

## Chapitre 1 / Introduction à l'informatique

### I) Définition de l'informatique

Le mot informatique a été proposé par Philippe Dreyfus en 1962; c'est un mot-valise, formé d'information et d'automatique. L'informatique c'est donc une automatisation de l'information, plus exactement ***un traitement automatique de l'information***. L'information désigne ici tout ce qui peut être traité par l'ordinateur (textes, nombres, images, sons, vidéos,...).

L'outil utilisé pour traiter l'information de manière automatique s'appelle un ordinateur. Ce nom a été proposé par Jacques Perret (professeur de Latin à La Sorbonne) en 1954. Ce mot était à l'origine un adjectif qui signifiait "qui met de l'ordre", "qui arrange".

L'informatique désigne donc un concept, une science, tandis que l'ordinateur est un outil, une machine conçue pour réaliser des opérations informatiques.

### II) Evolution de l'informatique et des ordinateurs

#### Première génération (1936-1956) : Lampes, relais, tubes à vide. « Salle entière »

**1937** : Alan Mathison Turing présente sa machine de Turing, le premier calculateur universel programmable, et invente les concepts de programmation et de programme.

**En 1945** à l'université de Pennsylvanie, l'ENIAC (Electronic Numerator Integrator and Computer) a été inventé

**1948** : Apparition des premières machines à architecture de Von Neumann

**En avril 1952**, IBM produit son premier ordinateur, l'IBM 701, pour la défense américaine.

**En juillet 1953**, IBM lance l'IBM 650,

**En avril 1955**, IBM lance l'IBM 704, premier ordinateur commercial capable aussi de calculer sur des nombres à virgule flottante.

#### Deuxième génération (1956-1963) : Les transistors, « Moitié d'une salle »

**En 1948**, l'invention du **transistor**, un circuit très compact qui ne craint pas les chocs et ne chauffe pas,

**En 1960**, l'IBM 7000 est le premier ordinateur à base de transistors.

#### Troisième génération (1963-1971) : Les circuits intégrés, « Boîte TV »

**En 1958** : l'invention des circuits intégrés par Jack St. Clair Kilby,

**1964** : IBM annonça la série 360, première gamme d'ordinateurs compatibles entre eux

Dans les années 1970 IBM a sorti une série de mini-ordinateurs. La série 3 : 3/6, 3/8, 3/10, 3/12, 3/15.

Ensuite dans les années 1980 la série 30 : 32, 34, 36, 38. Une troisième série a succédé à la série 30 : les AS/400.

#### Quatrième génération (1971 à 1989): Microchips (millions de transistors), Les micro-processeur, taille de la main :

**1971** : Intel dévoile le premier microprocesseur commercial, le 4004

**1974** : Le deuxième microprocesseur Intel 8080

**1981** : IBM lance le **PC** (pour *Personal Computer*, qui signifie « ordinateur personnel »)

**1982** : L'ordinateur le plus vendu de tous les temps fut sans doute le Commodore 64, dévoilé par Commodore International.

La première version de Windows, commercialisée par Microsoft en **1985**

Fin des années **1980** : Les **premiers ordinateurs portables** font leur apparition.

### **Cinquième génération (1989 à 2000) :**

L'ère d'**Internet** commence et la toile mondiale **WORLD WIDE WEB** est inventée en 1989.

### **Sixième génération (2001 à nos jours) :**

L'**apparition** d'une autre génération d'ordinateurs qui est dérivée de celle d'internet, c'est celle des réseaux **Peer-to-Peer (P2P)** et des **Grilles de calcul**.

## **III) Principe de fonctionnement d'un ordinateur**

Les deux principaux constituants d'un ordinateur sont la *mémoire principale* et le *processeur*. La mémoire principale (MP en abrégé) permet de stocker de l'information (programmes et données), tandis que le processeur exécute pas à pas les instructions composant les programmes.

### **Notion de programme**

Un programme est une suite d'instructions élémentaires, qui vont être exécutées dans l'ordre par le processeur. Ces instructions correspondent à des actions très simples, comme additionner deux nombres, lire ou écrire une case mémoire, etc. Chaque instruction est codifiée en mémoire sur quelques octets.

Le processeur est un circuit électronique complexe qui exécute chaque instruction très rapidement, en quelques *cycles d'horloges*. Toute l'activité de l'ordinateur est cadencée par une horloge unique, de façon à ce que tous les circuits électroniques travaillent ensemble. La fréquence de cette horloge s'exprime en MHz (millions de battements par seconde). Par exemple, un ordinateur "PC Pentium 133" possède un processeur de type Pentium et une horloge à 133 MHz.

Pour chaque instruction, le processeur effectue schématiquement les opérations suivantes:

1. lire en mémoire (MP) l'instruction à exécuter ;
2. effectuer le traitement correspondant ;
3. passer à l'instruction suivante.

Le processeur est divisé en deux parties (voir figure suivante), l'unité de commande et l'unité de traitement :

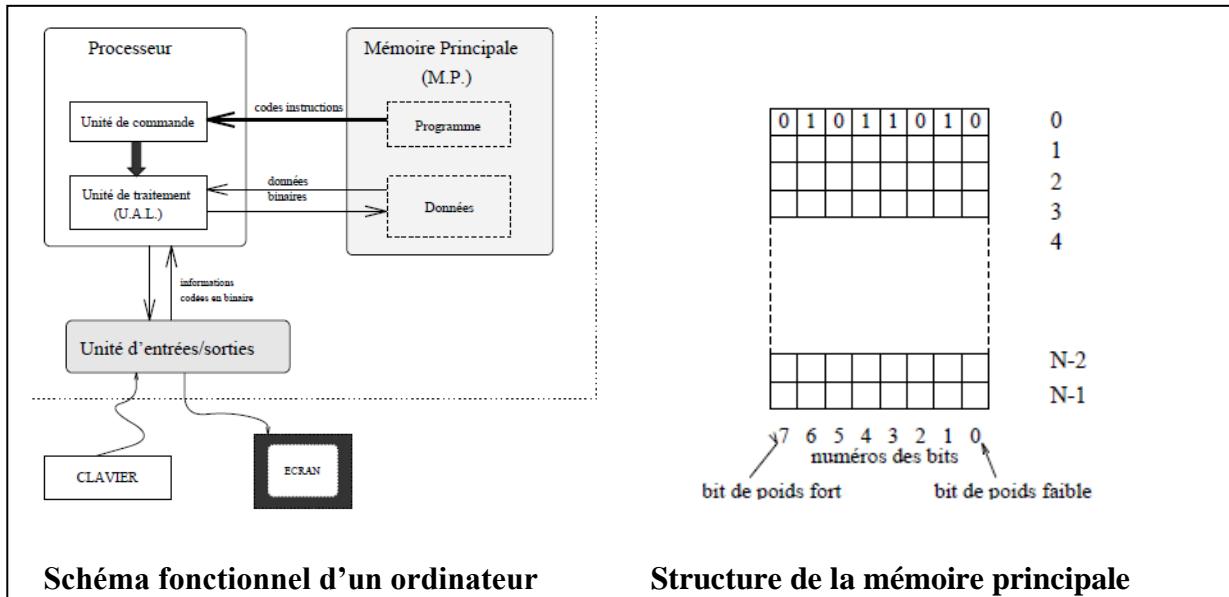
- l'unité de commande (U.C) est responsable de la lecture en mémoire et du décodage des instructions ;
- l'unité de traitement, aussi appelée *Unité Arithmétique et Logique* (U.A.L.), exécute les instructions qui manipulent les données.

### **1. La mémoire principale (MP)**

#### **Structure de la MP**

La mémoire est divisée en emplacements de taille fixe (par exemple 8 bits) utilisés pour stocker instructions et données.

En principe, la taille d'un emplacement mémoire pourrait être quelconque ; en fait, la plupart des ordinateurs en service aujourd'hui utilisent des emplacements mémoire d'un octet (*byte* en anglais, soit 8 bits, unité pratique pour coder un caractère par exemple).



Dans une mémoire de taille N, on a N emplacements mémoires, numérotés de 0 à N - 1. Chaque emplacement est repéré par son numéro, appelé *adresse*. L'adresse est le plus souvent écrite en hexadécimal.

La capacité (taille) de la mémoire est le nombre d'emplacements, exprimé en général en kilo-octets ou en mégaoctets, voire davantage. Rappelons que le kilo informatique vaut 1024 et non 1000 ( $2^{10} = 1024 \neq 1000$ ). Voici les multiples les plus utilisés :

1 K (Kilo)	$2^{10}$	= 1024
1 M (Méga)	$2^{20}$	= 1048 576
1 G (Giga)	$2^{30}$	= 1 073 741 824
1 T (Téra)	$2^{40}$	= 1 099 511 627 776

## Opérations sur la mémoire

Seul le processeur peut modifier l'état de la mémoire. Chaque emplacement mémoire conserve les informations que le processeur y écrit jusqu'à coupure de l'alimentation électrique, où tout le contenu est perdu (contrairement au contenu des mémoires externes comme les disquettes et disques durs).

Les seules opérations possibles sur la mémoire sont :

- *écriture* d'un emplacement : le processeur donne une valeur et une adresse, et la mémoire range la valeur à l'emplacement indiqué par l'adresse ;
  - *lecture* d'un emplacement : le processeur demande à la mémoire la valeur contenue à l'emplacement dont il indique l'adresse. Le contenu de l'emplacement lu reste inchangé.

## Unité de transfert

Notons que les opérations de lecture et d'écriture portent en général sur plusieurs octets contigüs en mémoire : un *mot* mémoire. La taille d'un mot mémoire dépend du type de processeur ; elle est de

- 1 octet (8 bits) dans les processeurs 8 bits (par exemple Motorola 6502) ;
  - 2 octets dans les processeurs 16 bits (par exemple Intel 8086) ;
  - 4 octets dans les processeurs 32 bits (par ex. Intel 80486 ou Motorola 68030).

## 2. Le processeur central

Le processeur est parfois appelé CPU (de l'anglais *Central Processing Unit*) ou encore MPU (*Micro-Processing Unit*) pour les microprocesseurs.

Un *microprocesseur* n'est rien d'autre qu'un processeur dont tous les constituants sont réunis sur la même puce électronique, afin de réduire les coûts de fabrication et d'augmenter la vitesse de traitement. Les *microcoordinateurs* sont tous équipés de microprocesseurs.

L'architecture de base des processeurs équipant les gros ordinateurs est la même que celle des microprocesseurs.

### Les registres et l'accumulateur

Le processeur utilise toujours des *registres*, qui sont des petites mémoires internes très rapides d'accès utilisées pour stocker temporairement une donnée, une instruction ou une adresse. Chaque registre stocke 8, 16 ou 32 bits.

Le nombre exact de registres dépend du type de processeur et varie typiquement entre une dizaine et une centaine.

Parmi les registres, le plus important est le registre *accumulateur*, qui est utilisé pour stocker les résultats des opérations arithmétiques et logiques. L'accumulateur intervient dans une proportion importante des instructions.

Par exemple, examinons ce qu'il se passe lorsque le processeur exécute une instruction comme “Ajouter 5 au contenu de la case mémoire d'adresse 180” :

1. Le processeur lit et décode l'instruction ;
2. le processeur demande à la mémoire la contenu de l'emplacement 180 ;
3. la valeur lire est rangée dans l'accumulateur ;
4. l'unité de traitement (UAL) ajoute 5 au contenu de l'accumulateur ;
5. le contenu de l'accumulateur est écrit en mémoire à l'adresse 180.

C'est l'unité de commande (voir figure) qui déclenche chacune de ces actions dans l'ordre. L'addition proprement dite est effectuée par l'UAL.

## IV) Partie Matériel d'un Ordinateur

**La Partie matérielle (Hardware)** : constituée de l'unité centrale du traitement et des organes d'entrée-sortie.

### 1- Unité Centrale (boîtier)

L'unité centrale est un boîtier métallique contenant les éléments internes de l'ordinateur.

#### 1-1-Unité Centrale de traitement (Central Processing Unit) CPU

L'unité Centrale de traitement ou le microprocesseur est le cerveau de l'ordinateur. Elle est constituée de :

- **L'unité de calcul** : permet d'effectuer les opérations arithmétiques (addition, multiplication, division, soustraction) et les opérations logique (la comparaison, et, ou)
- **L'unité de commande** : permet de contrôler, gérer et organiser les travaux réalisés par le CPU.

Le **microprocesseur** est caractérisé par :

- Sa **marque**
- Sa **fréquence** (vitesse) en **GHz**

#### Exemple

Intel Core 2 Duo 2,2GHz

AMD FX-8150 4,2 GHz

#### 1-2-Mémoire

La mémoire est un organe qui permet d'enregistrer, de stocker et de restituer des informations. On distingue deux types :

- **RAM** (Random Access Mémory) : est une **mémoire vive**, accessible en **lecture** et en **écriture**, sert à stocker **temporairement** les informations, elle dite **volatile** parce que

elle perd son contenu dès qu'elle est hors tension, elle se présente sous forme de petites barrettes. **Exemple :** RAM 4Go – RAM 512 Mo.

- **ROM** (Read Only Mémory) : est une **mémoire morte, permanente**, accessible seulement en **lecture**, Elle contient les programmes de constructeur (**bios**) nécessaires au démarrage de l'ordinateur.

### 1-3 Carte mère

La carte mère est la grande carte électronique de l'ordinateur, c'est sur elle que sont assemblés tous les éléments (microprocesseur, RAM, des disques durs, lecteurs, cartes d'extension, périphériques,...).

**1-4-Cartes d'extension :** Les cartes d'extensions servent à **étendre les capacités et les performances** de l'ordinateur. **Exemple :** Carte satellite -carte TV - carte réseau- carte USB...

**1-5-Disque dur :** Un disque dur permet de conserver et de stocker les données à long terme.

**1-6-Lecteur CD /DVD :** Un lecteur de CD-ROM (lecteur de DVD) permet de lire les CD-ROM (lire les DVD-ROM et les CD-ROM)

**1-7- Graveur CD / DVD :** Un graveur de DVD permet de graver (écrire) des CD et des DVD mais un graveur de CD ne gradera pas forcement les DVD.

**1-8-Boite d'alimentation :** La boite d'alimentation permet de fournir du courant électrique à l'ensemble des composants de l'ordinateur.

**2-Les périphériques :** Un périphérique est un dispositif extérieur, relié à l'unité centrale afin d'envoyer, recevoir ou stocker les informations.

**2-1-Les périphériques d'entrée :** Ces périphériques permettent d'envoyer les informations à l'unité centrale.

**2-2-Les périphériques de sortie :** Ces périphériques permettent de recevoir ou d'afficher les informations sortant de l'unité centrale.

**2-3 - Les périphériques d'entrée/ sortie :** Ces périphériques permettent d'envoyer et d'afficher ou recevoir les informations.

**3-Les périphériques de stockage :** Ces périphériques permettent de stocker et de conserver les informations.

**4- Les connecteurs (ports) des périphériques :** Le boîtier de l'unité centrale est muni d'un ensemble d'emplacements (ports) destinés à connecter (brancher) les périphériques.

## V) Partie Software

**La Partie logicielle (software) :** est un ensemble de programmes informatiques.

-**Un programme informatique** est une succession d'instructions exécutable par l'ordinateur.

-**Un logiciel** est un programme ou ensemble de programmes informatiques nécessaires au **fonctionnement** et à **l'utilisation** d'un ordinateur ou un système informatique. On distingue deux types de logiciels : les logiciels de base et les logiciels d'application.

## 1- Les logiciels d'applications

Un logiciel d'application est un ensemble de programmes destinés à traiter des tâches particulières.

Exemples :

- \_ Logiciels de traitement de texte (Word, AmiPro, Word Perfect)
- \_ Logiciels de tableur (Excel, OpenCalc, Quattro Pro,...)
- \_ Logiciels de jeux
- \_ Logiciels de programmation (C, Pascal, Java, Visual Basic,...)
- \_ Logiciels de base de données (Access, oracle,...)
- \_ Logiciels graphiques (Paint, Autocad, photoshop...)
- \_ Logiciels de présentation (power point,...)
- \_ Logiciels de gestion de base de données (Access, oracle,...)
- \_ Logiciels pour l'Internet (Internet explorer, MSN Messenger,...)
- \_ .....

## 2- Logiciel de base

Un logiciel de base est un logiciel indispensable au fonctionnement de l'ordinateur, il permet de contrôler et de gérer toutes les ressources matérielles et logicielles de l'ordinateur. Ce logiciel est appelé **système d'exploitation**.

Exemples :

- **MS-DOS** (Microsoft Disk Operating System),
- **Windows** (95, 98, 2000, XP, Vista, 7, 8),
- **Unix**,
- **Linux**,
- **MAC OS**

## 3/ Fonctionnalités de base d'un système d'exploitation

Le système d'exploitation est un ensemble de programmes permettant de :

- Assurer le **contrôle et le fonctionnement** de la partie matérielle ;
- Faciliter la **communication** entre l'utilisateur et le matériel à l'aide **d'une interface graphique** ;
- Réaliser la liaison entre les ressources matérielles d'un ordinateur et les logiciels d'applications installés ;
- Orienter les entrées / sorties ;
- Gérer la mémoire et le stockage des données dans un ordinateur ;
- Permettre l'installation et le lancement des applications sur ordinateur ;

## 4/ Concepts de base de l'environnement graphique d'un système d'exploitation (ex : WindowsXP)

### 1-Bureau

Le bureau est la zone de travail de l'écran dans laquelle apparaissent les icônes, les raccourcis, la barre des tâches, les fenêtres, les menus et les boîtes de dialogues.

#### a- Les icônes

Les icônes sont de petites images sur le bureau, chaque icône **représente un programme** ou **un dossier** ou **un fichier**. Pour ouvrir un fichier ou un programme, double cliquez sur l'icône correspondante.

#### b- Les raccourcis

Le raccourci est un **lien** vers un programme, un fichier, un dossier, un lecteur de disque.

Les icônes de raccourci sont identifiées par la **flèche** située sur le coin inférieur gauche de l'image.

## 2- Fenêtre

Les Fenêtres sont des cadres dans le bureau qui affichent le **contenu** : des fichiers, des dossiers et des programmes.

## 3- Barre des Tâches

La barre des tâches est une bande étroite en bas de l'écran, elle contient le bouton **Démarrer** à gauche et la **zone de notification** à droite.

Lorsque vous ouvrez une fenêtre, le bouton qui la représente apparaît sur la barre des tâches. Le bouton disparaît lorsque vous fermez la fenêtre.

## 4- Menu Démarrer

Le menu Démarrer contient tous les éléments dont vous avez besoin pour commencer à utiliser Windows.

Pour afficher le menu démarrer, cliquez sur le bouton **démarrer**

Volet de navigation

À partir de ce menu, vous pouvez :

- \_ Démarrer des programmes en cliquant sur **Tous les programmes**
- \_ Ouvrir des fichiers et des dossiers (poste de travail,...)
- \_ Personnaliser votre système avec le **Panneau de configuration**.
- \_ Obtenir de l'aide en cliquant sur **Aide et support**.
- \_ Rechercher des éléments sur votre ordinateur ou sur Internet en cliquant sur **Rechercher**.
- \_ Arrêter l'ordinateur

## 3-Notion du fichier et dossier

\_ **Un fichier** est un ensemble d'informations enregistrées dans un support de stockage.

Un fichier est enregistré sous la forme "**nom du fichier.ext**".

".ext" représente l'extension : c'est un **moyen** de reconnaître la ou les applications pouvant créer ou ouvrir ce type de fichier.

**Un fichier** est caractérisé par :

- Nom
- Taille
- Type (texte, son, image, vidéo,...)
- Extension (doc - ppt - xls - html - pdf...) - (mp3 - midi - wav - arm...) - (bmp - Jpg - gif...) - (avi - mpeg - flv - 3gp...)
- \_ **Un dossier ou répertoire** : est un emplacement qui permet de stocker des fichiers et des dossiers

## VI) Concept d'un algorithme

Qu'est-ce qu'un algorithme ?

Un algorithme est une suite ordonnée d'actions qui permet d'arriver à un but.

Il est caractérisé par :

- un début et une fin, permettant de délimiter l'algorithme.
- un ensemble d'étapes ou d'actions à exécuter.
- une description précise de ces actions.
- un ordre d'exécution de ces actions, déterminé par la logique d'enchaînement de celles-ci, et constitué par les structures mises en œuvre.

Dans la vie de tous les jours vous utilisez des algorithmes sans le savoir:

- Des fiches de cuisine.
- Des notices de montage de meuble.
- Dans le choix de vos vêtements suivant la saison, le temps, votre humeur, ...

## 1- Représentation d'un algorithme

Pour représenter un algorithme, on peut recourir à diverses méthodes :

- organigramme de programmation
- Langage structuré (pseudo-code)

**Exemple :**

Organigramme	Pseudo code
<pre> graph TD     DEBUT([DEBUT]) --&gt; Caff[/Caff/]     Caff --&gt; Decision{Caff &gt; 12000 ?}     Decision -- Oui --&gt; Rist1[Rist = Caff * 12]     Decision -- Non --&gt; Rist0[Rist = 0]     Rist1 --&gt; Parall[« La ristourne est de » ; Rist]     Rist0 --&gt; Parall     Parall --&gt; FIN([FIN])   </pre>	<b>VARIABLES</b> Caff : Réel Rist : Réel <b>DEBUT</b> Lire Caff SI Caff > 12000 ALORS Rist ← Caff * 0,12 SINON Rist ← 0 FIN SI Ecrire « La ristourne est de » ; Rist <b>FIN</b>

## 2- Structure générale d'un algorithme

Programme nom\_prog

Variables

Liste des variables

Constantes

Liste des constantes

Début

Corps de l'algorithme

Fin

Remarque:

Un algorithme s'exécute de façon séquentielle: Les instructions s'exécutent les unes après les autres dans l'ordre où elles sont écrites.

## 3- La démarche et analyse d'un programme

De manière générale, une analyse d'un problème dans le but de le résoudre peut se ramener à une suite d'interrogation qui s'enchaîne :

- Quel est le résultat à obtenir ? (Que veut – on ?)
- Quelles opérations permettent d'élaborer ce résultat ? (comment s'obtient ce résultat ?)
- Quelles sont les données nécessaires à la réalisation de ces opérations ? (De quoi a-t-on besoin ?)

Quel que soit le type de problème qu'on aura en face de soi, l'analyse consistera à :

- Définir précisément le problème pour identifier le(s) résultat(s) (Sortie) à obtenir.
- Identifier les traitements permettant d'obtenir le résultat dans le cadre du problème énoncé.

- Identifier les informations (Entrées) nécessaires à la réalisation des traitements envisagés.
- Dérouler l'algorithme étape par étape à travers un jeu de tests pour vérifier s'il produit le résultat demandé.

Ainsi l'algorithme peut se présenter par le schéma classique suivant :

Entrées(Données) ----->Traitement (opérations) -----> Sorties(Résultats)

### ***Exercice d'application***

**Enoncé :** Votre grand père détient un terrain sous forme rectangulaire au village. Il aimerait connaître la superficie occupée par son terrain.

**Travail à faire :** Concevoir un algorithme permettant à votre grand père de résoudre son problème.

#### **1) Analyse :**

- Quel est le résultat recherché ? (Sortie)

On veut obtenir la valeur de la superficie (la surface) occupée par le terrain.

- Comment s'obtient ce résultat ?(Traitement)

Le terrain de forme rectangulaire. Un rectangle est une figure géométrique ayant 4 cotés et constitué d'une longueur (L) et d'une largeur (l). Pour obtenir la surface, il est nécessaire de connaître la formule c'est-à-dire Surface = Longueur \* largeur et de faire le calcul du produit par la machine.

- Quelles sont les informations nécessaires ? (Entrées)

Dans la formule ci-dessus, on aura besoin de connaître la longueur du terrain et sa largeur.

#### **2) Structure de la solution :** la solution de l'énoncé implique la structuration suivante :

- Donner la Longueur (L)
- Donner la largeur (l)
- Calculer la surface (S) en effectuant :  $S = L * l$
- Afficher la surface (S).

## **VII Structure des données**

### **A) Les variables et les constantes**

Une variable ou une constante sert à mémoriser une valeur donnée, durant un algorithme.

Une variable ou une constante :

- Porte un nom (le plus significatif possible). Exemple pour le prix hors taxe : Px\_HT
- Est défini par un type de donnée appelé aussi format de donnée. Exemple Entier, Réel, Chaîne de caractères, booléen, ...
- Contient une valeur. Exemple : 117,95. Cette valeur peut lui être affectée par le développeur, l'utilisateur ou après calcul par l'algorithme.

#### **Remarque :**

***Le contenu de la variable pourra évoluer au cours de l'algorithme alors que celui d'une constante est défini au début de l'algorithme et ne pourra pas être modifié.***

Les variables et les constantes correspondent à des zones de stockage de la mémoire vive. Elles permettent de conserver des valeurs en vue d'un traitement.

Une variable peut être mise à jour par l'utilisateur (par l'action LIRE) exemple : **LIRE Caff** ou être modifiée par une action exemple : **Rist ← Caff\*0,12** : la variable Rist est affectée par le calcul Caff \* 0,12

Une constante, en revanche, est définie une fois pour toute à l'intérieur du « programme » (avant les déclarations de variables) et ne peut pas être modifiée par l'utilisateur. Une constante correspond à un paramètre.

### **B) Les types de données**

**ENTIER** : Un objet de ce type peut prendre une valeur entière comprise généralement entre -32768 et +32767 .

**REEL** : Un objet de ce type peut prendre une valeur décimale dont la valeur absolue est dans l'intervalle : [5.3976E -79 , 7.2370E+75]

**CARACTERE** : Il y a 128 caractères ("A", "a" ou "!", "@"). Attention : le symbole 1 correspond à la valeur entière "un", alors que "1" correspond au caractère 1. Certains caractères (de 0 à 19) ne peuvent être imprimés, par exemple le caractère de code 07 est en fait le bip sonore.

**CHAINE** : Tout objet pouvant prendre comme valeur une suite ou une chaîne de caractères.  
Exemple : "Bonjour à tous" est une expression type chaîne de caractères.

**BOOLEEN** : Un objet de type BOOLEEN prend sa valeur dans l'ensemble {vrai,, faux, }.

**Les tableaux** : On peut regrouper plusieurs variables sous un même nom, chacune étant alors repérée par un numéro. C'est ce que l'on appelle un tableau. On peut faire un tableau avec des variables de n'importe quel type.

### **C) Les opérations**

## **1- L'opération d'affectation**

Elle est formalisée par le symbole " $\leftarrow$ ". Elle indique qu'une variable est affectée (reçoit) une valeur.

**FORMAT :** *Variable*  $\leftarrow$  *valeur*

La valeur affectée peut être :

- Une variable de même type  $A \leftarrow B$  ( $A$  et  $B$  Entiers)
  - Une constante de même type  $Tx\_TVA \leftarrow 19,6$
  - Le résultat d'une opération de même type  $Px\_TTC \leftarrow Px\_HT + Mt\_TVA$

### Exemples :

**I←1** : Affecte la valeur 1 à la variable I

**I←NbArticles+10** : affecte à la variable I le contenu de la variable NbArticles augmentée de 10

**I←I+1** : Augmente la valeur de la variable I de 1 (on parle d'incrémentation) et affecte la nouvelle valeur à la variable

**NomCli**  $\leftarrow$  « Durand » : Affecte la chaîne de caractère Durand à la variable NomCli

**NomCli**  $\leftarrow$  « Durand » & « » & PrenCli : affecte la valeur Durand suivi d'un espace et du contenu de la variable PrenCli à la variable NomCli

L'opérateur & permet de concaténer des chaînes de caractères

## Espérance et Remarques :

- Pour additionner des nombres ou concaténer du texte, les types de données doivent être compatibles (ex : on ne peut pas additionner des chiffres avec du texte)

### Exercices :

- 1) Donnez les valeurs des variables A, B et C après exécution des instructions suivantes ?

### Variables A B C: Entier

Varieté

Debut

$$\begin{array}{r} A \\ B \end{array} \leftarrow \begin{array}{r} 5 \\ 7 \end{array}$$

$$\Delta \leftarrow B$$

```
C ← A + B
C ← B - A
Fin
```

- 2) Donnez les valeurs des variables A et B après exécution des instructions suivantes ?

**Variables A, B : Entier**

**Début**

```
A ← 1
B ← 2
A ← B
B ← A
```

**Fin**

Les deux dernières instructions permettent-elles d'échanger les valeurs de A et B ?

Ecrire un algorithme permettant d'échanger les valeurs de deux variables A et B ?

## 2- Les opérations arithmétiques

Soit 20 la valeur de a et 2 la valeur de b :

Opérations	Opérateurs	Exemples	Résultats
Addition	+	a + b	22
Soustraction	-	a - b	18
Multiplication	*	a*b	40
Division	/	a/b	10
Elévation à la puissance	^	a ^ b	400

### a. Opération de concaténation

Relie des caractères ou des chaînes de caractères pour constituer une nouvelle chaîne de caractères. On utilise essentiellement le caractère "&"

Exemples "A" & "B" = "AB"

"Bon"&"jour" = "Bonjour"

### b. Les opérateurs relationnels

Ils s'appliquent essentiellement aux objets de type entier, réel, caractère et chaîne de caractères. Ainsi, il est possible de comparer des données de même type entre elles pour savoir si elles sont égales, plus grandes ou plus petites. La comparaison de deux chaînes de caractères est également possible et s'effectue caractère par caractère, de gauche à droite. Le résultat est soit vrai (l'expression logique est vérifiée) soit faux (l'expression logique n'est pas vérifiée).

Opérations	Significations	Exemples	Résultats
=	égal	20=10*2	vrai
$\diamond$	différent de	"A" $\diamond$ "G"	vrai
<	inférieur	11<8	vrai
$\leq$	inférieur ou égal	20 $\leq$ 10*2	vrai
>	supérieur	8>11	faux
$\geq$	supérieur ou égal	"au revoir" $\geq$ "bonjour"	faux

### 3- Les opérateurs logiques

Les opérateurs logiques sont définis à partir du *calcul des propositions* en suivant les règles d'une algèbre particulière, algèbre logique ou algèbre de BOOLE.<sup>1</sup>

Définitions :

**Proposition :** une proposition est l'énoncé d'une propriété telle que l'on peut toujours lui affecter la valeur *vrai* ou la valeur *faux*.

Ex : le salaire de DUPONT est supérieur au salaire de DURAND est une proposition (P1).

**Valeur logique d'une proposition :** c'est un nombre qui est 0 si la proposition est fausse et 1 si elle est vraie. On représente généralement cette valeur logique à l'aide d'une variable caractéristique associée à cette proposition. Cette variable *booléenne* ou *binaire* a deux valeurs conventionnellement représentées par 0 ou 1.

Ex : Si la proposition P1 est vraie la variable booléenne associée sera dans ce cas a=1.

Les opérateurs de base *ou*, *et*, *non* permettent à partir de propositions élémentaires, de construire des propositions complexes.

#### i. Operateur union logique (ou somme logique) : OU

La table suivante définit l'opérateur OU :

P1		P2		P3 = P1 OU P2	
Valeur	Variable associée	Valeur	Valeur associée	Valeur	Variable associée
V	a = 1	V	a' = 1	V	a + a' = 1
V	a = 1	F	a' = 0	V	a + a' = 1
F	a = 0	V	a' = 1	V	a + a' = 1
F	a = 0	F	a' = 0	F	a + a' = 0

#### ii. Operateur intersection logique (ou produit logique) : ET.

La table suivante définit l'opérateur ET.

P1		P2		P3 = P1 ET P2	
Valeur	Variable associée	Valeur	Valeur associée	Valeur	Variable associée
V	a = 1	V	a' = 1	V	aa' = 1
V	a = 1	F	a' = 0	F	aa' = 0
F	a = 0	V	a' = 1	F	aa' = 0
F	a = 0	F	a' = 0	F	aa' = 0

#### iii. Operateur négation : NON

P1		NON P1	
Valeur	Variable associée	Valeur	Variable associée
V	a = 1	F	a' = 0
F	a = 0	V	a' = 1

<sup>1</sup>BOOLE George (1815 - 1864) Mathématicien anglais.

**b. Les priorités dans les opérations**

- Priorité de \*, / div et % par rapport à + et -  
 $5 + 9 * 3 = 32$  et non 42  
 $5 * 9 + 3 = 48$  et non 60
- Pour les opérateurs de même priorité, associativité à partir de la gauche  
 $15 / 5 * 3 = 9$  et non 1  
 $5 - 2 + 4 = 7$  et non -1
- On peut utiliser des parenthèses pour changer l'ordre des opérations :  
 $15 / (5 * 3) = 1$   
 $(5 + 9) * 3 = 42$

**VIII) Les instructions d'entrée / sortie**

L'instruction Saisir ou Lire, attend que l'utilisateur saisisse une valeur au clavier

L'instruction Afficher ou Ecrire, affiche un message ou la valeur d'une variable (constante).

**FORMAT :** *Lire variable*  
*Ecrire "message"*  
*Ecrire variable*  
*Ecrire "message", variable*

La saisie peut être précédée d'une phrase explicative :

**Forme 1**

**Ecrire** « *Saisissez le chiffre d'affaires* »  
**Lire** *Caff*

Exercice

Ecrire un programme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement.

**Variables** pht, ttva, pttc **en Réel**  
**Variable** nb **en Entier**  
**Début**  
**Ecrire** “Entrez le prix hors taxes :”  
**Lire** pht  
**Ecrire** “Entrez le nombre d'articles :”  
**Lire** nb  
**Ecrire** “Entrez le taux de TVA :”  
**Lire** ttva  
 $pttc \leftarrow nb * pht * (1 + ttva/100)$   
**Ecrire** “Le prix toutes taxes est : ”, pttc  
**Fin**