

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique



Université Mustapha Stambouli Mascara

Faculté des Sciences Exactes
Département de Physique

وزارة التعليم العالي
والبحث العلمي

جامعة مصطفى اسطمبولي
ماسكارا

كلية العلوم الدقيقة
قسم الفيزياء

Cours : Informatique 1

Bureautique & Technologie Web

Introduction à l'algorithmique

Dr KINANE DAOUADJI AMINA

Bureautique & Technologie Web

1. Bref historique de l'évolution de l'informatique
2. Architecture du PC
3. Les différents composants matériels du PC (hardware)
4. Les systèmes d'exploitation (software)
5. Les logiciels : MS Office : Word, Excel, Power point
6. Les réseaux informatiques
7. Les différentes topologies d'un réseau informatique
8. Fonctionnement client-serveur
9. Le réseau internet et le web

Introduction à l'algorithmique

1. Notion d'algorithmique : définition, syntaxe
2. Les éléments de base d'un algorithme
 - 2.1 Les instructions d'entrées sorties
 - 2.3 Les structures de contrôle
 - 2.3 Les tableaux
3. Processus de résolution d'un problème
4. Elaboration d'un algorithme complet
 - 4.1 Résolution d'une équation du second degré

Le but de ce cours est de permettre aux étudiants de 1ère année sciences de la matière non seulement de se familiariser avec l'outil informatique (matériel et logiciel) mais aussi d'acquérir les notions de base sur la programmation et les réseaux en général.

Coefficient = 2 / Crédits = 4

Table des matières

I.	Bureautique & Technologie Web	6
I.1	Bref historique de l'évolution de l'informatique	6
I.1.1	Définition de l'informatique	6
I.1.2	Définition d'un ordinateur	6
I.1.3	Quelques machines mécanique et automatique	7
I.1.3.1	L'abaque :	7
I.1.3.2	La Calculatrice de Schickard	7
I.1.3.3	La Pascaline	8
I.1.3.4	Step Reckoner	8
I.1.3.5	La machine de Babbage	8
I.1.3.6	ENIAC « Electronic Numerator Integrator and Computer »	9
I.1.4	Le langage machine	10
I.1.5	Les différentes Catégories d'ordinateurs	11
I.2	Architecture de l'ordinateur (aspect hard) - Les différents composants matériels du PC (hardware)	12
I.2.1	Architecture de Von Neumann	12
I.3	Les différents composants matériels du PC (hardware)	13
I.3.1	La partie matérielle	13
I.3.2	Périphériques d'entrée et périphériques de sortie	14
I.3.3	Périphériques de stockage	14
I.4	Les systèmes d'exploitation	15
I.4.1	Le rôle d'un système d'exploitation	15
I.4.2	Les composantes du système d'exploitation	16
I.4.3	Systèmes multitâches & Systèmes mono tâches	16
I.4.4	Système d'exploitation DOS :	16
I.4.5	Système d'exploitation Windows :	17
I.4.6	Système d'exploitation UNIX :	17
I.4.6.1	Quelques versions d'UNIX et Linux	18
I.5	Les logiciels : MS Office : Word, Excel, Power point	19
I.5.1	Microsoft Word	19
I.5.2	Microsoft Office Excel	19
I.5.2.1	Excel et fonctions de calculs :	19
I.5.3	Microsoft Office Power Point	20
I.5.3.1	Mode de conception	20
I.5.3.2	Mode de présentation	20

1.5.3.3	Les principaux onglets de Power point	21
I.6	Les réseaux informatiques	22
1.6.1	Classification des réseaux informatiques	22
1.6.2	Les différentes applications du réseau informatique.....	23
1.6.3	Les différents types de réseau informatique	23
1.6.4	Le modèle OSI	24
1.6.5	Adresse IP (internet Protocol) et numéro de Port.....	25
I.7	Topologies et fonctionnement d'un réseau.....	27
1.7.1	Topologie physique.....	27
1.7.1.1	Topologie en bus.....	27
1.7.1.2	Topologie en étoile.....	27
1.7.1.3	Topologie en anneau	28
1.7.1.4	Topologie en arbre ou hiérarchique.....	29
1.7.1.5	Topologie maillée.....	30
1.7.1.6	Topologie hybrid	30
1.7.2	Topologie logique	31
1.7.2.1	Ethernet.....	31
1.7.2.2	Token Ring.....	31
1.7.2.3	Fiber distributed data interface FDDI	31
I.8	Fonctionnement client serveur	32
1.8.1	Les caractéristiques du modèle client-serveur	33
1.8.2	Exemple d'utilisation	33
1.8.3	Avantages du modèle client-serveur	33
1.8.4	Inconvénient du modèle client-serveur	33
I.9	Le réseau internet et le web.....	34
1.9.1	La World Wide Web.....	34
1.9.2	Les protocoles utilisés sur internet	34
1.9.3	Les moteurs de recherches et navigateur internet.....	34
1.9.4	Page web et lien hypertexte.....	34
1.9.5	Le langage HTML.....	35
1.9.5.1	Structure d'un document HTML.....	35
1.9.5.2	Les principales balises du Langage HTML :	35
II.	Introduction à l'algorithmique	36
II.1	Notion d'algorithmique	36
II.1.1	Définition 1.....	36
II.1.2	Définition 2.....	36

II.1.3	Les caractéristiques d'un algorithme	36
II.1.4	Un programme	36
II.1.5	Langage de programmation	37
II.1.6	Un compilateur	37
II.1.7	Les étapes de mise en point d'un programme	37
II.1.8	Les éléments de base d'un algorithme	37
II.1.9	Syntaxe d'un algorithme	38
II.1.10	Les variables et les constantes	38
II.1.10.1	Les variables	38
II.1.10.2	Les constantes	38
II.1.10.3	Les opérateurs :	39
II.2	Les instructions d'entrées/sorties	40
II.3	Exemples	40
II.4	Les structures de contrôle	40
II.4.1	Bloc d'instruction	40
II.4.2	Les alternatives	41
II.4.3	les boucles	41
II.5	Les fonctions et procédures	43
II.5.1	Les fonctions	43
II.5.1.2	Appelle de fonction	43
II.5.2	Les procédures	43
II.5.2.1	Syntaxe d'une procédure	43
II.6	Les structures de données	44
II.6.1	Les tableaux à une dimension (vecteur)	44
II.6.2	Les tableaux à deux dimensions (matrice)	45
II.7	Processus de résolution d'un problème	45
II.7.1	Les différentes étapes du processus de résolution d'un problème	46
II.7.2	Elaboration d'un algorithme complet	46
II.7.2.1	Résolution d'une équation du second degré	46
II.8	Exercices avec correction	49

I. Bureautique & Technologie Web

Cette partie du programme concerne la manipulation de l'outil informatique avec l'assimilation des connaissances sur le web et les réseaux informatique.

I.1 Bref historique de l'évolution de l'informatique

Nous utilisons le concept de l'informatique dans notre vie quotidienne d'une manière directe ou indirecte, mais nous ignorons souvent ses origines et les circonstances de son développement. Cette technologie, qui était au début juste un loisir réservé aux hommes de sciences et quelques amateurs, est devenue actuellement un enjeu économique et social très important. La science de l'informatique est indissociable des ordinateurs, et bien-sûr, un ordinateur peut prendre différentes formes de machines ; d'un simple Smartphone aux supercalculateurs en passant par les PCs et les serveurs, donc pour comprendre cette science il faut comprendre cette machine.

I.1.1 Définition de l'informatique

L'informatique ou en anglais « computer science », est la science du traitement de l'information, sachant qu'une information peut prendre différentes formes : du texte, des images, du son, des vidéos, ... etc. Ce traitement est effectué à l'aide d'une machine baptisée "ordinateur".

I.1.2 Définition d'un ordinateur

Un ordinateur est un dispositif électronique programmable conçu pour recevoir des données comme entrées, d'effectuer des opérations mathématiques et logiques prescrites à grande vitesse, et de fournir les résultats de ces opérations comme sorties. Exemples : Les ordinateurs centraux, les ordinateurs de bureau et portables, les tablettes et les smartphones.

En plus simple, l'histoire de l'informatique est l'histoire des sciences impliquées dans le traitement rationnel de l'information sous les différentes formes qu'elle peut prendre, automatiser ce traitement était le rêve des anciens mathématiciens et physiciens. Donc, le concept et l'idée centrale de l'informatique a commencé avant même l'invention des premiers ordinateurs, c'est plutôt avec les premiers algorithmes que l'informatique a vu le jour, dans son état embryonnaire.

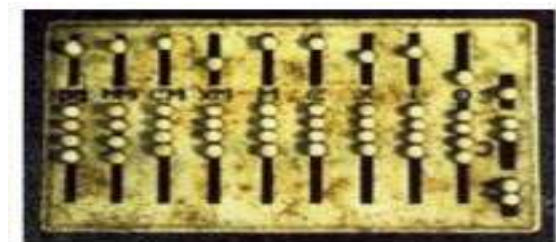
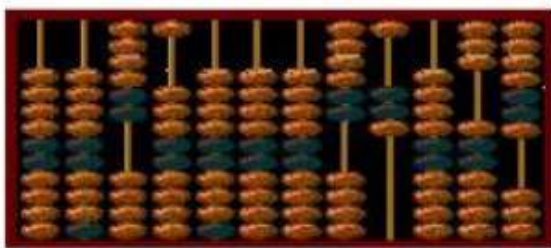
Le développement des moyens de calcul a débuté par la volonté humaine d'automatiser certaines tâches fastidieuses réalisées longtemps à la main.

I.1.3 Quelques machines mécanique et automatique

I.1.3.1 L'abaque :

C'est un dispositif destiné aux calculs arithmétiques, il est sous forme de planche avec des sillons dans lesquels on fait glisser des jetons, ou sous forme de bouliers avec tiges (le nombre de tiges est variable) sur lesquelles des boules ou des billes sont déplacées.

Prenant comme exemple l'abaque des Romains :



Abaque romain avec sillons

I.1.3.2 La Calculatrice de Schickard

En 1624 Wilhelm Schickard construit un prototype d'une machine à calculer capable de faire des additions et des soustractions sur des nombres jusqu'à 6 chiffres. , mais celui-ci fut détruit dans un incendie avant d'avoir été terminé.

En 1960 le mathématicien Bruno Von Freytag a réalisé la machine de Schickard en se basant sur ses plans. Cette réalisation a révélé que les plans étaient incorrects.



La Calculatrice de Schickard

I.1.3.3 La Pascaline

En 1642 le philosophe et mathématicien Blaise Pascal a décidé d'aider son père, un surintendant et percepteur de taxes, pour faire ses calculs, en inventant une machine capables d'effectuer des additions et des soustractions. La Pascaline ou « roue de pascal » est considérée comme la première calculatrice mécanique. Malheureusement cette invention n'a pas connu de succès commercial en raison de son prix.



La Pascaline

I.1.3.4 Step Reckoner

En 1671, le mathématicien et philosophe allemand Gottfried Wilhelm von Leibniz a conçu le désigne d'une machine à calculer appelée Step Reckoner basée sur les idées de Pascal, cette dernière, a été construite pour la première fois en 1673.



Step Reckoner

I.1.3.5 La machine de Babbage

En 1812 ou 1813, le mathématicien Charles Babbage a eu l'idée de calculer mécaniquement des tables mathématiques. Cette fameuse machine sous forme d'une petite calculatrice était capable d'effectuer certains calculs mathématiques jusqu'à huit décimales. En 1823, Babbage a obtenu le soutien du gouvernement pour la conception d'une machine baptisée « Difference Engine », d'une capacité de calcule jusqu'à 20 décimales.



La machine de Babbage

Jusqu'à maintenant nous avons cité que quelques machines mécaniques ou semi automatiques inventées pour le traitement de l'information, et qui nécessitent toujours l'intervention de l'utilisateur.

Nous allons présenter par la suite le premier véritable ordinateur numérique (ENIAC).

I.1.3.6 ENIAC « Electronic Numerator Integrator and Computer »

Qui a été inventé en 1945 aux états unis sous la direction de J. Presper Eckert et John Mauchly, il représente le premier calculateur électronique programmable, capable d'exécuter tout algorithme, mais n'empêche que son mode de programmation été primitif.



Electronic Numerator Integrator ENIAC

En 1981 : IBM a lancé le PC « Personal Computer" qui signifie " ordinateur personnel ». En 1984, les systèmes Macintosh d'Apple Computer sont les premiers à être dotés d'une souris

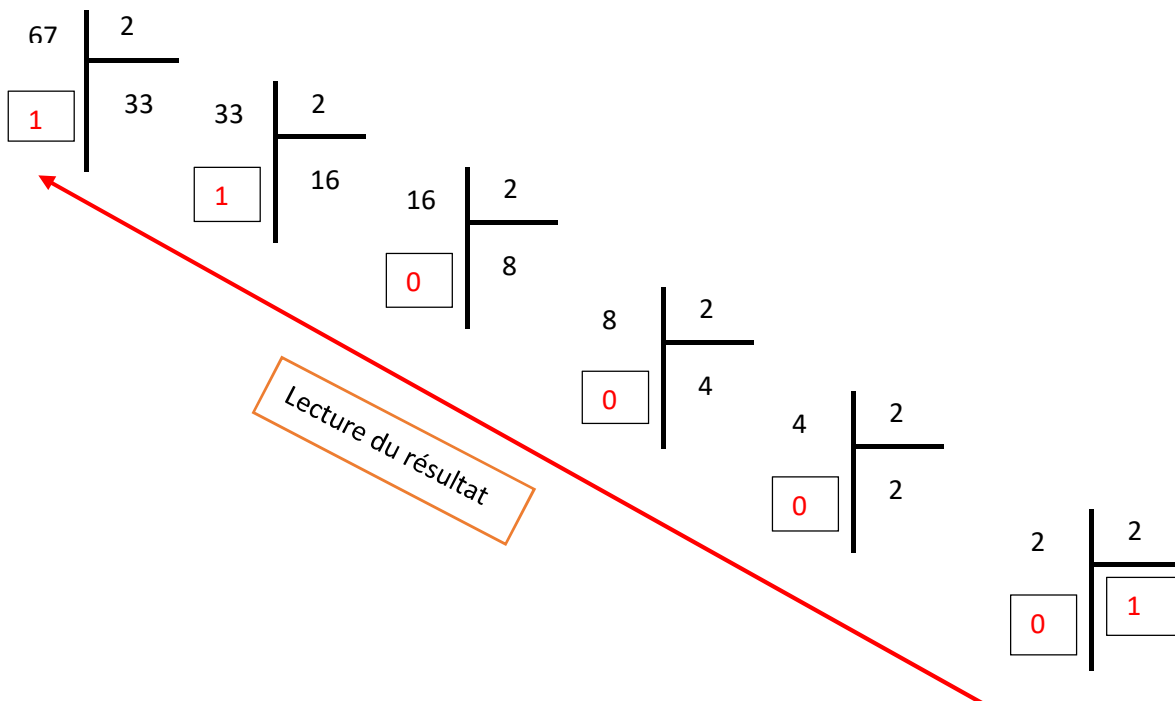
et d'une interface graphique au lieu d'avoir à taper des commandes au clavier. En 1985 la première version de Windows a été créée.

1.1.4 Le langage machine

En réalité, un ordinateur ne peut traiter que le code binaire, toute information (texte, image, vidéo, ... etc.) doit être convertie en langage machine, une suite de **zéro** et de **un**, qu'on appelle « bit ». Le bit ne peut prendre qu'une seule valeur à la fois ; zéro ou un, il représente l'unité la plus simple dans un système de numérotation.

Exemple : le chiffre $(67)_{10}$ est égal à $(1000011)_2$ en langage binaire.

- Convertir le nombre 67 du décimal au binaire



- Convertir le nombre (1000011) du binaire au décimal

1	0	0	0	0	1	1
2^6 ×	2^5 ×	2^4 ×	2^3 ×	2^2 ×	2^1 ×	2^0 ×

$$= 1 \cdot (1) + 1 \cdot (2) + 0 \cdot (4) + 0 \cdot (8) + 0 \cdot (16) + 0 \cdot (32) + 1 \cdot (64)$$

$$= (67)_{10}$$

Le langage machine est normalement affiché sous forme hexadécimale afin qu'il soit un peu plus facile à lire.

Exemple de chiffres en hexadécimal : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F.

Le langage d'assemblage est presque le même que le langage machine, sauf que les instructions, les variables et les adresses ont des noms au lieu de simples nombres hexadécimaux.

I.1.5 Les différentes Catégories d'ordinateurs

- **Ordinateur de bureau** : utilisé sur un bureau ou tout autre endroit fixe, destiné à un usage personnel ou professionnel. Les tâches qu'on peut effectuer sur un ordinateur de bureau sont beaucoup plus simples par rapport aux autres catégories avancées.
- **Ordinateur central** : est un type d'ordinateur à grande vitesse de traitement et qui sert d'unité centrale à un ensemble de terminaux (machines), on peut dire qu'il est utilisé pour le traitement des transactions (la lecture et l'écriture de disque, les appels au système d'exploitation, ... Etc.)
- **Les Supers ordinateurs ou calculateurs** : ce sont de grands systèmes avec une grande vitesse de calculs, spécialement conçus pour résoudre des défis scientifiques et industriels complexes (modèle météorologique, simulation financière, ... Etc.).

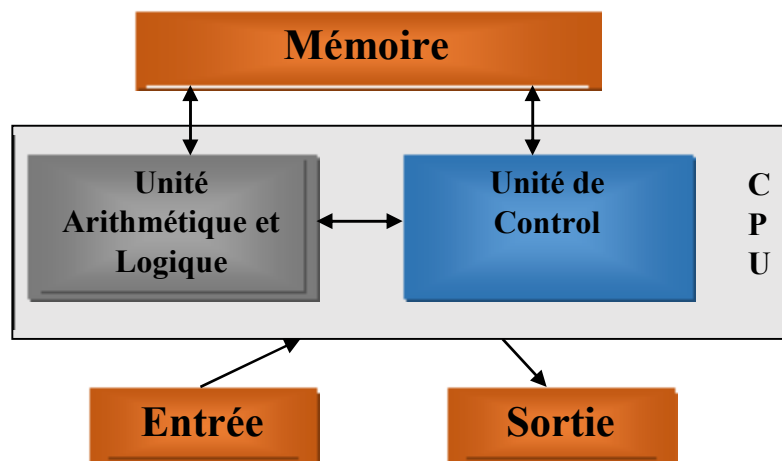
Donc ,l'évolution de l'informatique résulte de l'intersection entre les différentes découvertes, et les transformations techniques et sociales, on peut citer par exemple les découvertes physiques sur les semi-conducteurs, les progrès sur la miniaturisation des transistors et leurs impacts sur la conception des premiers ordinateurs personnels, les découvertes mathématiques sur la calculabilité et la théorie de l'information et leurs impacts sur l'algorithmique, on a aussi comme transformations techniques, l'introduction de la machines ou de composants informatiques, comme les machines à laver, les téléviseurs, les automobiles et les avions, et on a aussi les transformations sociales avec l'organisation des entreprises et des administrations autour de leurs systèmes d'information automatisés, où tout est représenté sous forme d'information.

1.2 Architecture de l'ordinateur (aspect hard) - Les différents composants matériels du PC (hardware)

Tous les ordinateurs sont des «machines de Von Neumann», du nom du grand mathématicien qui a défini leur architecture en 1945, tout simplement on peut dire que l'architecture d'un ordinateur est un ensemble de composants tel que l'unité d'entrée, l'unité de sortie, l'unité de control et bien d'autres... conçu pour simplifier des taches compliquées ou difficiles à résoudre.

1.2.1 Architecture de Von Neumann

Le concept de base de l'architecture de Von Neumann se base sur l'idée du stockage des instructions du programme en mémoire ainsi que les données sur lesquelles ces instructions opèrent. L'architecture de Von Neumann est composée de trois sous-systèmes distincts : une unité centrale de traitement (CPU), une mémoire et des interfaces d'entrée /sortie (E / S). Le schéma suivant illustre l'architecture de Von Neumann :



Architecture de Von Neumann

- La mémoire, est utilisée pour stocker les instructions et les données du programme.
- Le CPU, qui peut être considéré comme le cerveau du système informatique, il comprend les deux composants suivant :
 - Unité de Control : détermine l'ordre dans lequel les instructions doivent être exécutées
 - Unité Arithmétique et logique ou unité de traitement : chargée d'exécuter les instructions qui manipulent les données.

- c. interfaces d'entrée/sortie (E/S ou I/O) qui permettent de communiquer avec le monde extérieur.

I.3 Les différents composants matériels du PC (hardware)

L'ordinateur peut être divisé en deux parties : Matérielles et Logiciels

I.3.1 La partie matérielle

- **La carte mère (Motherboard)** : Elle assure la connexion entre tous les composants situés à l'intérieur du boîtier de l'ordinateur.
- **Le processeur (CPU)** : Le processeur est souvent considéré comme le «moteur» ou le «cerveau» de l'ordinateur, il permet de manipuler et de traiter les données qui lui sont fournies, sa vitesse d'exécution est influencée par sa puissance, exprimée en Gigahertz (GHz).
- **La mémoire :**
La mémoire de l'ordinateur est un périphérique physique capable de stocker temporairement ou de manière permanente les informations.
- **La mémoire vive (RAM)** : mémoire à accès aléatoire "Random Access Memory", est une mémoire volatile qui stocke les informations d'une façon temporaire, l'espace mémoire est libéré après la fermeture des applications et programmes. Elle est vidée complètement à chaque arrêt ou redémarrage du PC.
- **La mémoire morte (ROM)** : mémoire non volatile "read only memory" permet de conserver les données nécessaires au démarrage de l'ordinateur (BIOS), elle ne s'efface pas lors de la mise hors tension de la machine.
- **Le disque dur interne (hard disk)** : le disque dur permet de stocker les données d'une façon permanente, donc le disque dur est le composant qui conserve les systèmes d'exploitations (UNIX, WINDOWS..), les logiciels, les documents...Etc.
- **Le Bloc d'alimentation (Power Supply Unit):**
Le bloc d'alimentation alimente en énergie électrique chaque composant du PC, il est chargé de transformer le courant « alternatif » d'entrée en courant de sortie «continu», la tension d'entrée (220v ou plus) sera donc transformée en une tension de sortie compatible selon la machine et selon le composant (ex: 12v, 5v, 3.3v,). Le bloc d'alimentation a donc deux fonctions : la première est de permettre de fournir au PC une alimentation continue, et la deuxième est de réduire la tension.

- **La carte graphique (GPU)** : Elle permet d'assurer le fonctionnement des paramètres visuels du PC, et de transmettre les données traitées sous forme d'images à l'écran, elle est dotée d'un processeur dédié et d'une mémoire.
- **La carte son (audio card, sound card)**: C'est le composant qui permet de gérer les entrées-sorties sonores de l'ordinateur, (microphone, haut-parleurs) .
- **Le lecteur/graveur** : C'est le composant qui permet de lire les données qui se trouvent sur les CD et les DVD (photos, logiciels, jeux, musique ...Etc.), et dans sa fonction graveur, il permet de stocker (graver) les données sur les CD et les DVD.
- **Les cartes contrôleurs et ports** : Les cartes contrôleurs assurent la connexion entre les différents périphériques déjà cités dans ce cours et l'unité centrale grâce à des ports de sortie. La carte contrôleur est fixée sur un slot de la carte mère. Et les informations circulent dedans via ce qu'on appelle un bus. Exemple : carte contrôleurs SATA2, carte contrôleur PCI-E LP,...

1.3.2 Périphériques d'entrée et périphériques de sortie

Les périphériques d'entrée assurent la fourniture des informations au système informatique, tandis que les périphériques de sortie permettent au système informatique d'afficher ou de sortir les informations déjà traitées

Les périphériques d'entrée : le clavier, la souris, le scanner, l'appareil photo numérique, le micro...

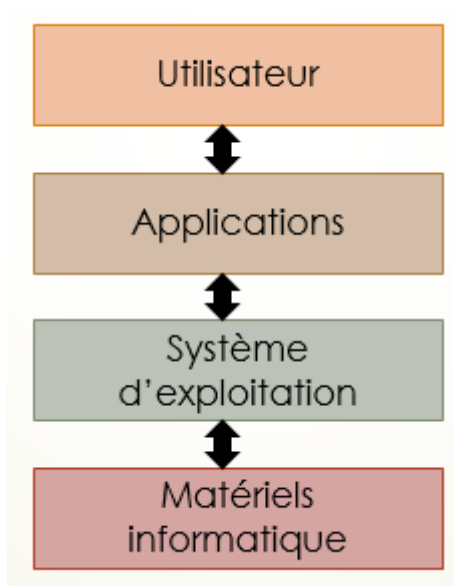
Les Périphériques de sortie : l'écran, l'imprimante, les haut-parleurs...

1.3.3 Périphériques de stockage

On peut citer : le disque dur externe, la clé USB, la carte mémoire, le CD/DVD...

1.4 Les systèmes d'exploitation

Un système d'exploitation ou en anglais operating system (OS), est un programme qui contrôle l'exécution de tous les autres programmes ou applications installées sur la machine, il agit comme un intermédiaire entre l'utilisateur et le matériel informatique.



1.4.1 Le rôle d'un système d'exploitation

- **Contrôle l'accès au matériel**

(Plug and Play "PnP") ou « connecter et jouer » ou « brancher et utiliser » c'est-à-dire que le système d'exploitation "OS" découvre et configure automatiquement le matériel.

- **Gestion des fichiers et des dossiers**

Le système d'exploitation permet d'organiser, copier, supprimer et déplacer des fichiers et dossiers, il permet aussi de restaurer les fichiers et les dossiers supprimés dans leur emplacement d'origine, de modifier les paramètres du système à l'aide du Panneau de configuration et d'effectuer d'autres tâches de gestion du système.

- **Interface graphique :**

Le système d'exploitation permet aux utilisateurs d'interférer (interagir) avec des appareils électroniques (appareil photos, imprimantes, scanner,..) à travers des icônes graphiques et des indicateurs visuels.

- **Gestion des applications :**

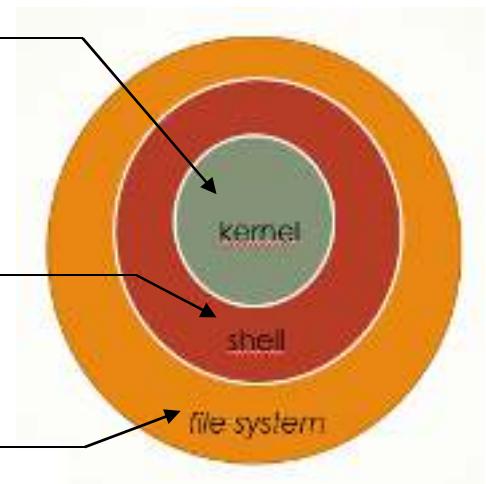
Le système d'exploitation est chargé de la bonne exécution et gestion des applications, son rôle est d'affecter aux applications les ressources nécessaires pour un bon fonctionnement.

- **Gestion des droits d'accès :**

La sécurité liée à l'exécution des programmes est assurée par le système d'exploitation, ce dernier garanti que les ressources ne doivent être utilisées que par les programmes et utilisateurs possédant les droits d'accès.

1.4.2 Les composants du système d'exploitation

- **Kernel ou noyau** il représente les fonctions fondamentales du système d'exploitation par exemple : la gestion de la mémoire, des fichiers, des entrées-sorties.
- **Shell ou interpréteur de commande**, il permet la communication avec le système d'exploitation par l'intermédiaire d'un langage de commande.
- **File system ou système de fichier** il permet l'enregistrement et l'indexation des fichiers dans une arborescence, Exemple : C:\Users\PC\Desktop



1.4.3 Systèmes multitâches & Systèmes mono tâches

Un système d'exploitation est dit multitâche ou en anglais «multitasking» lorsque plusieurs tâches ou processus peuvent être exécutées simultanément.

Un système d'exploitation mono tâche exécute une tâche seulement lorsque la précédente est terminée, donc le système ne gère qu'une seule activité à la fois.

1.4.4 Système d'exploitation DOS :

Le DOS est le premier système d'exploitation utilisé avec les ordinateurs, sa version la plus exploitée est celle de Microsoft, baptisée MS-DOS " Microsoft Disk Operating System", c'est un système mono tâche, il permet d'interpréter les commandes saisies au clavier par l'utilisateur (la gestion des fichiers et des dossiers, la configuration du matériel, l'exécution des programmes...Etc.).

I.4.5 Système d'exploitation Windows :

Windows (fenêtres en français), est le système d'exploitation le plus répandu et celui qui a succédé le système MS-DOS, développé par Microsoft Corporation pour être exécuté sur des ordinateurs personnels (PC).

Version	Année	Type
Windows 11	Septembre 2021	Multi tâches
Windows 10 version 20H2	Octobre 2020	Multi tâches
Windows 10	Juillet 2015	Multi tâches
Windows 8.1	Octobre 2013	Multi tâches
Windows 8	Octobre 2012	Multi tâches
Windows 7	Octobre 2009	Multi tâches
Windows vista	Janvier 2007	Multi tâches
Windows xp professionnel x64	Avril 2005	Multi tâches
Windows XP	Octobre 2001	Multi tâches
Windows ME (millenium)	Septembre 2000	Multi tâches
Windows 2000	Février 2000	Multi tâches
Windows 98	Juillet 98	Multi tâches
Windows 95	Aout 95	Multi tâches
Windows 1.0	Novembre 85	Mono tache

I.4.6 Système d'exploitation UNIX :

Unix est un système d'exploitation multi tache, multiutilisateurs conçu en 1969 aux Bell Labs (AT&T) American Telephone and Telegraph Company, ce système est protégé par un copyright, très utilisé dans les grandes entreprises, dans les serveurs Internet, les postes de travail et les PC.

La version open source Linux représente un clone du système d'exploitation Unix, développé par un étudiant finlandais Linus Torvalds en 1991. Linux peut être installé sur une grande variété de matériel informatique, téléphone mobile, tablette, PC....

La plupart des systèmes d'exploitation UNIX ne sont pas libres, par contre Linux est Open-source est **souvent** gratuit, il peut être téléchargé à partir d'Internet.

1.4.6.1 Quelques versions d'UNIX et Linux

UNIX	Linux
HP-UX	Redhat Enterprise Linux
IBM AIX	Fedora Linux
Solairs Sun	Debian Linux
Mac OS X	Suse Enterprise Linux
IRIX	Ubuntu Linux

1.5 Les logiciels : MS Office : Word, Excel, Power point

Ms office est une suite bureautique développée par Microsoft Corporation qui comprend Microsoft Word, Excel, Access, Publisher, PowerPoint, Outlook et d'autres programmes plus avancés. Chaque programme a une fonction différente. La suite des programmes est compatible avec le système d'exploitation Windows et Macintosh.

1.5.1 Microsoft Word

Microsoft Word, est un logiciel de traitement de texte ayant comme but d'effectuer des transformations sur du texte, ces transformations peuvent être des changements : (de couleur, de taille, d'alignement,... etc.) en vue d'une utilisation spécifique, comme par exemple : écrire une demande, réaliser une page de garde pour un mémoire, ... Etc., elle est constituée d'une barre d'accès rapide, ruban, barre d'état, zone de travail, barre d'insertion, règle, ...Etc.

- **La barre d'accès rapide**

Est une barre d'outils personnalisable qui contient un ensemble de commandes qui sont indépendantes de l'onglet ruban, elle contient, la commande enregistrer, annuler frappe, répéter frappe...

- **Le ruban**

Est un bandeau qui regroupe toutes les fonctions du logiciel, Police, Paragraphe, Style,...Etc.

- **Barre de titre**

La barre qui contient le titre du document.

- **Barre d'état**

Située dans le bas de la fenêtre Word, elle affiche des informations comme le nombre des pages, colonnes, le Zoom....Etc.

1.5.2 Microsoft Office Excel

Excel est un logiciel d'analyse de données (Tableur), et comme son nom l'indique, c'est un classeur qui regroupe plusieurs feuilles de calculs. Avec ce logiciel on peut réaliser des tableaux, trier des données, et même créer des fonctions de calcul.

1.5.2.1 Excel et fonctions de calculs :

Il existe des fonctions prédéfinies dans ce logiciel comme :

- **la fonction "somme", et la fonction "moyenne" :**

= somme (sélectionner les cellules des données)

= moyenne (sélectionner les cellules des données)

- **La fonction " min" et la fonction "max" :**

= min (sélectionner les cellules des données)

= max (sélectionner les cellules des données)

- **Le OU et le ET logique**

=OU (valeur logique 1; valeur logique 2...)

=ET (valeur logique 1; valeur logique 1...)

- **Le SI conditionnel :**

Le SI est une fonction, utilisée pour représenter une solution lorsqu'on a un choix :

= SI (Test logique ; valeur si vrai ; valeur si faux)

- Exemple 1: si la moyenne est supérieure à 10 alors « admis » sinon « ajourné ».

= SI (Moy >10; "admis" ; "ajourné").

- Exemple 2 : si le grade du diplôme est Bac1 et la moyenne est supérieur ou égale à 10 alors « admis » sinon « ajourné ».

=SI (Et (grade="Bac1";Moy>=10);"admis";"ajourné")

1.5.3 Microsoft Office Power Point

Est un logiciel de présentation assistée par ordinateur, il permet de préparer des présentations contenant du texte, des images, des effets visuels, du son...etc.

Il y a deux modes de fonctionnement :

1.5.3.1 Mode de conception

Permet de concevoir des vues en ajoutant des textes et des images.

1.5.3.2 Mode de présentation

Permet de voir le résultat du travail de conception sous forme de slide (diapositive), la touche de fonction « F5 » permet de passer au mode présentation, et la touche « Esc » permet de revenir en mode conception.

1.5.3.3 Les principaux onglets de Power point

- **Création** : permet de choisir les fonds, les couleurs, la taille de police.
- **Animation** : permet de choisir les transitions entre deux diapositives, ou des animations personnalisées sur une diapositive.
- **Diaporama** : permet de régler la façon dont le diaporama sera visualisé lors de la présentation.

1.6 Les réseaux informatiques

Les réseaux informatiques font référence aux différents appareils informatiques connectés et qui communiquent entre eux grâce à des liens physiques (filaire) ou sans fils, on peut citer comme appareils : les ordinateurs portables, les ordinateurs de bureau, les serveurs, les smartphones et les tablettes sans oublier aussi les appareils IoT (Internet of Things) tels que les caméras, les serrures de porte, les sonnettes, les réfrigérateurs, les systèmes audio/visuels,... etc.

1.6.1 Classification des réseaux informatiques

On peut classer les réseaux informatiques selon différents facteurs :

- **La portée géographique** : un réseau informatique peut ne pas dépasser quelques mètres, autrement dit dans une même pièce, réparti entre différents appareils compatibles, il peut aussi s'étendre sur tout un bâtiment, y compris les dispositifs intermédiaires pour connecter tous les étages, sur toute une ville, ou il peut couvrir comme un seul réseau le monde entier.
- **Inter connectivité** : Les composants du réseau peuvent être connectés selon différentes manières (logique, ou physique).
Chaque appareil peut être connecté à tous les autres appareils sur le réseau, comme il peut être connecté uniquement à ses pairs gauches et droits, ou tous les appareils connectés ensemble avec un seul appareil.
- **Administration** : le réseau peut être privé, dans ce cas-là il appartient à un seul système autonome et inaccessible en dehors de son domaine physique ou logique, comme il peut être public, accessible à tous.
- **Architecture de réseau** : Les réseaux informatiques peuvent être distingués en différents types : tels que (client-serveur, peer-to-peer ou hybride), en fonction de l'architecture du réseau.

On peut avoir un système qui opère en tant que serveur, et un autre en tant que client, dans ce type d'architecture le client demande au serveur de répondre à ses requêtes (ses besoins), Le serveur prend et traite la demande au nom du client (nous allons détailler cette architecture par la suite dans ce cours).

Un autre type d'architecture appelée peer-to-peer, deux systèmes (machines) peuvent être connectés point-à-point, ces deux systèmes résident au même niveau, c'est-à-dire que la machine peut être cliente et serveur en même temps.

On peut avoir un réseau hybride qui implique les deux types d'architecture déjà cité.

1.6.2 Les différentes applications du réseau informatique

Le réseau informatique assure plusieurs fonctions :

- Partage de ressources telles que les imprimantes et les périphériques de stockage
- Échange d'informations par e-mails ou FTP (File Transport Protocol), et partage d'informations via le Web ou Internet.
- Interaction avec d'autres utilisateurs à l'aide de pages Web dynamiques
- Téléphones IP
- Vidéoconférences
- Calcul parallèle et Messagerie instantanée.

1.6.3 Les différents types de réseau informatique

Les réseaux informatiques bien qu'ils sont similaires dans leurs objectifs généraux mais n'empêche qu'ils se distinguent en fonction de l'étendue géographique. Divers types de réseaux informatiques sont classés dans les grandes catégories ci-dessous :

- **Un réseau personnel (Personal Area Network) PAN**

PAN est le plus petit des réseaux informatiques, il a une portée de connectivité allant jusqu'à 10 mètres, ce réseau peut inclure les claviers et les souris d'ordinateur sans fil, des écouteurs compatibles Bluetooth, des imprimantes sans filetc.

- **Réseau local (Local Area Network) LAN**

Un LAN est un ensemble d'appareils connectés dans une seule zone limitée, tel qu'une maison, un bureau, une école, un collège ou une université, et il est exploité sous un système administratif unique, donc il est contrôlé de manière centralisé. La mise en réseau et le routage des réseaux LAN est peu coûteuse.

- **Réseau métropolitain (Metropolitan Area Network) MAN**

Le réseau métropolitain (MAN) s'étend généralement dans toute une ville, par exemple un réseau MAN peut aider une organisation à connecter tous ses bureaux dans une ville.

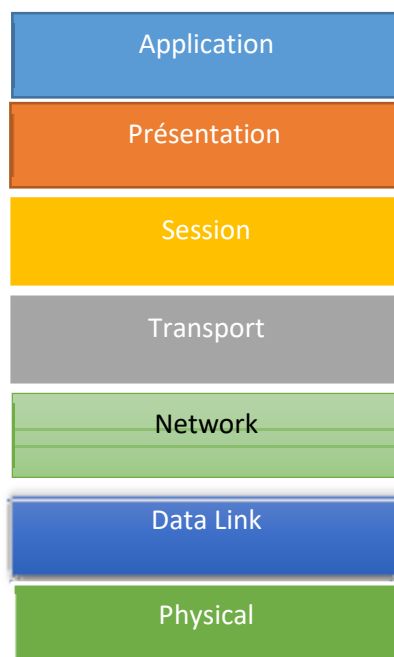
- **Réseau à grande distance (Wide Area Network) WAN**

Le réseau étendu (WAN) couvre une large zone qui peut s'étendre à travers des provinces et même un pays entier. En plus, les WAN utilisent des équipements réseau très coûteux, et ils peuvent être gérés par plusieurs administrations. Par exemple les réseaux de télécommunication sont des réseaux étendus.

Dans ce contexte, plusieurs technologies sont utilisées, on peut citer : les câbles en cuivre (FTP, UTP, STP, ...Etc.), la fibre optique (Multimode ou monomode), les satellites, les technologies sans fil (3G, 4G, 5G, et les ondes hertziennes).

1.6.4 Le modèle OSI

Le modèle « Open Systems Interconnection » (OSI) est un outil de référence pour comprendre les communications des données entre deux systèmes dans un réseau. Il divise (théoriquement) les processus de communication en sept couches. Chaque couche remplit à la fois des fonctions spécifiques pour prendre en charge les couches supérieures et offre des services aux couches inférieures.



Les sept couches du modèle OSI

- **La couche physique** : définit les spécifications des connecteurs et de l'interface, ainsi que les exigences du support (câble). Des spécifications électriques, mécaniques, fonctionnelles et procédurales sont fournies pour l'envoi d'un flux binaire sur un réseau informatique.
- **La couche DATA LINK ou liaisons de données** : Cette couche offre plusieurs services, on cite comme exemple :

- L'accès d'un appareil au réseau pour envoyer et recevoir des messages.
 - Elle utilise les adresses physiques (MAC) afin que les données d'un appareil spécifique puissent être envoyées sur le réseau, et afin d'avoir un seul identifiant pour chaque appareil réseau, l'adresse MAC est unique pour chaque carte réseau (adressage physique).
 - Elle fournit des services de détection d'erreurs, la gestion de files d'attente des paquets de données, le contrôle de la Qualité de service (QoS), les LAN virtuels (VLAN).
- **La couche network ou réseau** : elle fournit un système d'adressage logique de bout en bout pour qu'un paquet de données puisse être acheminé sur plusieurs réseaux, l'adresse IP doit être unique dans un même réseau ou sous réseau.
 - **La couche transport** : la couche transport offre une communication de bout en bout entre les terminaux (machines) via un réseau. Selon l'application, la couche de transport offre une communication fiable, orientée connexion ou sans connexion. Les protocoles de la couche transport les plus courants sont le protocole de contrôle de transmission (TCP) orienté connexion (session) et le protocole de datagramme utilisateur (UDP) sans connexion (pas de session).
 - **La couche session** : la couche session permet aux applications fonctionnant sur des périphériques d'établir, de gérer et de terminer un dialogue via un réseau. Elle fournit aussi divers services, notamment le suivi du nombre d'octets que chaque extrémité de la session a accusé sa réception de l'autre extrémité de la session.
 - **La couche présentation** : est responsable de la façon dont l'application va présenter les données, qui vont être par la suite envoyées sur le réseau. La couche présentation permet essentiellement à une application de lire (ou de comprendre) le message.
 - **La couche application** : fournit une interface pour l'utilisateur final qui utilise un appareil connecté à un réseau. Cette couche est ce que l'utilisateur voit, en termes de chargement d'une application (comme un navigateur Web ou un e-mail...etc.).

1.6.5 Adresse IP (internet Protocol) et numéro de Port

Une adresse IP est une adresse unique qui identifie un appareil sur Internet ou un réseau local. IP signifie "Internet Protocol", qui est l'ensemble des règles régissant (dirigeant) le format des données envoyées via Internet ou un réseau local, exemple d'adresse IP : (192.168.1.1), (10.0.0.2).

Le numéro de port permet à un ordinateur d'identifier facilement le trafic entrant et de l'envoyer aux processus appropriés. Différents numéros de port sont utilisés pour déterminer vers quel protocole le trafic entrant doit être dirigé prenons comme exemple le port 80 utilisé pour la communication entre le client et le serveur d'un site web via HTTP et le port 443 pour HTTPS.

1.7 Topologies et fonctionnement d'un réseau

Une topologie réseau est l'organisation avec laquelle les systèmes informatiques ou les périphériques réseau sont connectés les uns aux autres. Les topologies peuvent définir à la fois l'aspect physique et logique du réseau.

1.7.1 Topologie physique

La topologie physique d'un réseau définit la façon (l'aspect physique) dont les machines, et les appareils sont connectés au réseau. Ces topologies sont représentées et définies de la façon suivante :

1.7.1.1 Topologie en bus

Dans ce type de topologie, un seul câble relie les différents équipements qui se trouvent dans un bâtiment ou campus, ce câble réseau est appelé bus ou backbone. Son fonctionnement est simple chaque appareil peut être aisément connecté ou déconnecté du backbone.

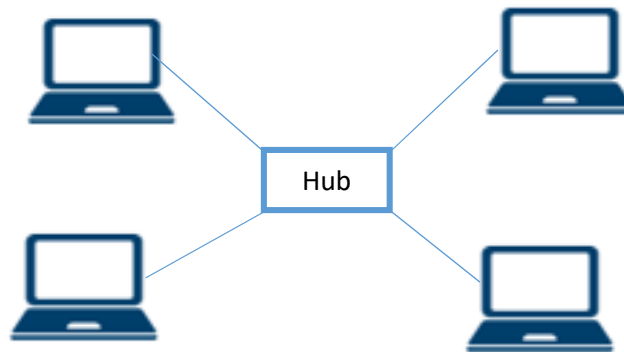


Topologie en bus

- **Avantages**
 - Fiable pour les petits réseaux.
 - Facile à utiliser et à comprendre.
- **Inconvénients**
 - Le trafic réseau est long, (n'importe quel ordinateur du réseau peut transmettre des données à tout moment).
 - Dans ce type de topologie, si une machine est détériorée cela engendre l'arrêt de tout le réseau.

1.7.1.2 Topologie en étoile

Dans ce type de topologie, chaque machine ou appareils du réseau se connecte à un concentrateur central, appelé hub. Ce hub agit comme un canal pour transmettre les messages.



Topologie en étoile

- **Avantages**

- Si l'une des machines tombe en panne, cela n'affecte pas le bon fonctionnement du réseau.
- Le centre d'un réseau en étoile est un bon endroit pour diagnostiquer les pannes du réseau
- Le remplacement ou le retrait des machines peut être facilement effectué.

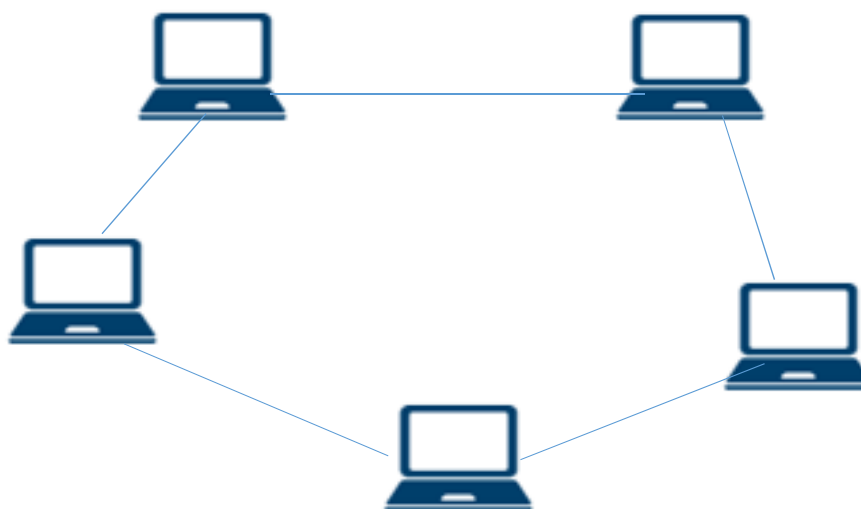
- **Inconvénients**

- Ce type de topologie est coûteux en termes de câble
- Le réseau dépend du hub, ou du nœud central, s'il ne fonctionne pas, alors l'ensemble du réseau ne parvient pas à fonctionner.

1.7.1.3 Topologie en anneau

Dans une topologie en anneau, chaque machine est connectée à celles qui se trouvent à ses côtés.

Les machines sont reliées à un répartiteur appelé MAU (multi station Access unit), qui gère la communication entre les différentes machines, en répartissant à chacune d'entre elles un temps de parole.



Topologie en anneau

Un jeton parcourt l'anneau en s'arrêtant à chaque machine. Si une machine souhaite transmettre des données, elle ajoute ces derniers au jeton, en joignant l'adresse de destination. Le jeton poursuit son chemin jusqu'à la machine de destination, cette dernière récupère les données transmises par le jeton.

- **Avantages**

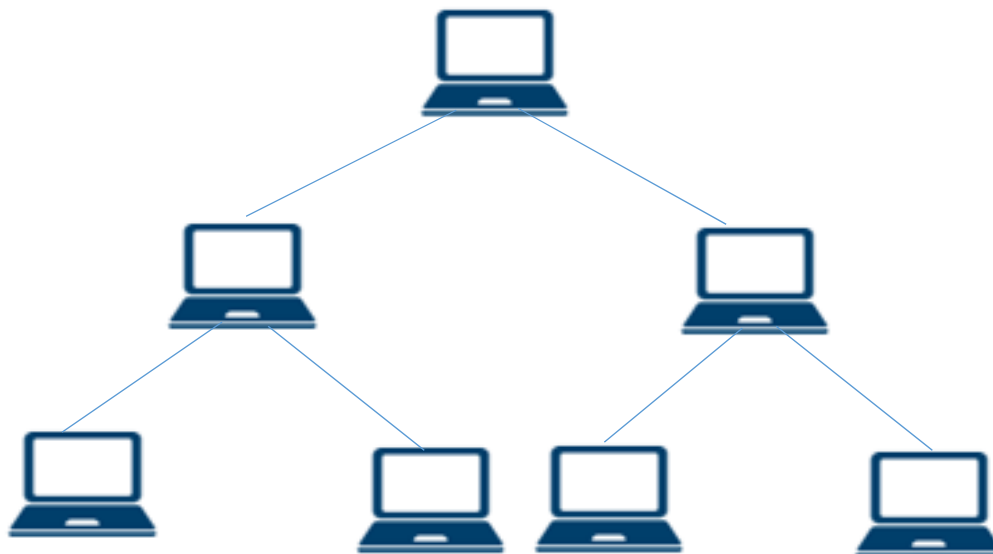
- Le réseau en anneau offre des performances élevées pour un petit nombre de postes de travail ou pour des réseaux plus importants.
- Les réseaux en anneau peuvent également couvrir de plus longues distances par rapport aux autres types de réseaux.
- Dans la topologie en anneau les jetons sont des paquets de données qui sont régénérés à chaque nœud.

- **Inconvénients**

- Ce type de topologie est coûteux en termes d'argent, de temps et d'efforts pour son installation.
- Si une machine tombe en panne, cela peut affecter l'ensemble du réseau et il est difficile de trouver un défaut dans le réseau.
- L'ensemble du système est perturbé lorsque des machines sont ajoutées ou remplacées dans le réseau.

1.7.1.4 Topologie en arbre ou hiérarchique

Comme son nom l'indique le réseau est divisé en niveau, nous avons le sommet du réseau qui représente le niveau supérieur, ce dernier est connecté à plusieurs nœuds de niveau inférieur, ces nœuds peuvent aussi être connectés à des nœuds de niveau inférieur.



Topologie en arbre

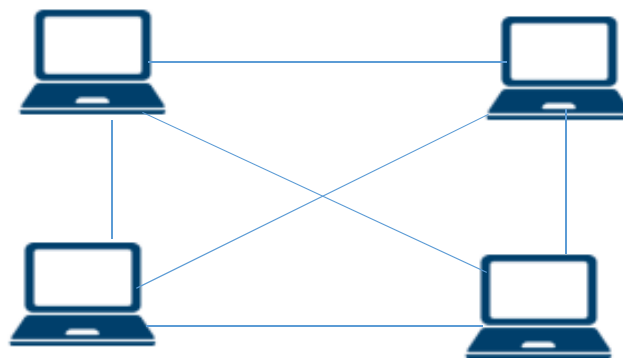
- **Avantages**

- Cette topologie présente un degré élevé de fiabilité.

- Topologie évolutive, on peut ajouter ou supprimer des machines et des connexions sans touchés aux fonctionnalités du réseau.
- Elle permet d'assurer l'intégration de plusieurs réseaux en un seul réseau.
- **Inconvénients**
 - Le maillon faible de cette topologie est constitué par les nœuds parents, l'échec d'un nœud parent, se répercute sur l'accès au sous-réseau qui lui appartient, et ce dernier reste donc isolé dans le réseau globale.

1.7.1.5 Topologie maillée

Dans une topologie maillée, chaque appareil est capable de se connecter à tous les appareils qui se trouvent à l'intérieur du réseau.



Topologie maillée

- **Avantages**
 - Ce type de topologie offre la plus grande redondance dans le cas où l'un des nœuds échoue lorsque le trafic réseau peut être redirigé vers un autre nœud.
 - C'est un bon système en matière de confidentialité car chaque ordinateur peut interagir avec tous les autres machines.
 - Les problèmes de réseau sont plus faciles à diagnostiquer.
- **Inconvénients**
 - La quantité de câbles requis est très élevée et coûteuse.
 - Un grand nombre de ports d'entrée/sortie sont nécessaires.

1.7.1.6 Topologie hybrid

Une topologie hybride est un mélange de deux ou de plusieurs topologies. Utilisée généralement par les grandes entreprises pour répondre à leurs besoins d'utilisation de réseau. Les capacités et les vulnérabilités dépendent fortement du type de topologies couplées.

- **Avantages**
 - Les topologies hybrides offrent la flexibilité, la fiabilité et l'évolutivité en combinant les aspects les plus forts de différentes topologies dans une seule configuration hybride.
 - Les topologies hybrides peuvent être modifiées selon les besoins d'utilisation.

- **Inconvénients**

- Les topologies hybrides sont très complexes.
- Les administrateurs doivent gérer chaque topologie impliquée dans une configuration hybride selon ses exigences uniques.

1.7.2 Topologie logique

La topologie logique assure la transmission des données sur une topologie physique. Elle ne concerne pas la disposition des machines, plutôt elle concerne les détails complexes du réseau, qui affectent sur le taux et la vitesse de livraison des paquets de données.

En plus simple, la topologie logique assure le fonctionnement des connexions des données sur le réseau informatique et elle peut révéler comment les appareils (machines) communiquent entre eux.

Les topologies logiques les plus courantes sont Ethernet, Token Ring et FDDI.

1.7.2.1 Ethernet

Ethernet ou technologie LAN est un type de protocole de communication qui relie les appareils (machines, serveurs, etc. ...) au sein de ce qu'on appelle un réseau local LAN. Ethernet permet évidemment l'échange des données (paquet de données) entre ces appareils une fois connectés (connexion Ethernet).

1.7.2.2 Token Ring

Token ring est une topologie de réseau local (LAN) qui envoie des données en utilisant un jeton, ce dernier permet à n'importe quelle machine dans le réseau (anneau) d'envoyer des données lorsque le jeton arrive à son emplacement, donc ce jeton assure le contrôle de la ligne de transmission.

1.7.2.3 Fiber distributed data interface FDDI

Interface de données distribuées par fibre (FDDI), est une norme de communication de données optiques utilisée pour les réseaux de longue distance.

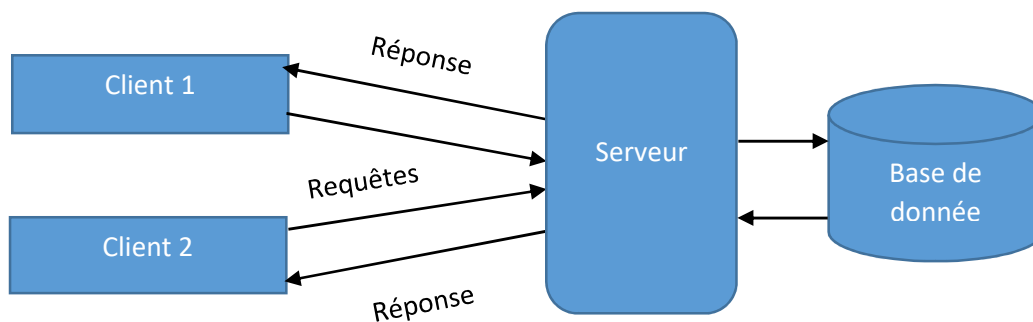
FDDI comporte deux anneaux de communication primaire et secondaire ; L'anneau primaire fonctionne le long du réseau, et l'anneau secondaire reste inactif et disponible pour la sauvegarde.

Le FDDI a été étendu par la suite au FDDI-2 pour la communication vocale et multimédia à longue distance.

1.8 Fonctionnement client serveur

Le modèle client-serveur est une architecture informatique dont Le traitement des données est divisé en deux parties distinctes. Une partie qui représente le client ou le demandeur de service et une autre partie qui représente le serveur ou le fournisseur de service. Avec des méthodes de communication standard (telles que TCP/IP et Z39.50) pour faciliter le partage d'informations entre le client et le serveur.

- **Serveur** : Un ordinateur, de grande puissance, qui contient des informations pouvant être manipulées par des clients dans le réseau. Un serveur doit être doté aussi des services dédiés aux taches attendues : service de serveur de fichiers, service FTP, service de serveur WEB, service de messagerie, ...Etc.
- **Client** : Un demandeur d'information (service) dans un réseau, habituellement une machine, qui peut interroger une base de données et/ou d'autres informations provenant d'un serveur.
- **Service** : toute ressource qu'on peut utiliser dans un réseau ou dans une machine, par exemple : donnée, fichier, contrôle, objet, temps du processeur, mémoire, dispositif d'affichage, etc...).



Modèle client serveur

Le client émet une requête vers le serveur grâce à son adresse IP (internet Protocol) et le numéro de port, qui désigne un service particulier du serveur.

Le serveur reçoit la requête et répond à l'aide de l'adresse IP de la machine et son numéro de port.

1.8.1 Les caractéristiques du modèle client-serveur

- Le client et le serveur se distinguent l'un de l'autre par les différences dans les tâches qu'ils exécutent ; Le client et le serveur fonctionnent généralement sur des plateformes informatiques différentes.
- Le client ou le serveur peut être mis à niveau sans affecter l'autre, mais en tenant compte de la compatibilité entre les deux composants.
- Les clients peuvent se connecter à un ou plusieurs serveurs et les serveurs peuvent se connecter à plusieurs clients simultanément.
- Les clients initient toujours le dialogue en demandant un service.

1.8.2 Exemple d'utilisation

- Transfert de fichiers : Il s'agit de la transmission de fichiers entre le client et le serveur, on y trouve la gestion des accès simultanés, les droits d'accès (privilège), le quota d'utilisation (quantité de données ou espace de stockage dont le client peut exploiter dans un serveur), ... Etc.
- Transfert de courrier (email) : il s'agit du transfert de messages tels que l'e-mail à l'aide du protocole de transfert de courrier Simple Mail Transfer Protocol (SMTP).
- Le transfert de fichiers multimédias tels que des images, du texte entre le client et le serveur, grâce au Protocole de transfert hypertexte (HTTP). HTTP est utilisé pour améliorer la communication entre le client et le serveur.

1.8.3 Avantages du modèle client-serveur

- Il répartit le traitement de la demande sur plusieurs machines.
- Il permet un partage plus facile des ressources du client aux serveurs.
- Il réduit la réplication (redondance) des données en stockant les données sur chaque serveur.

1.8.4 Inconvénient du modèle client-serveur

- Le nombre de personnes : Dans la plupart des réseaux client-serveur, peu de serveurs sont toujours impliqués, ce qui fait que la configuration ressemble à une perte de temps. Un réseau client-serveur est assez difficile à mettre en place, il nécessite donc beaucoup de serveurs pour ne pas rendre l'application inutile.
- La mise en place d'un réseau client-serveur est si complexe qu'elle nécessite des techniciens qualifiés et des ingénieurs de maintenance pour le gérer.
- Les serveurs sont assez chers : les serveurs sont conçus pour répondre à des normes élevées et pour avoir de meilleures performances.
- Sécurité : Le message échangé entre le client et le serveur entraîne de nombreux problèmes de sécurité. Le système d'exploitation client est facilement accessible par les serveurs, ce qui expose le système client à un certain nombre de problèmes, tel que : les dommages physiques, les menaces et les attaques de virus, etc....

1.9 Le réseau internet et le web

Le réseau Internet est le plus grand réseau informatique au monde, il permet de connecter une série de réseaux les uns aux autres, par divers moyens, notamment le câble à fibre optique, le satellite...etc.

1.9.1 La World Wide Web

Le terme World-Wide Web (WWW), généralement abrégé en « Web », est un système de documents hypertextes interconnectés accessibles via Internet, donc on peut dire qu'Internet est le système qui facilite le transfert de données et le Web est les données elles-mêmes.

1.9.2 Les protocoles utilisés sur internet

Un protocole informatique définit les règles et les procédures permettant à deux processus informatiques (agent, ordinateur, composants, ...Etc.) d'échanger des données.

Les protocoles déterminent, le type de vérification des erreurs, la manière dont les appareils sont interconnectés, comment savoir si le processus est en train d'envoyer ou de recevoir des données, quelle est le type de compression de données utilisée, ...Etc.

Parmi les protocoles les plus utilisés sur internet : le protocole de transfert hypertexte (HTTP), et le protocole de transfert de fichiers (FTP), aussi une autre version sécurisée de Http qui est le "HTTPS" (HyperText Transfer Protocol Secure), c'est la combinaison du HTTP avec une couche de chiffrement comme SSL ou TLS.

1.9.3 Les moteurs de recherches et navigateur internet

Les moteurs de recherche sont des applications web qui permettent d'effectuer des recherches en lignes, comme Google, Bing et Yahoo, à ne pas confondre avec les navigateurs qui sont des logiciels informatiques comme Mozilla Firefox, Safari, Opera conçu pour l'accès à des documents Web appelés « pages Web », lier les uns aux autres à l'aide de « liens hypertexte ».

1.9.4 Page web et lien hypertexte

Une page web est un ensemble de données ou d'informations conçues pour être consultées via un navigateur internet, c'est un simple fichier texte écrit dans un langage de description appelé « HTML ». Les pages web peuvent contenir des graphiques, des sons, des textes et des vidéos.

Un lien hypertexte est un élément d'une page Web qui crée un lien vers un autre endroit dans le même document, ou vers un document entièrement différent.

1.9.5 Le langage HTML

HTML ou Hypertext Markup Language est le langage informatique de base du Web appelé aussi langage de balisage ou de marquage ou de structuration, il facilite l'organisation dynamique de l'information à travers des liens et des connexions (appelés hyperliens).

HTML a été développé au début des années 1990 par Tim Berners-Lee conçu pour fournir un moyen par lequel les informations scientifiques pourraient être partagées entre divers chercheurs. On peut citer différentes version d'HTML : HTML 1.0 La version de base qui prend en charge les éléments de base tels que les contrôles de texte et les images. Ensuite d'autres versions plus développées comme HTML 2.0, HTML 3.0, html 4.0 et HTML 5.0.

1.9.5.1 Structure d'un document HTML

```
<html>
  <head>
    <title>
      première année SM 2021
    </title>
  </head>
  <body>
    .....
  </body>
</html>
```

1.9.5.2 Les principales balises du Langage HTML :

- <title> titre </title> : balises ouvrantes et fermantes de titre
- <h1> titre 1 </h1> : balises ouvrantes et fermantes de titre de niveau 1
- hyperlien : balise permettant d'introduire un lien hypertexte
- : balise permettant d'afficher une image
- Texte : mettre un texte en gras
- Texte ou <i> Texte</i> : mettre un texte en italique

A chaque balise de début d'une action <...>, correspond une balise de fin de l'action </...>.

II. Introduction à l'algorithmique

Cette partie du programme concerne tout simplement les notions de base de l'algorithmique. L'objectif est de faire comprendre aux étudiants la définition d'un algorithme, en quoi consiste, et comment peut-on élaborer une solution à un problème donné sous forme de règles et d'instructions (algorithme).

II.1 Notion d'algorithmique

En informatique, les algorithmes fournissent aux ordinateurs un guide ou un moyen pour accomplir des actions. Ce guide n'est donc qu'une suite d'instructions qui décrivent d'une manière précise et claire comment accomplir et exécuter une tâche donnée.

II.1.1 Définition 1

Un algorithme est un ensemble d'instructions ou de règles mathématiques et logiques, conçu spécialement pour assurer une solution à un problème bien défini.

II.1.2 Définition 2

Un algorithme est un ensemble ordonné d'instructions utilisées par un être humain ou une machine basé sur des solutions mathématiques appliquées pour transformer les données brutes (problèmes) en données traitées ou résolus.

II.1.3 Les caractéristiques d'un algorithme

L'algorithme doit se distinguer par les caractéristiques suivantes :

- Non ambigu : L'algorithme doit être clair et sans ambiguïté. Ses étapes doivent être bien définies et ne doivent conduire qu'à un seul sens.
- Entrée : Un algorithme doit avoir 0 ou plusieurs entrées bien définies.
- Sortie : Un algorithme doit avoir 1 ou plusieurs sorties bien définies.
- Fini : un algorithme doit se terminer après un nombre fini d'étapes.
- Concis : un algorithme doit être bref (le plus court possible), sinon, il faut décomposer le problème en plusieurs sous problèmes.
- Structuré : un algorithme doit comporter différentes parties facilement identifiables.

II.1.4 Un programme

Un programme est un ensemble d'instructions bien structurées écrit dans un langage de programmation et exécuté par un ordinateur (processeur).

II.1.5 Langage de programmation

Le langage de programmation tel que : PASCAL, C, C++, C#, JAVA,.....Etc., est l'intermédiaire entre la machine et l'être humain, il permet de présenter l'algorithme sous forme d'instructions proche au langage machine et compréhensible par l'être humain.

II.1.6 Un compilateur

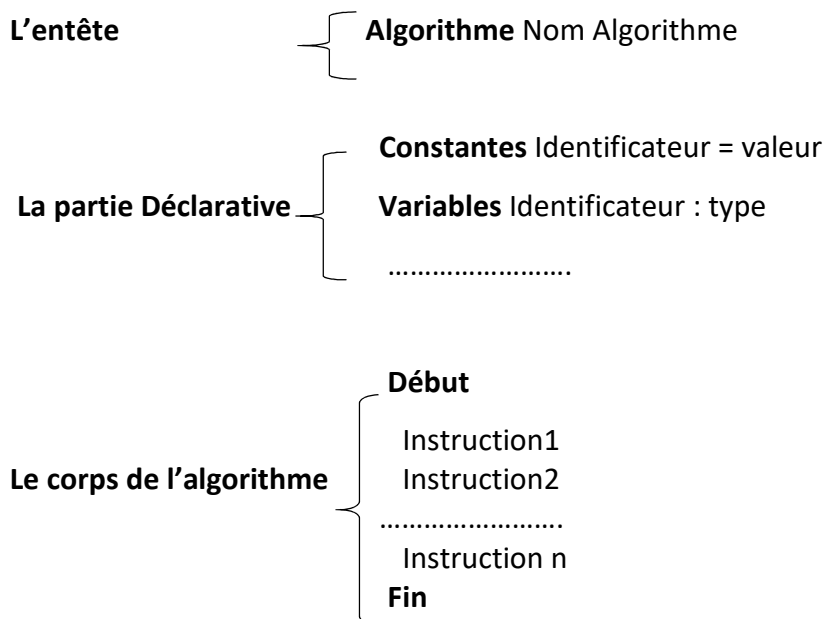
Le compilateur a comme rôle la traduction d'un programme, écrit dans un langage de programmation (lisible par un être humain), en un programme exécutable par une machine (binaire).

II.1.7 Les étapes de mise en point d'un programme

La mise au point d'un programme informatique s'effectue en se basant sur les étapes suivantes :

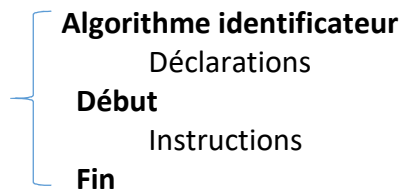
- **L'Analyse** : l'étape primordiale qui comporte la compréhension du problème, et l'élaboration de la solution sous forme d'un algorithme.
- **La programmation** : cette étape concerne la transformation de l'algorithme (la solution) en un ensemble d'instructions écrites en langage de programmation.
- **La compilation et l'édition des liens** : la compilation consiste à transformer le programme sous sa forme lisible en programme compilé et prêt à être exécuté par la machine ; L'édition des liens consiste à lier le programme avec tous les éléments externes (souvent des bibliothèques auxquelles il fait référence).

II.1.8 Les éléments de base d'un algorithme



II.1.9 Syntaxe d'un algorithme

La syntaxe d'un algorithme est présentée de la façon suivante :



- **Identificateur** : permet de nommer l'algorithme
- **Déclarations** : permet de déclarer les variables ou les constantes utilisées par l'algorithme
- **Instructions** : permet d'indiquer l'ensemble des actions élémentaires nécessaires pour résoudre le problème.

II.1.10 Les variables et les constantes

II.1.10.1 Les variables

Sont des valeurs qui peuvent changées ou variées dans le temps, ou pour chaque utilisation.

- **Syntaxe**

Variable identificateur : **type**

L'identificateur de la variable est le nom qu'on affecte à une variable donnée

Exemples :

Variable surface : réel

Variable a : entier

Variable Nom : chaîne

- **Quelques types de variables**

- **Type entier** : les nombres entiers par exemple : 3, 25, 250
- **Type réel** : les nombres réels par exemple : 3.14 ou 17.5
- **Type booléen** : Ce type ne peut prendre que deux valeurs possibles : VRAI ou FAUX
- **Type caractère** : Toute lettre, ponctuation, code des opérations, espace,... par exemples : 'x' '+' '.'
- **Type chaîne de caractère** : Ce type est une suite de caractères par exemple : « informatique », « bonjour »

II.1.10.2 Les constantes

Les constantes peuvent être définies comme des valeurs fixes, donc elles ne changent pas dans le temps.

- **Syntaxe**

Constante identificateur = valeur

Exemple : Constante PI = 3.14

II.1.10.3 Les opérateurs :

Un opérateur est un outil qui permet d’agir sur une variable ou d’effectuer des calculs. Il existe plusieurs types d’opérateurs :

- **Les opérateurs arithmétiques**

Les opérateurs arithmétiques permettent d’effectuer des opérations arithmétiques entre des données numériques, on distingue :

+	addition
-	soustraction
*	multiplication
/	division
mod	modulo
^	Puissance (en Pascal il n y a pas de puissance)
div	Division entière

- **Les opérateurs logiques et les opérateurs relationnels**

Les opérateurs logiques combinent des données booléennes pour former des expressions logiques plus complexes :

Opérateur unaire	La négation (non) : not
Opérateurs binaires	La conjonction (et) : and
	La disjonction (ou) : or

- **Les opérateurs relationnels :**

<	Inférieur
>	Supérieur
>=	Supérieur ou égal
<=	Inférieur ou égal
=	Egal
≠, <>	Différent

- **L’opérateur d’affectation**

L’opérateur d’affectation, est souvent représenté par le symbole «← » ou (:=), qui attribue une valeur à une variable ou à une constante.

Exemple 1 : $A \leftarrow B$ {on assigne à A la valeur de B}

Exemple 2 : $X \leftarrow X+1$ {on assigne à X la valeur de « X+1 », une incrémentation par 1}

II.2 Les instructions d'entrées/sorties

L'instruction **d'entrée ou de lecture** permet par exemple à l'utilisateur de saisir une donnée au clavier, pour qu'elle soit utilisée par le programme. Cette instruction est notée : lire « identificateur », (read ou readln) en langage pascal.

Exemple :

Lire (A)

Lire (A,B,C)

Les instructions **d'entrées** permet aussi de recevoir différents types d'informations : lecteur optique (code à barres), lecteur QR, lecteur de cartes magnétiques, ... Etc. afin de les traiter par la suite.

L'instruction **de sortie** permet d'afficher des informations à l'écran après traitement, cette instruction est notée : écrire « identificateur », (write ou writeln) en langage pascal.

Exemple : Ecrire (donner la valeur de A)

II.3 Exemples

Algorithme qui calcule la somme de deux nombres réels x,y

Algorithme somme deux nombres
Déclarations
x,y : réel
début
écrire ("donner deux nombres réels")
lire(x,y)
écrire("la somme de',x,'et',y,'est',x+y)
fin

II.4 Les structures de contrôle

On distingue trois structures de contrôle principales pour la conception des algorithmes :

II.4.1 Bloc d'instruction

Algorithme	En langage pascal
Début	Begin
instruction1	instruction1;
instruction2	instruction2 ;
.....
Fin	end.

II.4.2 Les alternatives

- **Alternative simple (si..... alors)**

Algorithme Si <condition> alors Instruction1 Instruction2 Sinon Instruction3 Fin si	En langage pascal If (condition) then Instruction1; Instruction2; else Instruction3;
--	---

- **Alternative multiple**

Algorithme Selon que cas1 :Instruction1 cas2 :Instruction2 cas n : Instruction n fin selon que	En langage pascal Case..... of cas1 :Instruction1 ; cas2 :Instruction2 ; cas n :Instruction n ; end ;
---	--

II.4.3 les boucles

Il existe trois types de boucles pour assurer la notion de répétition d'un processus dans un algorithme ;

- **La boucle (Pour ... faire) ou (for...do)**

Algorithme Pour indice=1 jusqu'à n faire Instruction1 Instruction2 Fin pour	En langage pascal for indice =1 to n do Instruction1; Instruction2; end.
--	---

- **La boucle (Tant que ... faire) ou (while ...do)**

Algorithme Tant que <condition> faire Instruction1 Instruction2	En langage pascal While <condition> do Instruction1; Instruction2; End.
---	--

- **La boucle (Répété ... jusqu'à) ou (repeat ...until)**

Algorithme	En langage pascal
Répété	Repeat
Instruction1	Instruction1 ;
Instruction1	Instruction2 ;
Jusqu'à <condition>	until <condition>
Fin répété	End.

II.5 Les fonctions et procédures

Les fonctions et procédures sont de petites sections de code (bloc d'instructions) utilisées pour effectuer une tâche particulière, leur rôle principal est d'éviter la répétition de commandes dans le même programme et le rendre plus flexible, elles peuvent être appelées selon le besoin, plusieurs fois dans le même programme, et avec les mêmes paramètres (variables) ou avec des paramètres différents selon les cas.

II.5.1 Les fonctions

Une fonction est un bloc d'instructions qui retourne obligatoirement une et une seule valeur (résultat) à l'algorithme appelant.

II.5.1.1 Syntaxe d'une fonction

```
Fonction  Identificateur_Fonction  (identificateur_variable  :  Type_variable;.....)  :  
type_Fonction  
  Déclaration  
    Identificateur_variable : Type_variable  
  Début  
    Instructions  
  Retourner    Résultat  
Fin Fonction
```

Remarque : il est nécessaire de préciser le type de la fonction qui est en réalité le type de la variable qu'elle doit retourner.

II.5.1.2 Appelle de fonction

L'appelle d'une fonction est effectué par son exécution. Elle peut être sous la forme suivante :

< Identificateur _Fonction > (variable-effectifs)

II.5.2 Les procédures

Une procédure est une séquence d'instructions groupées dans un bloc pour le but d'effectuer une ou plusieurs tâches. Cette tâche est exécutée par un simple appel de la procédure dans un algorithme ou dans un autre sous-algorithme. Une procédure ne retourne pas forcément une valeur, pas comme le cas des fonctions.

II.5.2.1 Syntaxe d'une procédure

```
Procédure Identificateur_Procedure (identificateur_variable : Type_variable;.....)  
  Déclaration  
    Identificateur_variable : Type_variable  
  Début  
    Instructions  
  Fin Procédure
```

Remarque : l'appelle des procédures se fait de la même manière que les fonctions, en indiquant le nom suivi des variables entre parenthèses.

II.6 Les structures de données

Les structures de données sont utilisées pour organiser et stocker les données dans la mémoire, de manière à pouvoir les manipuler et les traiter (analyser, comparer, appliquer des opérations mathématique, ...Etc.) ultérieurement. Il existe plusieurs types :

Les structures de données séquentielles (tableaux) ;

Les structures de données linéaires (listes chaînées) ;

Les arbres ;

Les graphes ;

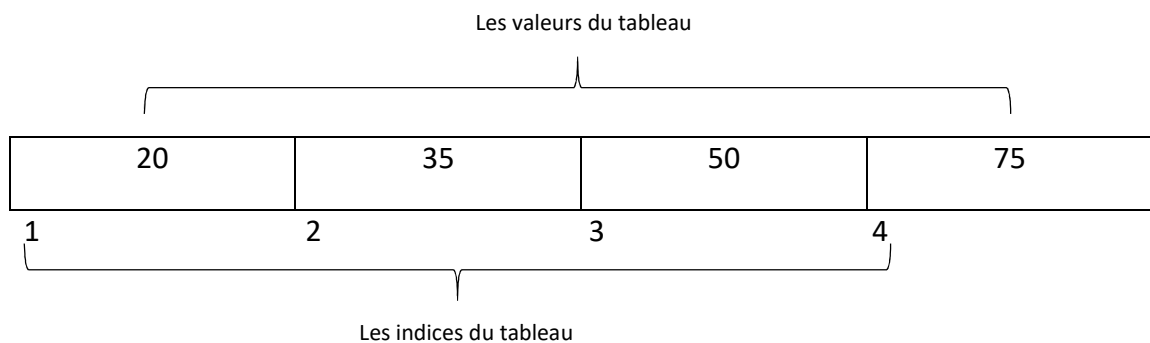
Dans notre cours, on s'intéresse aux structures de données séquentielles.

II.6.1 Les tableaux à une dimension (vecteur)

Un vecteur est un ensemble ordonné de n valeurs du même type, ayant comme caractéristiques les points suivants :

- Le vecteur doit avoir un nom, par exemple, le vecteur V.
- Le vecteur doit être déclaré dans la partie déclaration du programme et son type doit être précisé.
- Tous les éléments du vecteur sont du même type.
- Le nom de chaque élément est composé du nom du vecteur suivi d'un indice (index).

Exemple :



- **Syntaxe**

La déclaration de tous les éléments du tableau est réalisée avec un mot-clé «identificateur du tableau» en indiquant entre parenthèses le minimum et le maximum d'éléments et leur type de donnée

Exemple :

En algorithmique V : Tableau [1..50] de réel.

En langage Pascal V : array [1..50] of real;

II.6.2 Les tableaux à deux dimensions (matrice)

Une matrice est un ensemble de données du même type référencé par deux indices (les lignes et les colonnes).

- La matrice doit avoir un nom, par exemple, la matrice M.
- La matrice doit être déclarée dans la partie déclaration du programme et son type doit être précisé.
- Tous les éléments de La matrice sont du même type.
- Le nom de chaque élément est composé du nom de La matrice suivi de deux indices (deux index).

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 6 & 7 & 4 \end{pmatrix} \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 3 \\ \hline \end{array} \text{ Indice j}$$

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline \end{array} \text{ Indice i}$$

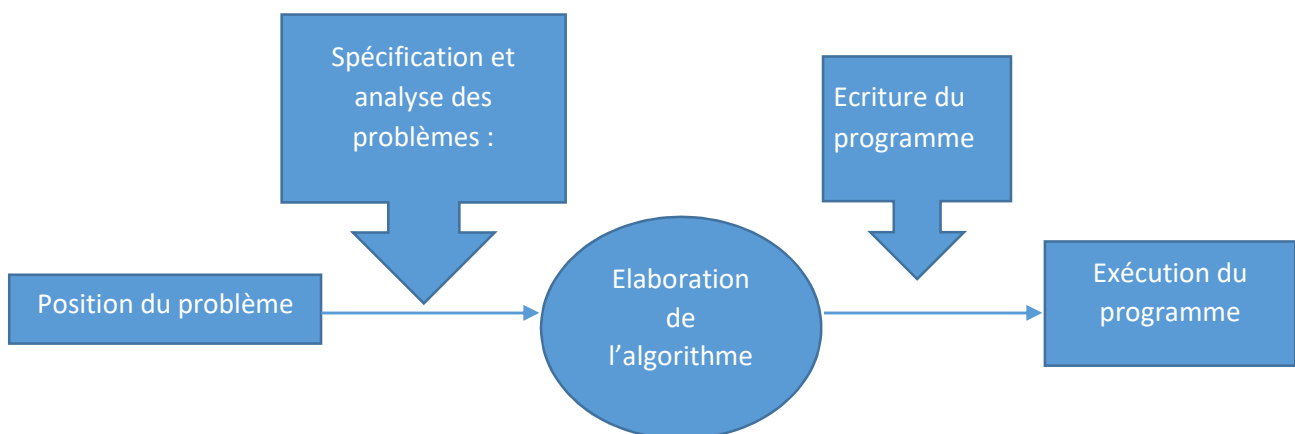
Exemple

En algorithmique M : Tableau [1..50,1..50] de réel.

En langage Pascal M : array [1..50,1..50] of real;

II.7 Processus de résolution d'un problème

Le processus de résolution d'un problème peut être résumé en 5 étapes dans le schéma suivant :



Processus de résolution d'un problème

II.7.1 Les différentes étapes du processus de résolution d'un problème

II.7.1.1 Position du problème :

Cette étape concerne la reformulation du problème pour pouvoir le résoudre correctement.

II.7.1.2 Spécification et analyse des problèmes :

Cette étape est indispensable pour déterminer les formules de calculs, les règles de gestion et tout ce qui peut aider pour proposer la bonne solution.

L'analyse des problèmes se concentre sur :

- Les données nécessaires aux traitements (les entrées)
- Le traitement (les instructions nécessaires pour atteindre le résultat),
- Le résultat souhaité (les sorties),

II.7.1.3 Elaboration de l'Algorithme :

Cette étape concerne la représentation de la solution proposée par des instructions rédigées par ordre chronologique.

II.7.1.4 Ecriture du programme :

Cette étape concerne la traduction de l'algorithme proposé par un langage de programmation choisi tel que : pascal, c, c#, java.....

II.7.1.5 Exécution du programme :

La cinquième et dernière étape permet de tester si le programme réalisé assure le résultat de la solution déjà proposée dans la partie **Spécification et analyse des problèmes**.

II.7.2 Elaboration d'un algorithme complet

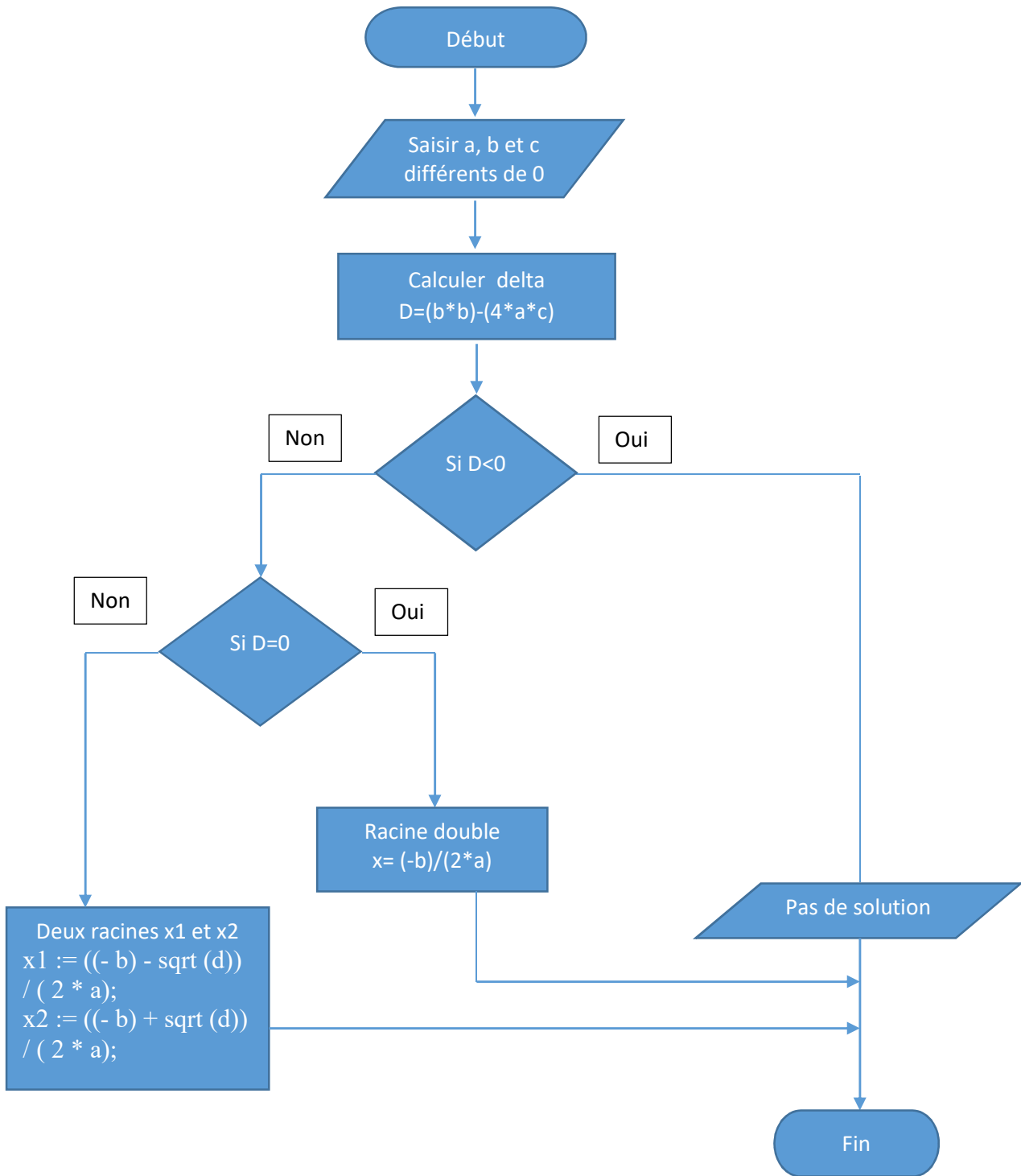
II.7.2.1 Résolution d'une équation du second degré

Proposer un algorithme et un programme en pascal pour résoudre une équation de type :

$$ax^2 + bx + c = 0$$

Solution :

On peut présenter la solution sous forme d'un organigramme :



Organigramme pour la solution d'une équation du second degré

Program equation;**var**

a,b,c,x,x1,x2,d : real;

begin**writeln** (ce programme concerne la résolution d'une équation de 2nd degré ');**writeln** ('de forme E: ax^2+bx+c ');**writeln** ('Donner les valeurs a, b et c dans l'ordre');**writeln**('donner la coefficient a');**readln**(a);**writeln** ('donner la coefficient b');**readln**(b);**writeln** ('donner la coefficient c');**readln** (c);**writeln**('calculer delta Δ ');

d:= (b * b) - (4 * a * c) ;

if d < 0 **then****writeln** ('Pas de solution');**if** d = 0 **then****begin**

x := (- b) / (2 * a);

writeln ('E admet une racine double ',x:6:2);**end;****if** d > 0 **then****begin**

x1 := ((- b) - sqrt (d)) / (2 * a);

x2 := ((- b) + sqrt (d)) / (2 * a);

writeln ('E admet deux racines');**writeln** (x1:6:2) ;**writeln** (x2:6:2);**end;****end.**

II.8 Exercices avec correction

Exercice 1

Ecrire un programme qui demande un entier (n) en entrée, et affiche la somme des entiers compris entre 1 et n.

Solution

```
Program suite ;
Var i ,n,s :integer ;
Begin
Writeln('donner la valeur de n');
Read(n) ;
S:=0;
For i :=1 to n do
s:=s+i;
writeln(' la valeur de la suite de n est ',s);
end.
```

Exercice 2

Ecrire un programme simulant une calculatrice à 4 opérations en utilisant CASE pour le choix de l'opération à effectuer.

Solution

```
Program calculatrice ;
var
  val1, val2, resultat : real;
  operation : char;
begin
  writeln('Première valeur ?');
  readln(val1);
  writeln('Opération (+ - * /) ? ');
  readln(operation)
  writeln('Deuxième valeur ? ');
  readln(val2);
  case operation of
    '+' : resultat := val1 + val2;
    '-' : resultat := val1 - val2;
    '*' : resultat := val1 * val2;
    '/' : resultat :=v al1 / val2
  end;
  writeln('Résultat : ',resultat)
end.
```

Exercice 3

Ecrire un programme en pascal qui calcule le minimum de trois entiers saisis.

Solution

```
Program min ;
var
a,b,c:integer;
begin
readln(a,b,c);
if(a<b)and(a<c)then
writeln('Le plus petit est:',a)
else
if(b<a)and(b<c)then
writeln('Le plus petit est: ',b)
else
writeln('Le plus petit est:',c);
end.
```

Exercice 4

Ecrire un programme qui lit un caractère en entrée, et affiche "minuscule" si le caractère est une lettre minuscule, "majuscule" si c'est une majuscule, et "autre" si ce n'est aucun des deux.

Solution

```
Program caractere;
var
c:char;
begin
readln(c);
if ord(c) in [65..90] then writeln('majuscule') else
if ord(c) in [97..122] then
writeln('minuscule')
else
writeln('autre');
end.
```

Exercice 5

Ecrire un programme en pascal qui permet de calculer le plus grand diviseur commun entre deux nombre x et y.

Solution

```
Program diviseur;
Var
x,y:integer;
begin
writeln('donner la valeur de x');
read(x);
```

```
writeln('donner la valeur de y') ;
read(y) ;
while x<>y do
if x>y then
x:=x-y
else
y:=y-x;
writeln('le plus grand diviseur commun est',x);
end.
```

Exercice 06

Ecrire un programme en Pascal qui permet de calculer le nombre d'occurrence d'un élément (X) dans un tableau.

Solution

```
Program occurrence;
var
T : array [1..100] of integer ;
n,i,f,x : integer ;
Begin
writeln(' donner le nombre des elements à remplir n');
read (n) ;
writeln ('entrer les valeur du Tableau T :') ;
For i:= 1 to n do
begin
read (T[i]) ;
end;
writeln ('entrer la valeur de l''élément à chercher x :') ;
read(x) ;
f := 0 ;
for i := 1 to n do
begin
if T[i] = x then
f := f+1 ;
end ;
writeln ('l''élément apparait : ', f,'fois dans le tableau T ');
end.
```

Exercice 07

Ecrire un programme en Pascal qui permet de lire une valeur entière (n), puis calcule et écrit le n-ième terme de la suite :

$$U_{n+1} = 2U_n + 3, \text{ sachant que } U_0 = 1.$$

Solution

```
Program suite ;
var
i,n:integer;
u0:real;
begin
writeln('donner le nombre d'iteration n');
read(n) ;
u0:=1;
for i:=1 to n do
begin
u0:=2*u0+3;
end ;
writeln('la suite u',n,'=',u0) ;
end.
```

Exercice 8

Ecrire un programme qui permet de tester si un nombre entier (n) est premier.

Solution

```
Program nombrepremier;
var
i,compt,n:integer;
begin
writeln('donner n');
read(n);
compt:=0;
for i:=1 to n do
begin
if (n mod i )=0 then
compt:=compt+1;
end;
if compt=2 then
writeln('le nombre est premier')
else writeln('le nombre n'est pas premier');
end.
```

Exercice 9

Ecrire un programme en pascal pour calculer la somme et la moyenne générale d'un étudiant qui possède dix notes d'examen, nous vous proposons de résoudre ce problème en utilisant un tableau.

Solution

```
Program tableau;
var
t=array[1..20] of integer ;
i: integer ;
somme:real;
```

```

begin
writeln (veuillez introduire les 10 notes' ) ;
for i:=1 to 10 do
begin
readln (T[i]);
end;
somme:=0;
for i:=1 to 10 do
begin
somme:=somme+T[i];
end;
writeln('la somme est',somme) ;
writeln('la moyenne est',somme/n) ;
end.

```

Exercice 10

Ecrire un programme en pascal, qui permet de calculer le factoriel (!) d'un nombre entier (N).

Solution

```

Program factoriel;
Var i ,N,fact :integer ;
Begin
Writeln('Donner la valeur de N');
Read(N) ;
fact:=1;
For i :=1 to N do
fact:=fact*i;
writeln(' fact N != ',fact);
end.

```

Exercice 11

Ecrire un programme permettant de résoudre un système de 2 équations à 2 inconnues :

$$\left\{ \begin{array}{l} u_1x+v_1y=w_1 \\ u_2x+v_2y=w_2 \end{array} \right.$$

Solution

```

Program equation;
var
u1,u2,v1,v2,w1,w2:real;
x,y,d:real;
begin
writeln('donner les coefficients u1, u2 , v1 , v2 , w1 , w2');
readln(u1,u2,v1,v2,w1,w2);
//calcul du determinant
d:=u1*v2-u2*v1;
//resolution du systeme

```

```

if (d <> 0) then
begin
x:=-( v1*w2-v2*w1)/d;
y:=(u1*w2-u2*w1)/d;
writeln( 'x = ', x, ' y=',y);
end
else
writeln('determinant nul');
readln;
readln;
end.

```

Exercice 12

Ecrire un programme qui permet de remplir une matrice A, puis affiche le maximum, et le minimum de cette matrice

Solution

```

program minmax;
var
A:array[1..50,1..50] of integer;
i,j,n,m,max,min:integer;
begin
writeln('donner le nombre des ligne n et le nombre de colonne j') ;
readln(n,m);
for i:=1 to n do
for j:=1 to m do
begin
writeln('A[',i,',',j,']=');
readln (A[i,j]);
end;
max:= A[ 1,1 ] ;
for i:=2 to n do
for j:=2 to m do
begin
if A[i,j] > max then
max:= A[i,j];
end;
min:= A[ 1,1 ] ;
for i:=2 to n do
for j:=2 to m do
begin
if A[i,j] < min then
min:= A[i,j];
end;
writeln('le maximum est=',max);
writeln('le minimum est=',min);
end.

```

Exercice 13 :

Ecrire un programme en pascal utilisant les fonctions, pour résoudre la loi de probabilité suivante :

$$c_n^p = \frac{n!}{(n-p)! * p!}$$

Solution

```
program fact;
var
n,p,c,t,r :integer ;
f:real;
function facto (a:integer;m:integer): integer;
var
i:integer;
begin
m:=1;
for i:=1 to a do
m:=m*i;
facto:=m;
end;
begin
writeln('donner n et p') ;
readln(n,p);
f:=facto(n,c)/(facto(n-p,t)*facto(p,r)) ;
writeln('la formule est',f) ;
end.
```

Exercice 14

Sachant que la formule pour calculer le poids (p) d'un objet est la suivante :

$$p=m*g$$

Où **m** est la masse de l'objet exprimée en kg, et (g) est l'intensité de la pesanteur (تسارع الجاذبية).

Sur la terre : g = 9,81 N/kg, et Sur la lune : g = 1,6 N/kg.

Ecrire un programme en pascal qui permet de calculer le poids (p) de n'importe quel objet de masse (m), sur la terre et sur la lune.

Solution

```
Program poids;
var p, m, g : real ; x: string;
begin
writeln ('donner la valeur de la masse m en kg') ;
readln (m) ;
writeln ('choisissez terre ou lune') ;
```

```
readln (x);  
Select x of  
'terre' : p:=m*9.81 ;  
'lune' : p:=m*1.6 ;  
End ;  
writeln ('le poids sur la ',x, ',p)  
End.
```


Références

Michael L.Schmit .Chapter 2 - What Is Assembly? Quantasm Corporation, Cupertino, California, Available online 27 June 2014.

Paul Simoneau .The OSI Model: Understanding the Seven Layers of Computer Networks. Copyright ©2006 Global Knowledge Training LLC. All rights reserved.

Haroon Shakirat Oluwatosin (2014) .Client-Server Model.e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. IX (Feb. 2014), PP 67-71

<https://www.dictionary.com/browse/computer>

<https://archimede.mat.ulaval.ca/amq/bulletins/dec06/boulier.pdf>

<http://www.gwleibniz.com/calculator/calculator.html>

<https://www.britannica.com/biography/Charles-Babbage>

https://ftms.edu.my/v2/wpcontent/uploads/2019/02/DHMBussComp_Chapter-1.pdf