

Partie 1: Bureautique & Technologie Web

1. Bref historique sur l'évolution de l'informatique :

Les générations des ordinateurs

1^{er} Génération : (1942-1958)

Qui se caractérise par:

- L'invention de l'ordinateur « MARK1 » en 1942 par BM
- La fabrication de l'ordinateur « ENIAC » en 1957 par « VON NEWMAN »

2^{ème} Génération : (1959-1965)

- Le remplacement des tubes électroniques par des transistors qui a permis de diminuer la taille des ordinateurs.
- La création des « Circuits Intégrés » remplaçant des milliers de transistors.
- L'invention de la notion « Temps Partagé » qui permet d'exécuter plusieurs programmes simultanément (alternativement confondus un par un).

3^{ème} Génération : (1966-1969)

- Évolution sur tous les matériels de l'ordinateur telle que : Les Mémoires, les unités d'entrée/sortie, écran, ...
- L'augmentation de la vitesse d'exécution.
- La conception de ce qu'on appelle un « Réseau d'ordinateurs »

4^{ème} Génération : (1970-1990)

- La fabrication des microprocesseurs d'une petite taille,
- La création des logiciels sophistiqués (multimédias: voix, image et texte...).

5^{ème} Génération : (1990 à nos jours)

- L'évolution des langages de programmation : (Pascal, C, C++, Matlab, Java...).
- L'invention de l'Internet (réseau des réseaux).
- L'évolution des systèmes d'exploitations : Windows 10, XP, Linux, Mac...

2. Définitions :

INFORMATIQUE = INFORMAtion + automATIQUE

a. L'INFORMATIQUE :

Elle représente le Traitement Automatique de l'Information par un Ordinateur d'une façon logique (numérique).

b. L'Information :

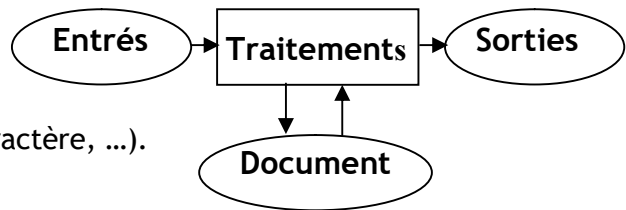
Désigne toute sorte d'information qui peut se communiquer et se formaliser sous forme de textes, d'images ..etc, exemple :

- Les valeurs (Chiffres[0,9]).
- Les caractères d'un texte.
- Les renseignements concernant des personnes, Nom, Prénom, photo...etc

c. Le traitement :

Désigne toutes les manipulations pouvant être effectuées sur les informations, on peut citer :

- Mémorisation (Archivage).
- Les Calculs et Modification
- Tri (Croissant & Décroissant).
- Les Recherches d'information (valeur, caractère, ...).



d. L'ordinateur :

– Est une machine électronique destinée à traiter les informations automatiquement.

On distingue selon leurs puissances et leurs tailles 3 Types d'ordinateurs :

- Les gros Ordinateurs.
- Les Mini-Ordinateurs.
- Les Micro-Ordinateurs.

e. Programme :

Est une suite d'instructions stockées en Mémoire, qui doivent être exécutées une par une.

f. Logiciel :

C'est un ensemble de programmes.

g. Réseau d'ordinateur :

C'est un moyen de communication entre les machines, dans le but d'accéder à n'importe quel ordinateur sans se déplacer ou utiliser une clé USB (flash disque).

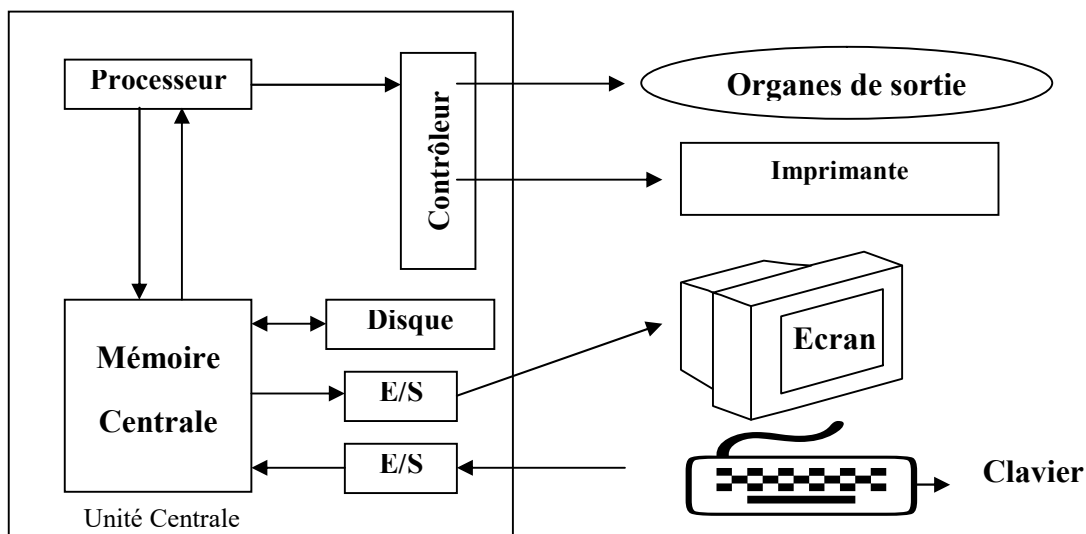
2- Architecture du PC :

Un Ordinateur est composé de deux parties :

2.1- Le Hardware :

Le Hardware est la partie électronique (les Matériels), Par exemple : (Clavier, Souris, Imprimante, ...)

2.1.1- les différentes composants matériels du PC : Les différentes parties d'un ordinateur sont::



• **Processeur :**

- Il exécute les instructions et traite les données
- Il effectue les opérations arithmétiques et logiques.
- Il contrôle les autres organes (E/S, DD ..etc).

- **La Mémoire :**

Elle mémorise les données et les instructions.

On distingue deux types de mémoires :

- 1- **Mémoire vive (RAM) : « Random Access Memory »**

C'est la partie dynamique de la mémoire, on peut faire toutes les opérations suivantes : l'ajout, la suppression et la Modification des données.

- 2- **Mémoire morte (ROM) : « Read Only Memory »**

C'est la partie Statique on ne peut rien changer, elle contient les instructions et les données de base nécessaire au démarrage du système. Ces instructions restent mémorisées même après l'extinction d'ordinateur (mémoire permanente)

- **Les Périphériques :**

Ce sont les organes d'un ordinateur.

1. **Les Périphériques d'entrée :** Ce sont les organes qui permettent d'introduire des informations dans la machine, telle que : Clavier, Scanner, écran tactile...etc
2. **Les Périphériques de sortie :** les organes permettant a la machine de communiquer ses informations vers l'extérieure : Imprimante, Ecran, réseau ...etc
3. **Les Périphériques d'entrée/sortie :** les organes qui permettent d'introduire des informations dans la machine, et aussi de les faire sortir vers l'extérieure : flash, Lecteur-Graveur DVD,...

2.2 - Software :

Le software représente l'ensemble de programmes et de données stockées sous forme de fichiers. C'est la partie non touchable dans un ordinateur, telle que : Les applications, les Logiciels, les Langages de programmations et les systèmes d'exploitation....

2.3- L'unité de mesure de l'information :

Qu'est ce qu'un Bit (Binary Digit)?

C'est une unité qui mesure la quantité d'informations stockées dans la mémoire.

1Octet = 8 Bits, l'octet c'est l'unité la plus petite, les autres unités sont :

1Kilo octet = 1024 Octet.

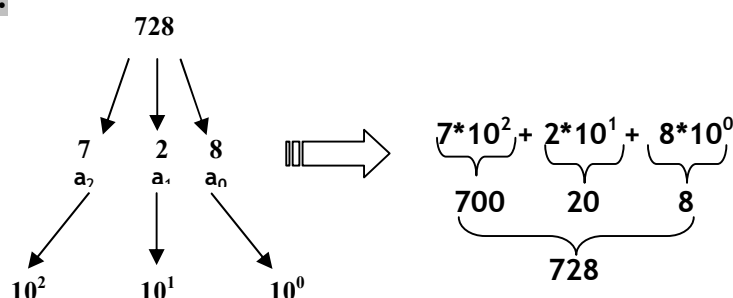
1Méga octet = 1024 Kilo octet.

1Géga octet = 1024 Méga octet.

3- Les systèmes de numérations (Le codage d'Information) :

3.1- Système Décimal : On utilise l'emplacement du numéro pour déterminer sa position.

Exp :



Règle générale :

$$a_0 \times 10^0 + a_1 \times 10^1 + \dots + a_n \times 10^n$$

3.2- Système binaire : Il représente le système numérique, il se compose de 02 chiffres (0,1) appelé aussi système base 2.

3.3- Transformation d'un nombre « Binaire en Décimal » :

On additionne le résultat de chaque multiplication d'un nombre par son classement

<p>Exp1 : $(101)_2 = ()_{10} ?$</p> $\begin{array}{r} \swarrow \quad \downarrow \quad \searrow \\ 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5 \\ 4 + 0 + 1 \end{array} \Rightarrow \boxed{(101)_2 = (5)_{10}}$	<p>Exp2 : $(11)_2 = ()_{10} ?$</p> $\begin{array}{r} \swarrow \quad \downarrow \\ 1 \cdot 2^1 + 1 \cdot 2^0 = 3 \\ 2 + 1 \end{array} \Rightarrow \boxed{(11)_2 = (3)_{10}}$
--	---

3.4- Transformation d'un nombre « Décimal au Binaire » :

On divise le nombre par 2 à chaque fois jusqu'à que le quotient = 0 et le résultat c'est tous les restes de la division et qui s'écrit du droit à gauche (du bas vers le haut).

<p>Exp1 : $(26)_{10} = ()_2 ?$</p> $\begin{array}{r} 26 \mid 2 \\ 0 \mid 13 \mid 2 \\ \quad 1 \mid 6 \mid 2 \\ \quad \quad 0 \mid 3 \mid 2 \\ \quad \quad \quad 1 \mid 1 \mid 2 \\ \quad \quad \quad \quad 1 \mid 0 \end{array} \Rightarrow \boxed{(26)_{10} = (11010)_2}$	<p>Exp2 : $(4)_{10} = ()_2 ?$</p> $\begin{array}{r} 4 \mid 2 \\ 0 \mid 2 \mid 2 \\ \quad 0 \mid 1 \mid 2 \\ \quad \quad 1 \mid 0 \end{array} \Rightarrow \boxed{(4)_{10} = (100)_2}$
--	--

3.5- Les opérations sur le système « Binaire » :

3.5.1-Règles d'additions :

$$\begin{array}{l} 0+0 = 0 \\ 0+1 = 1 \\ 1+0 = 1 \\ 1+1 = 10 \end{array}$$

$$\begin{array}{l} 1+1+1 = 11 \\ 1+1+1+1 = 100 \end{array}$$

Exp1 :

$$\begin{array}{r} + 10110 \\ + 11101 \\ \hline = \boxed{110011} \end{array}$$

Exp2 :

$$\begin{array}{r} + 110 \\ + 111 \\ + 010 \\ \hline = \boxed{1111} \end{array}$$

Exp3 :

$$\begin{array}{r} 1110 \\ + 1011 \\ + 0010 \\ + 0010 \\ \hline = \boxed{11101} \end{array}$$

3.5.2-Règles de la soustraction :

$$\begin{array}{l} 0-0 = 0 \\ 0-1 = 1 \\ 1-0 = 1 \\ 1-1 = 0 \end{array}$$

Exp1 :

$$\begin{array}{r} 1110 \\ - 101 \\ \hline = \boxed{1001} \end{array}$$

Exp2 :

$$\begin{array}{r} 1011 \\ - 110 \\ \hline = \boxed{0101} \end{array}$$

3.5.3-Règles de la multiplication :

Exp :

$$\begin{array}{r}
 * \quad 1011 \\
 \quad \quad 11 \\
 \hline
 + \quad 1011 \\
 \quad 1011. \\
 \hline
 = \boxed{100001}
 \end{array}$$

$0*0 = 0$
$0*1 = 0$
$1*0 = 0$
$1*1 = 1$

3.5.4-Règles de la division :

Exp :

$$\begin{array}{r}
 11011 \overline{) 11} \\
 \underline{0011} \quad \boxed{1001} \\
 0
 \end{array}$$

$1/1 = 1$
$0/1 = 0$

4- Le système d'exploitation :

Le système d'exploitation est un ensemble de programmes assurant les fonctions de contrôle et de gestion de l'ordinateur et de la communication machine-utilisateur ;

* Il gère toutes les ressources de l'ordinateur notamment :

- Le processeur.
- La mémoire.
- Les disques.
- Les lecteurs.
- Les imprimantes et ces ports.
- Ecran + clavier.

* Parmi les systèmes d'exploitation, on peut citer : Windows, Linux (UNIX), Mac, OS/2,...

5- Les logiciels MS office : (Word, Excel,):

Microsoft Office fait son apparition en 1989, et était au début une sorte de paquet comprenant des applications qui étaient vendues séparément auparavant. L'avantage de la suite sur les logiciels séparés était le moindre coût. La première version de la suite bureautique contenait **Word, Excel et PowerPoint**.

Microsoft Office est actuellement la suite bureautique la plus connue dans le monde. Office 2015, sorti fin janvier 2015, a une interface utilisateur très ergonomique, avec comme principal point le remplacement des menus et des barres d'outils par un bandeau contenant les icônes d'outils et diverses fonctions appelé "**Ribbon**" ou "**Ruban**" en français.

6- Les réseaux informatiques

Un réseau informatique est un ensemble d'ordinateurs reliés entre eux grâce à des lignes physiques et échangeant des informations sous formes de données binaires.

Différents types de réseau :

Pour aider à comparer les réseaux en terme de performance et d'utilisation, on les classe couramment en deux catégories :

- Les grands réseaux WAN,
- Les réseaux locaux LAN,

6.1-Les réseaux WAN :

WAN signifie (*Wide Area Network*). Appelés également réseaux étendus ou réseaux à grande distance. Il offre des moyens de communication entre des ordinateurs très éloignés.

6.2-Les réseaux LAN :

LAN signifie (*Local Area Network*) est un réseau permettant de relier généralement des ordinateurs ou des ressources telles que les imprimantes à l'aide de supports de transmission filaire (câbles coaxiaux, fibre optique,....) d'une entreprise ou d'une organisation.

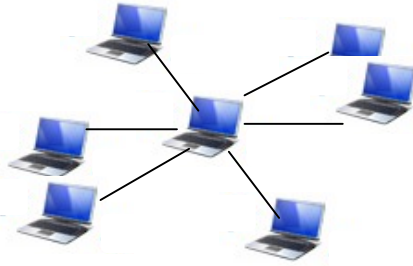
6.3-Intérêt d'un réseau :

Les réseaux et les LAN en particulier permettent :

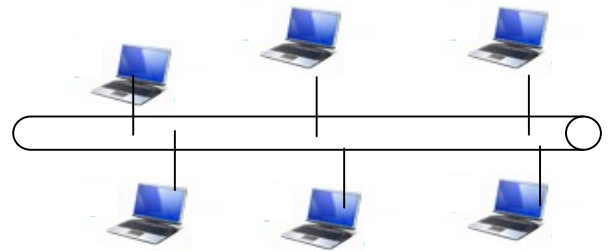
- Le partage de fichiers et d'applications
- La communication entre personnes (grâce au courrier électronique, la discussion en direct, ...)
- La communication entre processus (entre des machines industrielles)
- La garantie de l'unicité de l'information (bases de données)
- Le jeu à plusieurs candidats, ...
- Diminution des coûts grâce aux partages des données et des périphériques

7. Topologie et fonctionnement d'un réseau :

Les ordinateurs d'un réseau peuvent être soit groupés dans une même région soit dispersés géographiquement (selon la topologie).



a. Topologie en Etoile



b. Topologie en Bus

8- Le réseau Internet et le Web

8.1- L'internet :

- L'Internet est un réseau, en réalité une fédération de réseaux c'est à dire des ordinateurs reliés dans le monde entier par des câbles en une toile d'araignée complexe. En un mot, une infrastructure.
- Il désigne aussi les individus qui utilisent ce réseau pour échanger et partager des informations. L'Internet, c'est une communication d'utilisateurs.
- Il représente également les informations échangées sur ce réseau par cette communauté : Une immense masse mondiale de données distribuées.

8.2-Les services d'Internet

8.2.1-Le courrier électronique

Le courrier électronique est le moyen de rédiger sur un ordinateur des messages textuels, de les adresser à un ou plusieurs correspondants, et réciproquement d'en recevoir, de les consulter et de les classer.exp : xxxx@gmail.com

8.2.2-La discussion en ligne

Elle permet aux utilisateurs du monde entier de s'entretenir à l'aide de leurs claviers et caméras. Les courts messages et les images sont exploités en temps réel (en direct). Exp: Skype.

8.2.3-Les forums électroniques

Ils Favorisent les échanges entre de nombreuses personnes au tour du monde. Exp tweeter

8.3- Le Web

Est un système permettant d'accéder à des pages d'informations textuelles, puis rapidement multimédia, ces pages sont reliées des liens hypertextes, sachant que ces pages étaient stockées sur des serveurs du réseau Internet, On aboutissait donc à un système de serveur des pages hypertextes multimédias et distribuées permettant de

naviguer (surfer dans le jargon de l'Internet) de page en page sur un serveur et entre serveurs.

Ce système est appelé le World Wide Web, puis W3 et maintenant le Web (ou la toile) en référence à l'image de la toile d'araignée.

8.3.1- URL (uniforme ressource locator)

L'URL est un moyen standard pour désigner de manière universelle l'ensemble de ressources de l'Internet.

Une URL est construite de la manière suivante :

<Protocole>://<nom du serveur> .<nom de domaine>/<dossier ; fichier>

Exp: https://progres.mesrs.dz/webfve/pages/index.xhtmll

8.3.2-Techniques de programmation Web

Nous citons ci-dessus les différentes techniques et langages de script côté client et coté serveur.

8.3.2.1- Côté client (internet explorer, fire fox, chrome)

Le SGML (Standard Generalized Markup Language)

SGML est un langage normalisé pour la génération de langages de balises. Ce langage très professionnel a la particularité d'être très concis et très abstrait. Sa descendance est assez nombreuse, par exemple le langage de balises utilisé pour la publication sur le Web : Le HTML (Hyper Text Markup Language).

Le HTML

HTML est un langage de description de document il utilise des marqueurs explicites (appelés aussi tags ou balises) qui précisent la structure et la mise en forme du contenu du document. Ces marqueurs seront reconnus par les navigateurs, et interprétés comme des directives, afin de réaliser la présentation attendue sur le poste client.

Le XML

Tout comme HTML, le langage XML (eXtensible Markup Language) représente un sous-ensemble du langage SGML et une « recommandation » W3C(World Wide Web Consortium) pour un langage de balisage décrivant la structure et la nature des données d'un documents.

Java Script

JavaScript est un langage de scripts fondé sur la notion d'objets, permettant l'écriture de scripts qui sont ensuite intégrés aux balises HTML. Cela signifie, que dans un document HTML, la syntaxe JavaScript va se trouver mêlée à la syntaxe HTML. Ces scripts sont utilisés pour animer les pages web.

8.3.2.2- Coté serveur (apache...)

Le PHP

Le PHP (Personal home page, Hypertext Processor) est un langage de script incorporé au document HTML, mais exécuté par le serveur Web et non par le client. Conçu pour gérer les bases de données sous réseaux. Il est donc possible à l'aide de PHP de réaliser dynamiquement le contenu des documents avec des informations externes provenant de bases de données, fichiers,...

Voici un exemple sous la forme d'un document HTML généré dynamiquement à partir d'un script écrit en PHP

```
<html><head>< title> bonjour</title></head>
<body>
< ?PHP echo" bonjour" ;
? >
</body></html>
```

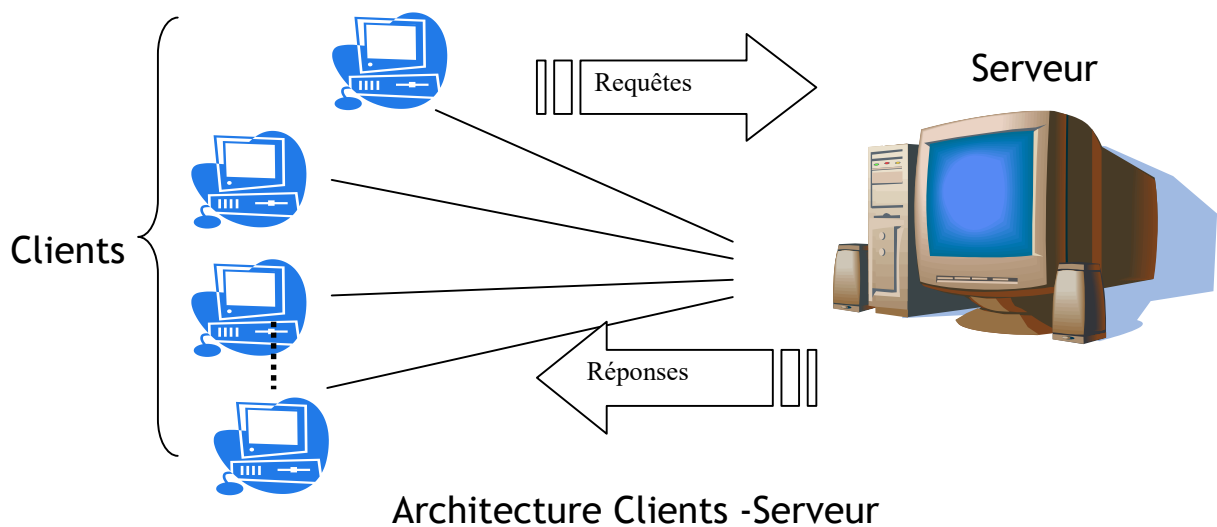
9- Fonctionnement clients -serveur :

9.1- Client (Navigateur Web) :

Les clients Web sont les logiciels de navigation qui permettent d'accéder puis de visualiser les documents présents sur le Web en interprétant le langage HTML. Ces logiciels sont désignés par les termes : Navigateur (browser), fureteur ou butineur. Les clients les plus connus de nos jours sont: *Internet Explorer, Firefox et Google chrome*.

9.2- Serveur Web :

Un serveur Web est un programme spécifique situé sur l'une des machines de l'Internet qui attend qu'une interface de navigation (client) se connecte et lui présente une requête (demande d'une page web). Une fois la requête est reçue le serveur localise le fichier (la page demandée) et procède à son envoi (il s'agit de la réponse) . Parmi les serveurs les plus utilisés, on cite le serveur *Apache*, le serveur *NCSA* et le serveur *IIS*.



Partie 02: Introduction à l'algorithmique

I - Introduction

Le mot algorithme provient du nom d'un célèbre mathématicien arabe de la première moitié du IXe siècle: Muhammad ibn Musa al Khawarizmi.

Les algorithmes sont fondamentaux, ils aident à l'écriture des programmes (qu'ils sont leur traduction dans un langage compréhensible par l'ordinateur, exp: Pascal, Matlab..), ils sont indépendants à la fois de l'ordinateur qui les exécute, des langages dans lesquels ils sont énoncés et traduits.

II - La Notion D'algorithme.

1°) Définition de la notion d'algorithme.

C'est une suite finie d'opérations élémentaires constituant un schéma (diagramme) de calcul ou de résolution d'un problème.

Tout algorithme est caractérisé par :

- Un ensemble d'actions ou d'opérations à exécuter.
- Un ordre d'exécution de ces différentes opérations déterminé par la logique d'enchaînement et conditionné par les structures mises en œuvre.
- Un début et une fin.

Un algorithme est facile à comprendre, c'est pourquoi on utilise au préalable le langage algorithmique (proche du langage naturel), afin de décrire pas à pas une solution à un problème donné.

2°) **Syntaxe** : un algorithme a la forme suivante :

```

Algorithme Nom_Algo // entête de l'algo
Variables // déclaration des variables
Début // le début de L'algo
| ..... } Actions {Instructions}
| ..... }
| ..... }
Fin // la fin de l'algo
    
```

Exemple :

On veut calculer la moyenne des notes d'un étudiant dans un module donné, on suppose que le nombre des notes est égal à 3 :

Algorithme Moy

Variables

NomE, module : chaîne de caractères
 Moyenne, Note1, Note2, Note3 : réel

Début

- 1 : **Saisir** NomE, module, Note1, Note2, Note3
- 2 : **Calculer** la moyenne : Moyenne ← (Note1+Note2+Note3)/3
- 3 : **Afficher** NomE, module, Moyenne

Fin

Cette suite d'opérations qui permet de passer des données de base aux résultats correspond à un **algorithme**.

3°) **Représentation d'un algorithme : "Programmation"**

Pour un ordinateur, l'algorithme doit être décrit par un programme informatique. C'est à dire une suite d'instructions exprimées dans un langage de programmation.

Exemple : Ecrire un Algorithme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.



N.B : *L'algorithme prépare le travail de rédaction d'un programme*

ALGORITHME	PROGRAMME en Pascal
Algorithme Carré Variables nb, carr : Entier ; Début <i>' Entrée des informations élémentaires</i>	Program Carre; Var Nb, carr: Integer; Begin

Ecrire ("Entrez un nombre :") ; Lire (nb) ; <i>' Traitement des informations</i> carr ← nb * nb ; <i>' Sortie des informations</i> Ecrire ("Son carré est : ", carr) ; Fin	Write (" Entrez un nombre ?") ; Read (nb) ; carr := nb *nb; Write ("Son carré est : ", carr); End.
--	---

La mise en œuvre du traitement nécessite des variables que l'ordinateur est capable de les manipuler. Il s'agit des variables **nb** et **carr**.

Variable	Contenu
Nb	Un nombre saisi au clavier
Carr	Le carré d'un nombre donné

III - Les Eléments de base d'un algorithme :

Tout algorithme utilise des objets (variables) ou données élémentaires comme par exemple des littéraux (valeurs), des constantes ou des variables.

1°) Littéral.

C'est une valeur de type numérique ou alphanumérique.

Exemple : 11 ; 20.6; "Bonjour"

2°) Constante et variable.

Une constante est un objet qui ne peut pas être modifié par l'algorithme.

Une variable est un objet appelé à subir des transformations au sein de l'algorithme.

Constante et variable se caractérisent par :

- Un identificateur : nom de la variable ou constante qui ne doit pas contenir d'espaces.
- Une valeur : contenu de la variable ou constante
- Un type : domaine (entier, réel, logique) où la variable puise (prend) sa valeur.

Exemples : Constante: Pi=3.1416

Variables: Diamètre, circonférence : réel

Remarque : Une variable définie au début d'un algorithme correspond à une zone dans la mémoire identifiée par une adresse

3°) Types de variables existantes

Types	Ensemble de valeurs possibles	opérations
Booléen	Vrai, Faux	Comparaison (=, <, >, ..), NON, ET, OU
Caractère	'A', 'c'....	Comparaison, conversion
Entier	45 ; 123.....	+, -, /, *, DIV, Comparaison
Réel	12,345 ; 4,32	arithmétique : +, -, /, *. Comparaison.
Chaîne	"Bonjour monsieur"	Comparaison, longueur, extraction, concaténation

IV - Les Structures De Base

1°) L'instruction d'affectation :

Elle permet d'affecter ou d'initialiser une variable à partir du contenu d'une autre variable, d'une constante, d'un littéral, d'une expression arithmétique ou logique.

Le symbole utilisé est : \leftarrow ou := (le cas du Langage Pascal)

Exemples d'affectation :

```
Total := 826      ou   Total ← 826
PRIX ← Total
Somme ← Total+8
Somme ← Somme+50
```

Exercice : Que produit l'algorithme suivant ?

Algorithme exo

Variables A, B, C : Chaîne de Caractères

Début

A ← 'aze'

B ← 'rty'

C ← A & B

Fin. Re : cet algorithme fait la concaténation de 02 chaînes, le résultat est "azerty"

2°) Instructions d'entrée (saisie) et de sortie (affichage):

L'instruction d'entrée autorise la saisie de l'information à partir du clavier.

L'instruction de sortie autorise l'affichage des informations à l'écran.

a) Instruction d'entrée (la saisie) :

Lire (Nom_variable)

Consiste à affecter une valeur saisie à partir du clavier à une variable.

b) Instruction de sortie (l'affichage):

ECRIRE (Nom_variable)

Permet d'afficher la valeur d'une variable à l'écran.

Exemple : ECRIRE (" Taper deux nombres : ")

Lire (A, B)

Somme ← A+B

ECRIRE (" la somme est de : ", Somme)

Remarques:

- Les messages à afficher sont définies entre guillemets " ".
- On peut saisir plusieurs variables en même temps, séparées par des virgules.

3°) Les fonctions prédéfinies (Fonctions mathématiques)

Il existe bien entendu de nombreuses fonctions (procédures prédéfinies) nécessaires que l'on pourra utiliser en fonction des bibliothèques fournies par les divers langages de programmation, le tableau suivant montre quelques unes:

Notation mathématique	Pascal	Type de x	Type du résultat	Signification
-----------------------	--------	-----------	------------------	---------------

$ x $	ABS(x)	Entier ou réel	Type de x	Valeur absolue de x
x^2	SQR(x)	Entier ou réel	Type de x	Carré de x
$x^{1/2}$	SQRT(x)	Entier ou réel	Réel	Racine carré de x
$\sin x$	SIN(x)	Entier ou réel	Réel	sinus de x (x en radians)
$\cos x$	COS(x)	Entier ou réel	Réel	cosinus de x (x en radians)
$\arctg x$	ARCTAN(x)	Entier ou réel	Réel	Angle (en radians) dont la tangente vaut x
e^x	EXP(x)	Réel	Réel	Exponentielle de x
$\ln x$	LN(x)	Réel	Réel	Logarithme népérien de x
$[x]$	TRUNC(x)	Réel	Entier	Partie entière de x
$[x]$	INT(x)	Réel	Réel	Partie entière de x
arrondi de x	ROUND(x)	Réel	Entier	Entier le plus proche de x
décimal de x	FRAC(x)	Réel	Réel	Partie décimale de x

V - Les Structures de contrôle

a) La séquence :

Il s'agit de la suite d'actions qui se situent entre DébutFin :

```

Début
  ....
  ....
  .....
Fin
  } Actions en Séquence.

```

b) Le test (SialorsFinSi)

La structure conditionnelle permet un aiguillage des traitements.

Selon la valeur d'une expression booléenne, on pourra exécuter une suite d'actions I (instructions) ou une suite d'actions II.

La Syntaxe :

```

SI <expression logique> ALORS
    action1
SINON
    action2
Fin Si

```

Le résultat de l'expression logique (ou condition) est un booléen.

Quand l'expression logique est vraie alors la suite d'actions située après le mot **ALORS** (action1) est exécutée.

Si le résultat est faux, on exécute la suite d'actions située après le mot **SINON** (action2).

Exemple : une remise de 5 % est accordée si la somme des achats dépasse 100 DA.

```

SI Montant >100 ALORS
    Rem ← Montant*0,05
SINON
    Rem ← 0
Fin Si

```

Remarques :

-L'expression logique (condition) peut effectuer une comparaison entre plusieurs grandeurs. Elle utilise alors les opérateurs de comparaison : >, >=, <, <=, <>.

-Elle peut être complexe et peut faire intervenir les opérateurs logiques : ET, OU.

Exemple : (a>b) ET (a>c).

-Parfois la structure alternative peut être simple, et ne comporte pas la clause SINON.

```

SI <expression logique> ALORS
    actions
Fin Si

```

- On peut emboîter plusieurs structures alternatives dans certains cas de figure.

```

SI exp_log1 ALORS
    SI exp_log2 ALORS
        action1
    SINON
        action2
    Fin Si
SINON
    action 3
Fin Si

```

Le mot clé **FinSi** permet de lever toute ambiguïté.

Exemple : Afficher le plus grand de deux nombres entiers?.

Algorithme PlusGrand

Variables

a, b : Entier

Début

```

    ECRIRE ("Entrez deux nombres : ")
    LIRE (a, b)
    Si a>b Alors
        ECRIRE (" Le plus grand des deux est : ", a)
    Sinon
        Si a=b Alors
            ECRIRE (" Les nombres ",a," et ",b," sont égaux")
        Sinon
            ECRIRE (" Le plus grand des deux est : ",b)
        FinSi
    FinSi

```

Fin.

C) - La répétition (Boucles)

Un ensemble d'actions qui se répète toujours dans un ordre précis, un nombre déterminé de fois constitue un traitement itératif.

Un tel traitement est défini par une structure itérative qui définit la suite des opérations à effectuer ainsi que le nombre de répétitions.

Dans la plupart des cas, on ne connaît pas le nombre de répétitions, on précise alors le point d'arrêt de la répétition (boucle).

1°) Structure Tant queFaire..

Syntaxe :

```
Tant que <expression logique> faire
    Actions
Fin Tant Que
```

L'ensemble d'actions doit être exécuté tant que l'expression logique est vraie. Lorsque l'expression logique est fautive, le processus itératif s'arrête.

Phases d'exécution :

- 1. Evaluation de l'expression logique.
- 2. Résultat vrai : Exécuter les actions.
Reprise de l'étape 1.
- 3. Résultat faux : Arrêt de l'itération et le programme poursuit son exécution.

Exemple : Calculer la somme des n premiers nombres entiers?.

Algorithme Somme

Variables

s, i, n : Entier

Début

ECRIRE ("Entrer la valeur de n ")

Lire (n)

$s \leftarrow 0$

$i \leftarrow 1$

Tant que $i \leq n$ faire

$s \leftarrow s+i$

$i \leftarrow i+1$

FinTq

ECRIRE ("La somme des ", n ," premiers entiers est : ", s)

Fin

La variable S est utilisée pour additionner les différentes valeurs. Elle est initialisée à 0.

Compteur ou Indice

Cette instruction permet de faire évoluer ' i ' à la valeur suivante.

2°) La structure Répéter...jusqu'à..

Syntaxe :

```
Répéter
```

```
...
```

```
Actions
```

```
...
```

```
Jusqu'à <expression logique>
```

Phases d'exécution :

- 1. Le bloc d'actions est exécuté.
- 2. L'expression logique est testée
- 3. Dans le cas où elle est fausse, on recommence au point 1.
- 4. Dans le cas où elle est vraie. Le programme poursuit son exécution.

Dans le cas de la boucle répéter, la suite d'actions est exécutée au moins une fois, car le test de l'expression logique est effectué après exécution de l'ensemble d'actions.

Exemple : Traduire l'algorithme précédent en utilisant la structure : Répéter...Jusqu'à.
(Expression logique($I > N$))

Comparaison des deux structures :

-Dans la **structure répéter ..**, le test est placé en fin d'itération, par conséquent les actions répétitives sont exécutées au moins une fois.

-Dans la **structure Tant que..**, le test est placé avant le corps de la boucle. Dans certains cas, il est possible de ne pas exécuter aucune fois les opérations répétitives.

3°) La structure Pour ...Fin pour.

Cette structure permet de répéter un ensemble d'actions un nombre connu de fois.

Syntaxe :

```

Pour Nom_var ← valeur-initiale à valeur-finale faire
    Actions
    [Auto-incrémentation]
Fin pour

```

Nom_var est la variable d'énumération des répétitions. Elle sera initialisée à la valeur de début, l'ensemble des actions sera exécuté, elle passera automatiquement à la valeur suivante jusqu'à la valeur finale.

Exemple :

Algorithme Compteur

Variables s,i, : Entier

Début

```

    S ← 0
    Pour i ← 1 à 10 faire
        S ← S+i
        { i suivant}
    Fin pour

```

ECRIRE ("La somme des 10 premiers entiers est : ", S)

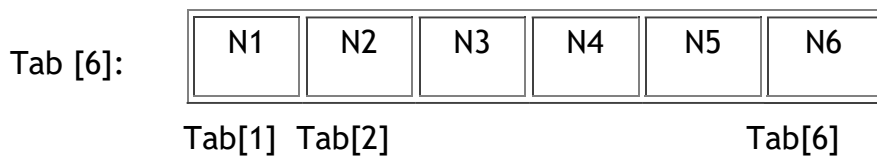
Fin

VI - Les Tableaux

Imaginons que, dans un programme, nous ayons besoin simultanément de 6 valeurs (par exemple, des notes pour calculer une moyenne). Evidemment, la seule solution dont nous disposons à l'heure actuelle consiste à déclarer six variables, appelées par exemple N1, N2, N3, etc. Mais cela ne change pas fondamentalement notre problème, car arrivé au calcul, et après une succession de six instructions « Lire » distinctes, cela donnera :

$$\text{Moy} \leftarrow (N1+N2+N3+N4+N5+N6)/6$$

Et pour simplifier les choses on peut rassembler toutes ces variables en une seule (un tableau ou variable indexée), au sein de laquelle chaque valeur sera désignée par un numéro:



Exemple : écrire un algorithme qui lit 05 lettres puis les affiche en utilisant un tableau.

Algorithme TabCar

Variabes tab[5]: Tableau de Caractère

i: Entier, c: Caractère

Début

Pour i<--1 à 5 **faire**

Ecrire('Donner la',i,'emme lettre :')

Lire(c)

tab[i] <-- c

FinP

Pour i<--1 à 5 **faire**

Ecrire('la',i,'emme lettre était :',tab[i])

FinP

Fin.

Exercice : écrire un algorithme qui lit 10 entiers (en utilisant un tableau) puis qu'il affiche les nombres pairs ?

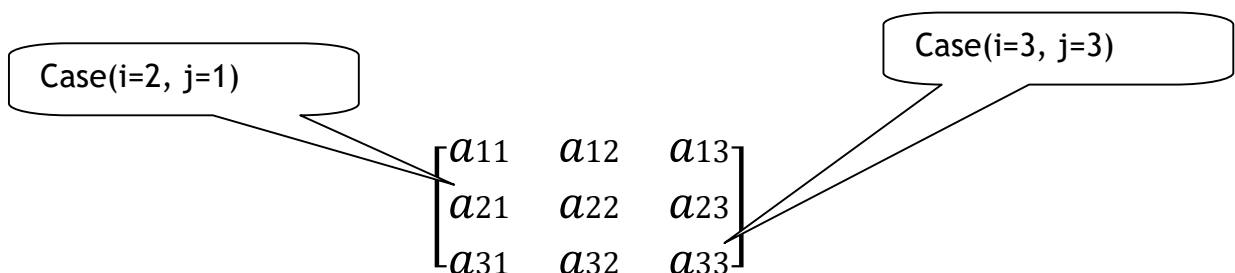
VII - Les Tableaux Multidimensionnels (Matrices)

L'informatique nous offre la possibilité de déclarer des tableaux dans lesquels les valeurs ne sont pas repérées par une seule, mais par **deux coordonnées**.

Un tel tableau se déclare ainsi :

Case[3, 3] : Tableau de réels

Cela veut dire : réserve-moi un espace de mémoire pour 3 x 3 entiers, et quand j'aurai besoin de l'une de ces valeurs, je les repèrerai par deux indices (comme Excel, la seule différence étant que pour les coordonnées, on n'utilise pas de lettres, juste des chiffres).



Remarque : dans les tableaux à deux dimensions le premier indice "i" représente les lignes et le deuxième "j" les représente colonnes

Exemple

La somme de deux matrices Mat1[i,j] et Mat2 [i,j], le résultat est Mat3[i,j]. (i=j=1..2)

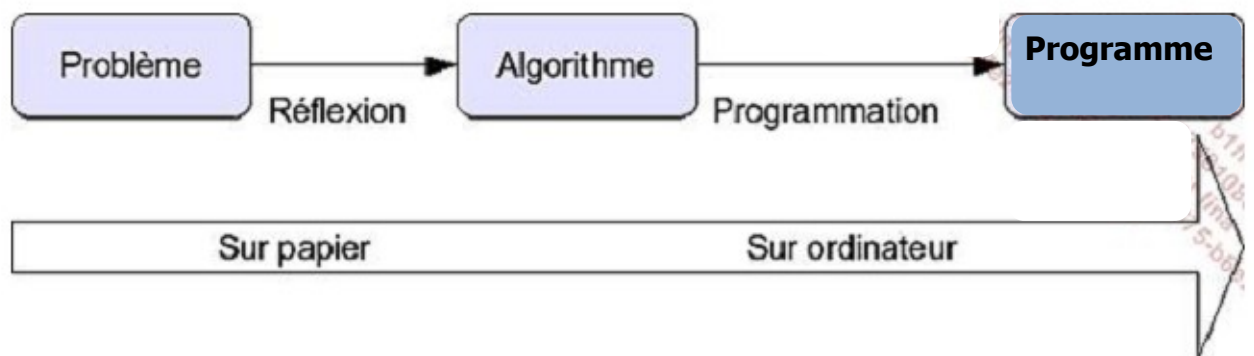
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

Mat1 + Mat2 = Mat3

- 1- La lecture des deux matrices Mat1 & Mat2 ;
- 2- Le calcul de Mat1+ Mat2 ;
- 3- L'affichage du résultat Mat3

VIII Le processus de résolution d'un problème

Quelque soit le domaine, la démarche de **résolution d'un problème** (conception du programme) reste identique :



De la réflexion à la programmation

Les différentes étapes de résolution d'un problème sont :

1. **Etude Préalable** : Compréhension du problème, réflexion sur le problème.
2. **Spécification des données et des résultats** : Recenser les informations et préciser leur nature.
3. **Spécification de fonctionnalités** : Recenser et préciser les fonctionnalités désirées.
4. **Solution en langage naturel** : Savoir résoudre le problème avant d'automatiser la solution (dessiner un diagramme).
5. **Données structurées/Algorithme** : Mise en forme algorithmique des informations et des traitements à réaliser.
6. **Programmation** : Choix du langage, traduction de la solution (écrire un algorithme) sous forme de programme.
7. **Programme exécutable** : Compilation du programme en programme exécutable
8. **Test et évaluation du travail réalisé** : Test de la cohérence par rapport aux spécifications (tester les fonctionnalités).

IX Elaboration d'un algorithme complet :

La résolution d'une équation du second degré $ax^2 + bx + c = 0$ en langage Pascal.