

CHAPITRE I : INTRODUCTION A L'INFORMATIQUE

I.1. Définition de l'Informatique

L'informatique est la **science** qui s'occupe du **traitement automatique** de l'**information**. Le terme "*informatique*" vient de la contraction des mots "*information*" et "*automatique*".

- **Rôle :**

L'informatique a pour rôle :

- La conception et la construction des ordinateurs;
- Le fonctionnement et la maintenance des ordinateurs;
- Leur exploitation (utilisation des ordinateurs dans les différents domaines d'activités).

I.2. Evolution de l'informatique et des ordinateurs

L'histoire de l'informatique est justement marquée par la volonté des hommes d'automatiser certaines tâches longtemps réalisées à la main, en particulier le calcul.

Date	Inventeur	Machine	Fonction
1642	Blaise Pascal	Pascaline	Effectue des additions et soustractions (des nombres de six chiffres)
1822-1832	Charles Babbage	Machine à Différences (Difference Engine)	Effectue des calculs numériques compliqués (Tables numériques).
1887	Hermann Hollerith	Machine mécanographique (Calculateur de statistiques à cartes perforées)	Accélérer le traitement des données lors du recensement américain de 1890 (invention de la carte perforée).
1945 (inauguré en 1946)	John Presper Eckert & John William Mauchly	"ENIAC" Electronic Numerical Integrator and Computer	Premier calculateur numérique électronique programmable. 18.000 tubes, 30 tonnes. Multiplication de deux nombre de 10 chiffres signés en moins de 3 millisecondes.
	John Von Neumann	"EDVAC" Electronic Discrete Variable Automatic Computer	Enregistrer le programme en mémoire.

I.2.1. Définition de l'ordinateur : Un ordinateur est une **machine automatique programmable**, utilisée pour le **traitement** de l'**information**.

I.2.2. Générations des ordinateurs :

- **1^{ère} Génération (1945 - 1957) :**

C'est les ordinateurs construits sur la base des "*tubes électroniques*" (aussi appelés "*tubes à vide*").
Exemple: ENIAC (1945) et IBM 701 (1952).

- **2^{ème} Génération (1958 - 1963) :**

Elle est définie par l'invention du "*Transistor*" en 1947 (le transistor est un composant électronique capable de réguler le courant). Ces ordinateurs sont 100 plus rapides que ceux de la 1ère génération et consomment moins d'énergie électrique et sont moins volumineux (occupent moins d'espace).
Exemples: PDP I (1er ordinateur de 2ème génération (1960)), IBM 7030 (1961).

- **3^{ème} Génération (1964-1971) :**

Elle est définie après l'apparition du "*Circuit Intégré*" en 1958, les ordinateurs de cette génération utilisent les transistors et les circuits intégrés. **Exemple:** IBM 360 (1964), DEC PDP-8 (1964).

- **4^{ème} Génération (1971 - 1980) :**

Elle survient après l'invention du "*microprocesseur*". Le premier microprocesseur est fabriqué par la société INTEL en 1971. On parle, désormais, de micro-ordinateurs. **Exemple:** MICRAL 8008 (1973), ALTAIR 8800 (1975).

- **5^{ème} Génération (1980 - 2000) :**

Les ordinateurs entrent dans les foyers, on parle alors d'informatique familiale (c'est la naissance des ordinateurs personnels (PC)). **Exemple :** IBM PC (1981), Intel Pentium (1993).
Aussi, les systèmes d'exploitation MS-DOS et Mac OS font leur apparition.

- **6^{ème} Génération (2000 – aujourd'hui) :**

On assiste, dans cette génération, à l'apparition des ordinateurs portables et des réseaux sans fil.

Actuellement l'évolution des ordinateurs tend à exploiter le laser, les fibres optiques, la biochimie et même l'informatique moléculaire.

I.3. Architecture et fonctionnement d'un ordinateur :

La Figure 1 représente l'architecture générale d'un ordinateur :

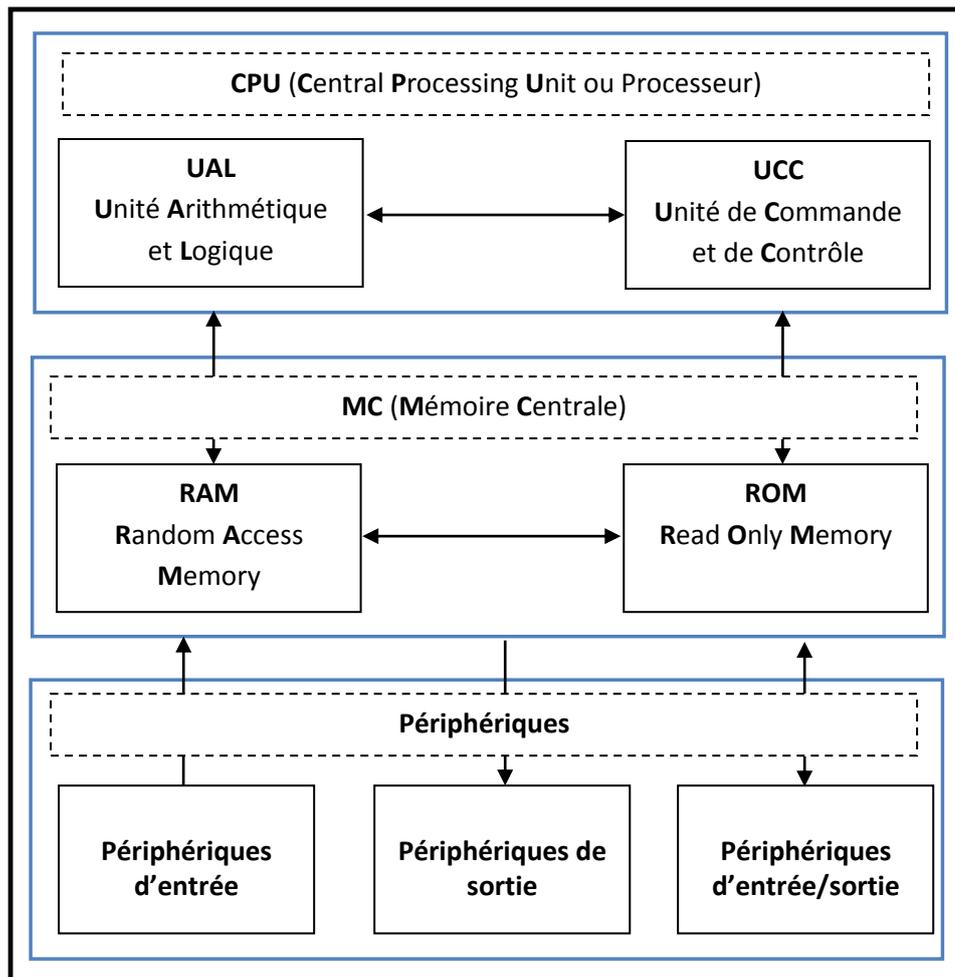


Figure 1 Architecture d'un ordinateur

Un ordinateur est constitué, d'une manière générale, d'une Unité Centrale de Traitement, d'une mémoire centrale et d'un ensemble de périphériques.

a. **L'Unité Centrale de Traitement (UCT/CPU) ou bien Processeur (Microprocesseur) :**
Le processeur est le cœur de l'ordinateur, chargé de traiter et d'exécuter les programmes (les instructions) de l'ordinateur. Il comporte, essentiellement, deux unités : l'Unité de Commande et de Contrôle (UCC) et l'Unité Arithmétique et Logique (UAL).

- i. **L'unité de Commande et de Contrôle (UCC) :** Elle est responsable de la lecture en mémoire centrale et du décodage des instructions ;
- ii. **L'unité Arithmétique et Logique (UAL) :** Aussi appelée unité de traitement, cette dernière exécute les instructions. Elle se charge d'effectuer toutes les opérations arithmétiques, logiques et relationnelles contenues dans l'instruction et effectue aussi des échanges de données (réception/transmission) avec la mémoire vive.

Ces deux unités communiquent, non seulement, avec la mémoire centrale, mais également avec les différents périphériques.

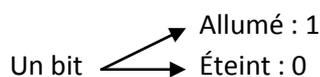
b. **La Mémoire Centrale (ou Principale) :** Elle est constituée d'une mémoire vive (RAM : Random Access Memory) et d'une mémoire morte (ROM : Read Only Memory).

- i. **La RAM** : Une RAM ou mémoire vive sert au stockage temporaire des données et du programme à exécuter. Les mémoires vives sont en général volatiles : elles perdent leurs informations en cas de coupure du courant électrique.
 - ii. **La ROM** : Une ROM ou mémoire morte, par opposition à la RAM, ne s'efface pas à la coupure du courant. Elle peut être lue mais pas (ou peu de fois) écrite. Elle conserve des programmes nécessaires au fonctionnement du matériel, surtout lors du démarrage (avant le chargement du système d'exploitation dans la RAM).
- c. **Les périphériques** : Les périphériques sont composés de périphériques d'entrée, de sortie et des périphériques d'entrée/sortie.
- i. **Périphériques d'entrée** : C'est l'ensemble de périphériques qui permet de transmettre des données à l'ordinateur : Le clavier, la souris, le microphone, la webcam, etc.).
 - ii. **Périphériques de sortie** : C'est l'ensemble de périphériques qui permet de recevoir des données de l'ordinateur et de les renvoyer vers l'extérieur : L'écran, l'imprimante, les haut-parleurs, etc.).
 - iii. **Périphériques d'entrée/sortie** : C'est l'ensemble de périphériques qui permet, à la fois, de transmettre et de recevoir des données. Ils sont, également, appelés périphériques de stockage ou mémoires externes (mémoires auxiliaires) : La disquette, le CD-ROM, la clé USB, le disque dur externe, etc.).
- ✓ **Remarque** : Tous les composants de l'ordinateur, cités ci-dessus, communiquent entre eux grâce à des liaisons électriques appelées **Bus**.

I.4. Les systèmes de codage de l'information

Toute information (numérique, textuelle, image, vidéo, son,.. etc.) manipulée par un ordinateur, est représentée par des séquences de deux chiffres : **0** et **1**, ces derniers sont désignés par **BIT (Binary digiT)**.

En informatique, il y a deux états possibles : il y a du courant électrique ou il n'y a pas de courant électrique, c'est la seule information qu'un ordinateur puisse comprendre. Cet état est déterminé par le **bit (C'est la plus petite unité qui existe en informatique)**. Par conséquence :



I.4.1. Les systèmes de numérotation :

- **Définition** : C'est la représentation d'un entier naturel **N** en une base **b**. Ainsi, un système de numérotation se définit par deux éléments :
 1. La base du système (**b**);
 2. Les symboles du système (l'ensemble des chiffres et des lettres qui représentent la base).

En informatique, les systèmes les plus utilisés sont :

Système	Base	Symboles	Nombre de symboles du système
Décimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	10
Binaire	2	0, 1	2
Octal	8	0, 1, 2, 3, 4, 5, 6, 7	8
Hexadécimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	16

Avec : $(A)_{16}=(10)_{10}$, $(B)_{16}=(11)_{10}$, $(C)_{16}=(12)_{10}$, $(D)_{16}=(13)_{10}$, $(E)_{16}=(14)_{10}$, $(F)_{16}=(15)_{10}$.

✓ **Remarque** : Le nombre de symboles d'un système = b.

• **Notation** : Soit N un nombre entier naturel exprimé en une base b. N est noté comme suit :

$$N = (a_{n-1}, a_{n-2}, a_{n-3}, \dots, a_0)_b \tag{1}$$

Avec :

a_i : Symbole du système

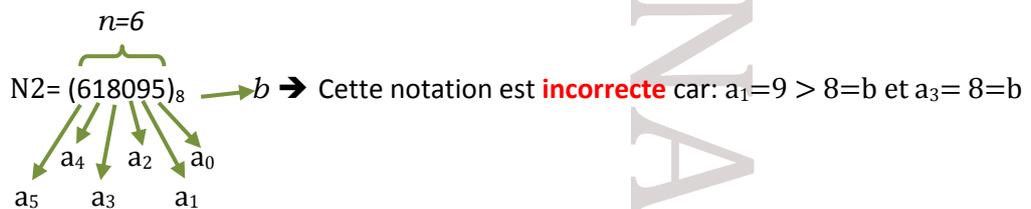
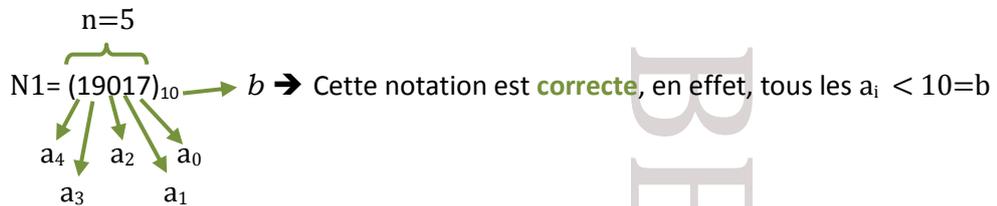
n : Le nombre de symboles qui constituent N.

tel que :

$a_i = 0, 1, 2, \dots, n-1$

$a_i < b$

▪ **Exemples**:



Ainsi, pour exprimer N en fonction d'une base b, il suffit de multiplier chaque a_i par b et faire la somme.

$$(N)_b = (a_{n-1} * b^{n-1} + a_{n-2} * b^{n-2} + a_{n-3} * b^{n-3} + \dots + a_0 * b^0) \tag{2}$$

Ou

$$(N)_b = \sum_{i=0}^{n-1} a_i * b^i \tag{2}$$

On appelle i le poids du symbole a_i

- **Exemple:** Reprenons l'exemple précédent, la représentation de N1 en fonction de sa base est :

$$N1 = (\overset{a_4}{1} \overset{a_3}{9} \overset{a_2}{0} \overset{a_1}{1} \overset{a_0}{7})_{10} = 7 \cdot 10^0 + 1 \cdot 10^1 + 0 \cdot 10^2 + 9 \cdot 10^3 + 1 \cdot 10^4$$

- ✓ **Remarque :** Quand la base n'est pas mentionnée, on considère qu'on est en base 10.

I.4.2. Conversion d'un nombre N en une base b :

- b. **Conversion en décimal (b → 10) :** Pour avoir la représentation en décimal (base 10), du nombre N exprimé dans une base b quelconque, il suffit d'appliquer la formule (2) citée ci-dessus.

- **Exemples:**

Soit $N = (1011100)_2$ $n=7$ $b \rightarrow$ D'abord, identifier la base et les symboles.
 Ensuite, faire les calculs.

$$N = 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 = 4 + 8 + 16 + 64 = (92)_{10}$$

$$\text{Soit } N1 = (\overset{a_3}{3} \overset{a_2}{1} \overset{a_1}{7} \overset{a_0}{5})_8 = 5 \cdot 8^0 + 7 \cdot 8^1 + 1 \cdot 8^2 + 3 \cdot 8^3 = 5 + 56 + 64 + 1536 = (1661)_{10}$$

$$\text{Soit } N2 = (\overset{a_2}{A} \overset{a_1}{2} \overset{a_0}{4})_{16} = 4 \cdot 16^0 + 2 \cdot 16^1 + A \cdot 16^2 = 4 + 32 + 2560 = (2596)_{10}$$

- c. **Conversion à partir du décimal (10 → b) :** Pour avoir la représentation en base b, d'un nombre N exprimé en décimal (base 10), il suffit d'effectuer des divisions entières successives sur b, jusqu'à obtention d'un résultat nul, autrement dit :

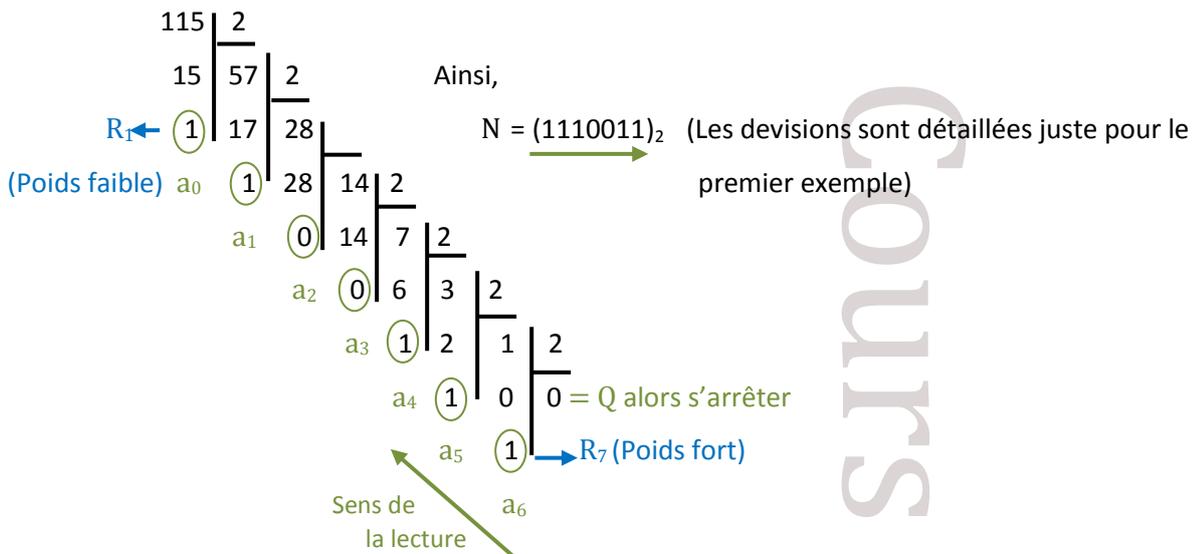
- Effectuer une division entière de N sur b ($N \text{ div } b = Q$; le quotient) ;
- Si ($Q > 0$), aller à i ;
- Si ($Q = 0$) alors s'arrêter. $(N)_b =$ La concaténation de tous les restes R en commençant par le dernier. ($R = \{R_1, R_2, \dots, R_j\}$, j est le nombre de division)

- ✓ **Remarques :**

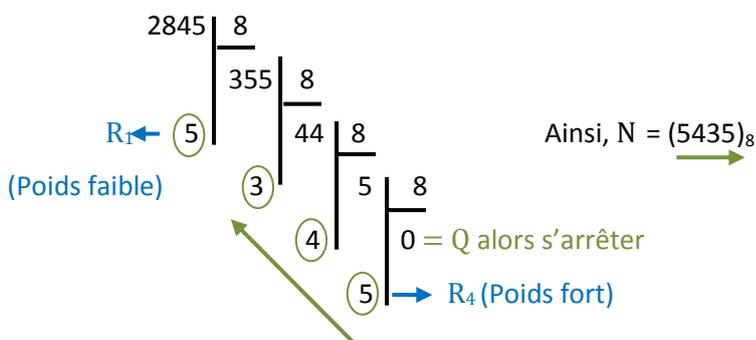
- Le **premier** reste (R_1) est appelé "**poids faible**" (le reste le plus à gauche dans la division), ce reste représente a_0 dans la nouvelle base.
- Le **dernier** reste (R_j) est appelé "**poids fort**" (le reste le plus à droite dans la division), ce reste représente a_{n-1} dans la nouvelle base.

- **Exemples :**

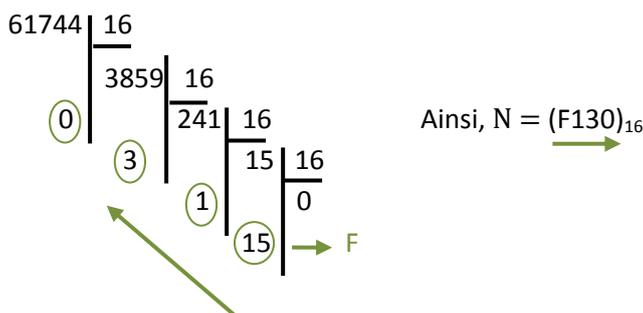
$$\text{Soit } N = (115)_{10} \rightarrow N = (?)_2$$



Soit $N = (2845)_{10} \rightarrow N = (?)_8$



Soit $N = (61744)_{10} \rightarrow N = (?)_{16}$



❖ **Important :**

Le passage (la conversion) d'une base b_1 à une base b_2 , se fait à travers la base 10 (Il faut donc passer par le décimal et faire $b_1 \rightarrow 10$ ensuite $10 \rightarrow b_2$).

✚ **Cas particulier :**

Lorsqu'une base est une puissance d'une autre base, la conversion ne nécessite pas le passage par la base 10.

▪ **Exemples :**

1) Soit $N = (7302)_8 \rightarrow N = (?)_2$

On a 8 (b1) est une puissance de 2 (b2) :

$\rightarrow 8=2^3$

\rightarrow Chaque symbole de la base 8 s'écrit sur trois (3) positions en base 2 (On convertit chaque chiffre octal en un nombre binaire puis on complète avec des zéros pour les nombres qui s'écrivent sur moins de 3 positions).

$\rightarrow N = (7302)_8 = (111011000010)_2$

2) Soit $N = (B14)_{16} \rightarrow N = (?)_2$

$\rightarrow 16=2^4$

\rightarrow Chaque symbole de la base 16 s'écrit sur quatre (4) positions en base 2 (On convertit chaque symbole hexadécimal en un nombre binaire puis on complète avec des zéros pour les nombres qui s'écrivent sur moins de 4 positions).

$\rightarrow N = (\overset{\text{B}}{\color{red}} \overset{1}{\color{blue}} \overset{4}{\color{green}})_{16}$
 $= (101100010100)_2$

3) Soit $N = (10011101010)_2 \rightarrow N = (?)_{16}$

$\rightarrow 2=2^{4/4}$

\rightarrow Quatre symboles de la base 2 représentent un symbole en base 16 (On regroupe par quatre (4) chiffres binaires en commençant par la droite et on convertit à la base 16).

$\rightarrow N = (010011101010)_2$
 $= (\underset{\text{4}}{\color{green}} \underset{\text{14}}{\color{orange}} \underset{\text{10}}{\color{purple}})_{16} = (4EA)_{16}$

4) Soit $N = (56723)_8 \rightarrow N = (?)_{16}$

\rightarrow Pour cette conversion, il est correct, de passer par la base 10. Toutefois, il existe une méthode beaucoup plus simple, qui est le passage par la base 2 (car $8=2^3$ et $16=2^4$, donc la base 2 est une base intermédiaire).

$N = (56723)_8 = (0101110111010011)_2$
 $= (\underset{\text{5}}{\color{red}} \underset{\text{13}}{\color{orange}} \underset{\text{13}}{\color{green}} \underset{\text{3}}{\color{blue}}) = (5DD3)_{16}$

d. **Le code DCB (Decimal Coded Binary) :** Consiste à convertir chaque symbole (chiffre) décimal en un nombre binaire sur 4 positions.

▪ **Exemple:** Soit $N = (329)_{10} \rightarrow N = (329)_{10} = (001100101001)_{\text{DCB}}$

I.5. Les expressions arithmétiques et logiques

I.5.1. Expressions arithmétiques : C'est une combinaison d'opérande(s) (valeur, constante, variable) et d'opération(s) arithmétiques. Le tableau 1 résume ces opérations.

Tableau 1

Opération	Symbole
Addition	+
Soustraction	-
Multiplication	*
Division réelle	/
Division entière	Div
Modulo (reste de la division entière)	Mod

Priorité dans les expressions arithmétiques :

1. ()
2. *, /, Div, Mod
3. +, -

} Applicables sur des entiers.

✓ **Remarque :**

La division entière donne comme résultat un quotient entier tandis que la division réelle donne un quotient réel. Le Mod, quant à lui, est le modulo, c.à.d. le reste de la division entière.

$$10 \text{ Div } 4 = 2 \text{ tandis que } 10 / 4 = 2.5 \quad \rightarrow \quad 10 \text{ Mod } 4 = 2$$

$$20 \text{ Div } 6 = 3 \text{ tandis que } 20 / 6 = 3.33 \quad \rightarrow \quad 20 \text{ Mod } 6 = 2$$

$$6 \text{ Div } 8 = 0 \text{ tandis que } 6 / 8 = 0.75 \quad \rightarrow \quad 6 \text{ Mod } 8 = 6$$

I.5.2. Expressions relationnelles : C'est les expressions qui utilisent des opérateurs relationnels (ou de comparaison). Le tableau 2 résume ces opérateurs.

Tableau 2

Relation	Symbole
Égalité	=
Inférieur	<
Inférieur ou égal	<=
Supérieur	>
Supérieur ou égal	>=
Différent	<>

I.5.3. Expressions logiques (booléennes) : C'est une combinaison de variables de type booléen (**True** : Vrai ou **False** : Faux) et d'opérateurs booléens (**Not** : Non, **And** : Et, **Or** : Ou). Le tableau 3 représente la table de vérité.

Tableau 3

A	B	And	Or	Not A	Not B
True	True	True	True	False	False
True	False	False	True	False	True
False	True	False	True	True	False
False	False	False	False	True	True

Priorité dans les expressions logiques :

1. ()
2. Not
3. And
4. Or

I.5.4. Les fonctions : Le tableau 4 présente une liste non-exhaustive de fonctions standards (ou prédéfinies) applicables sur des entiers ou des réels.

Tableau 4

Fonction	Appel	Résultat retourné
Abs	Abs(x)	La valeur absolue d'un nombre x.
Exp	Exp(x)	L'exponentiel d'un nombre x.
Ln	Ln(x)	Le logarithme népérien d'un nombre x.
Log10	Log10(x)	Le logarithme à base 10 d'un nombre x.
Sqrt	Sqrt(x)	La racine carrée d'un nombre x.
Sqr	Sqr(x)	Le carré d'un nombre x.
Arctan	Arctan(x)	L'arc tangente d'un nombre x.
Cos	Cos(x)	Le cosinus d'un nombre x.
Sin	Sin(x)	Le sinus d'un nombre x.
Round	Round(x)	La valeur arrondie d'un nombre x.
Trunc	Trunc(x)	La partie entière d'un nombre x.
Etc.

I.5.5. La priorité dans les expressions : La priorité des opérateurs dans les expressions arithmétiques, logiques et relationnelles est comme suit :

1. Les parenthèses ;
2. Les fonctions ;
3. Le moins unaire, le Not ;
4. *, /, Div, Mod, And
5. +, -, Or
6. =, <>, <, >, <=, >=

I.5.6. Évaluation des expressions :

L'évaluation d'une expression consiste à calculer, au fur et à mesure, les résultats des calculs jusqu'à obtenir un résultat finale. Cela se fait en plusieurs étapes :

- Écrire l'expression sous forme linéaire (Il faut noter qu'en algorithmique, les expressions s'écrivent sous forme linéaire: $\frac{x+z}{y*2} \rightarrow (x+z)/(y*2)$);
- Remplacer les identifiants (c'est à dire les noms) des variables et des constantes par leurs valeurs ;
- Évaluer (Calculer) étape par étape chacune des sous-expressions en commençant par les sous-expressions qui sont dans les parenthèses les plus internes.
- Indiquer à chaque calcul, le rang d'évaluation.

✓ **Remarque :**

Si les opérateurs ont le même rang de priorité, l'évaluation se fait de gauche à droite.

▪ **Exemple :**

Soit x, y, z, k des variables réelles, tel que x = 4, y = 3, z = 2 et k=-3. Évaluer l'expression suivante en indiquant l'ordre d'évaluation.

$(x < y) \text{ Or } (z + k > (x/z)) \text{ And Not } (8 \text{ Div } z > 7 \text{ Mod } y)$

$(4 < 3) \text{ Or } (2 + -3 > (4/2)) \text{ And Not } (8 \text{ Div } 2 > 7 \text{ Mod } 3)$ Remplacement des variables
(1) 2

$(4 < 3) \text{ Or } (2 + -3 > 2) \text{ And Not } (8 \text{ Div } 2 > 7 \text{ Mod } 3)$ Les parenthèses sont de même degré,
(2) False on commence par la plus à gauche

False Or $(2 + -3 > 2) \text{ And Not } (8 \text{ Div } 2 > 7 \text{ Mod } 3)$ Le moins unaire est plus prioritaire que
(3) -3 + et >

False Or $(2 - 3 > 2) \text{ And Not } (8 \text{ Div } 2 > 7 \text{ Mod } 3)$ Les opérations sont évaluées une par une
(4) -1

False Or $(-1 > 2) \text{ And Not } (8 \text{ Div } 2 > 7 \text{ Mod } 3)$
(5) False

False Or False And Not $(8 \text{ Div } 2 > 7 \text{ Mod } 3)$ Div et Mod sont au même rang → le plus à
(6) 4 gauche

False Or False And Not $(4 > 7 \text{ Mod } 3)$
(7) 1

False Or False And Not $(4 > 1)$
(8) True

False Or False And Not True Not est plus prioritaire que Or et And
(9) False

False Or False And False
(10) False

False Or False
(11) False

= False