exercice précédent en utilisant la représentation en « complément à deux ».

Question 2 : Transformez en base 10 les entiers naturels suivants :

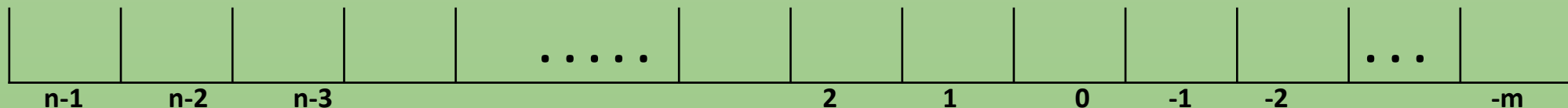
$$(111001)_2 ; (00000001)_2 ; (1010101)_2$$

Codage des nombres réels – Virgule fixe

- Un nombre réel est composé d'une partie entière et d'une partie fractionnaire après la virgule.
- Soit $(x_n x_{n-1} \dots x_0, y_1 y_2 \dots y_m)_b$ un nombre réel **X** écrit en base **b** avec **n+m+1** chiffres :

$$X = x_n \times b^n + x_{n-1} \times b^{n-1} + \dots + x_0 \times b^0 + y_1 \times b^{-1} + \dots + y_m \times b^{-m}$$

- « x_k » est le chiffre associé à la puissance « b^k ».
- « y_k » est le chiffre associé à la puissance « b^{-k} ».
- « x_k, y_k » $\in \{0, 1, \dots, b-1\}$
- Soit le schéma d'une mémoire à **n+m** bits :



- Les cases du schéma représentent les bits,
- Le chiffre marqué représente le rang du bit.

Codage des nombres réels

Exemples :

- $(128,75)_{10} = 1 \times 10^2 + 2 \times 10^1 + 8 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}.$
- $(101,01)_2 = 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-2} = 1 \times 4 + 1 + 0,25 = 5,25$

Exercice à rendre :

- ✓ Q1 : Décomposez les nombres suivants : 652,375 ; 18,023 ; 0,180
- ✓ Q2 : Transformez en base 10 les nombres suivants :

$$(11100,10001)_2 ; (110,101)_2 ; (0,01010101)_2$$

Questions ?

Merci de votre attention!

