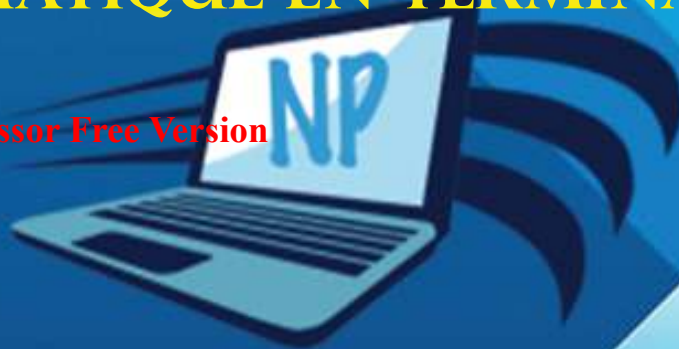


INFORMATIQUE EN TERMINALE C-D-E

PDF Compressor Free Version



Cours et exercices corrigés
selon l'APC
classe de
terminales C-D-E



Par un groupe d'enseignants

Groupe whatsapp : nouveau programme info

Administrateur principal: NJINKEU Jean Jules

EDITION 2020

NOUVEAU PROGRAMME Tles CDE

Une œuvre du groupe WHATSAPP : NOUVEAU PROGRAMME INFO

CHEF D'ATELIER DES TRAVAUX TERMINALE CDE

NJOOUNANG WANGKO STEVE CHRISTIAN (PLEG)

N°	LEÇONS	ENSEIGNANTS AYANT CONÇU ET REDIGE LA LECON :
1.	Types de systèmes informatiques, Types de logiciels et rôle des pilotes	TUAYO ZOYEM LIONEL
2.	Caractéristiques de quelques périphériques et composants internes de l'unité centrale	WALDLE BIRBI epse GNAWE NGUIMEYA T BAUDOIN
3.	Types de maintenance et causes du dysfonctionnement du matériel et du logiciel	NGANSSO DUPLEX
4.	Rôle des équipements de protection et risques	FOKOU NOUMBISSI STEPHANE NANWOU ALAIN GISCARD
5.	Tableurs et Mise en forme d'une feuille de calcul	NGANTCHEU PIERRE FLAUBERT
6.	Fonctions et Références dans les feuilles de calcul	TAOWE OULE ROMAIN
7.	Tableau croisé dynamique, zone impression, protection feuille de calcul	ONANA BELINGA RODRIGUE
8.	Produire Facture, Bulletin de Paie ou de note	TANEKAM CELESTIN
9.	Types de réseaux et avantages des réseaux sans fils	ZADANA SIDONIE SOBGUE CORINE
10.	Topologies et architectures réseaux	NGANTCHEU PIERRE FLAUBERT, ATATCHIM BRICE
11.	Équipements et protocoles réseaux	BAYE RUTH Epse ABADOMA NJOFANG HERMAN RAOUL
12.	Décrire l'adresse IP version4	NJOFANG HERMAN RAOUL

PDF Compressor Free Version

13.	Modes d'adressages et commandes réseaux	WOTCHUENG TEUSSIDO HERMANN
14.	Méthodes de conception d'un S.I	DJOUMESSI GERMAINE Epse AWOUEDEU
15.	Élaboration d'un MCD	NJINKEU JEAN JULES , ISSA SADJO
16.	Règles de Passage du MCD au MLD et	ONANA BELINGA RODRIGUE
17.	Introduction au Langage SQL	NJOUNANG WANGKO STEVE CHRISTIAN
18.	Requêtes SQL de définition et de manipulation des données.	WAMBA SEZAWO ELIHUC
19.	Création d'une BD en mode graphique dans Access	EBODE ONAMEWALI SIMEON YANNICK
20.	Structures de contrôles Algorithmiques	TUAYO ZOYEM LIONEL
21.	Structures de données et déclaration des Tableaux	SOPTSI VOLTAIRE
22.	Parcours et recherche séquentielle dans un tableau	NJOUNANG WANGKO STEVE CHRISTIAN
23.	Exécution d'un algorithme de recherche dans un tableau	PECHANGOU JEREMIE SERGE
24.	Notions de bases de la programmation en C	GEPNANG FRANCK YANNICK , PECHANGOU JEREMIE SERGE
25.	Instructions élémentaires (bases) en C	MEMBOU TAGHEU GHISLAIN BRICE
26.	Structures de Contrôle en C	ONANA BELINGA RODRIGUE , LOYEM TIEGHOUE ROSTAND
27.	Utiliser un Tableau et traduire un algorithme en C	NJINKEU JEAN JULES
28.	Identifier/interpréter/corriger erreurs de programmation en C	NJINKEU JEAN JULES

PROGRAMME (Terminales CDE)

PDF Compressor Free Version

MODULE 1: SYSTEMES INFORMATIQUES

UA 1: DESCRIPTION DE L'ARCHITECTURE D'UN SYSTÈME INFORMATIQUE

Lecon1 : Types de systèmes informatiques, Types de logiciels et rôle des pilotes

Lecon2 : Caractéristiques de quelques périphériques et composants internes de l'unité centrale

UA 2: PROTECTION D'UN ENVIRONNEMENT DE TRAVAIL

Lecon3 : Maintenance et causes du dysfonctionnement du matériel et du logiciel

Lecon4 : Rôle des équipements de protection et risques

UA 3: PRODUCTION D'UN CLASSEUR

Lecon5 : Tableurs et Mise en forme d'une feuille de calcul

Lecon6 : Fonctions et Références dans les feuilles de calcul

Lecon7 : Tableaux croisés dynamiques , zone d'impression, protection feuille

Lecon8 : Produire une Facture , bulletin de Paie ou de notes

UA 4: CONFIGURATION D'UN RESEAU INFORMATIQUE

Lecon9 : Types de réseaux et avantages des réseaux sans fils

Lecon10 : Topologies et architectures réseaux

Lecon11 : Équipements et protocoles réseaux

Lecon12 : Décrire l'adresse IPV4

Lecon13 : Modes d'adressages et commandes réseaux

MODULE 2: SYSTEMES D'INFORMATION ET BASES DE DONNEES

UA 5: CONCEPTION DES MODELES DE DONNEES D'UN S.I

Lecon14 : Méthodes de conception d'un S.I

Lecon15 : Élaboration d'un MCD

Lecon16 : Passages du MCD au MLD

UA 6: IMPLEMENTATION D'UNE BASES DE DONNES

Lecon17 : Introduction au langage SQL

Lecon18 : Requêtes SQL de définitions et de manipulation de Données

Lecon19 : Création d'une Bases de Données en mode graphique dans Ms Access

MODULE 3: SYSTEMES D'INFORMATION ET BASES DE DONNEES

UA7: ECRITURE ET EXECUTION DES ALGO INTEGRANT STRUCTURES DE DONNEES

Lecon20 : Structures de contrôles algorithmiques

Lecon21 : Structures de données et déclaration des tableaux

Lecon22 : Parcours et recherche séquentielle dans un tableau

Lecon23 : Exécution d'un algorithme de recherche dans un tableau

UA 8: PROGRAMMATION EN LANGAGE C

Lecon24 : Introduction/Notions de bases au langage C

Lecon25 : Instructions de élémentaires

Lecon26 : Structures de contrôle en C

Lecon27 : Les tableaux en langage C et traduire un algorithme en C

Lecon28 : Identifier/interpréter/corriger les erreurs de programmation en C

MODULE 1 :SYSTEMES INFORMATIQUES

PDF Compressor Free Version

UA1 :Description de l'architecture d'un système Informatique

UE1 : Les types de systèmes informatique, types de logiciels, rôle des pilotes

Compétence visées :

- *Décrire les trois types de systèmes informatiques (personnel, d'organisation, de contrôle et de commandes);*
- *Décrire les principaux types de logiciels(rôle et 04 fonctionnalités);*
- *Enoncer le rôle d'un pilote;*
- *Décrire les principaux composants internes de l'unité Centrale (carte mère, processeur, RAM, DD, bloc d'alimentation, cartes d'extension) ;*

Situation problème :

Monsieur TOUKAPA est le directeur de l'entreprise ZOWAEFS. Après avoir informatisé sa structure, il a recruté une secrétaire particulière pour tout travail de bureautique surtout avec ordinateur. Sa secrétaire s'étant absenté un matin pour cause de maladie, Monsieur Toukapa souhaite modifier le contenu d'un rapport, il a appelé sa secrétaire qui lui a demandé de démarrer son ordinateur, puis dans le système d'exploitation, lancer le logiciel Word, puis connecter sa clé USB sur un port USB se trouvant sur la carte mère de l'unité centrale afin d'ouvrir ce document. Ce dernier ne se retrouve pas dans tout cela et vous pose les questions suivantes :

Qu'est ce qu'un système d'exploitation ? un logiciel ? Word ?

C'est quoi l'unité centrale ? La carte mère ?

Lequel de toutes ces choses va lui permettre de modifier le rapport ?

I. Définitions :

Système informatique : Ensemble de moyens d'acquisition et de restitution, de traitements et de stockage de données dédiée au traitement de l'information ;

Système d'exploitation : Ensemble de programmes qui contrôle et coordonne l'utilisation des ressources de l'ordinateur;

Logiciel : Ensemble des programmes et des procédures nécessaires au fonctionnement d'un système informatique ;

Pilote : Programme informatique permettant de faire fonctionner un périphérique ;

Périphérique : Appareil que l'on peut connecter ou intégrer à un ordinateur pour compléter ses fonctions ;

RAM :Random Access Memory;

CPU :Central Processing Unit;

BIOS :Basic Input Output System;

II- les Types de Systèmes informatiques



Il existe 03 types systèmes informatique : **les systèmes informatiques personnels, les systèmes informatiques d'organisation et les systèmes informatiques de contrôle et de commandes**

II-1- Les systèmes informatiques personnels

Ils constituent la partie la plus visible et mieux connue des systèmes informatiques car ils ont pour but d'offrir des services utilitaire ou ludique à leur propriétaire. Ils peuvent comporter différents matériels connectés de manière permanente ou non parmi lesquelles :

- **Des ordinateurs personnels encore appelés ordinateur de bureau ou PC (Personal Computer), des ordinateurs portables, des tablettes, assistant personnel**
- **Des périphériques tel qu'imprimante, scanner**
- **Des équipements de transmission de donnée tel que modem, switch, box internet**

Egalement, ces systèmes peuvent offrir de nombreux service grand public parmi lesquelles :

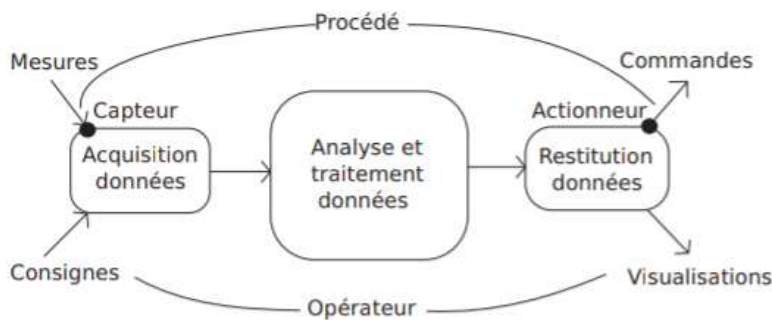
- **La navigation sur le web**
- **La bureautique (avec traitement de texte, calcul, gestion de données personnel)**
- **Les messagerie et réseaux sociaux**
- **Les jeux**
- **Le multimédia avec retouche d'images, traitement de son, montage vidéo, etc...**

II-2- Les systèmes informatiques d'organisation

Ces systèmes sont hébergés au sein d'une organisation, que ce soit des entreprises, des associations, des administrations, des laboratoires de recherche ou autres. Ils sont généralement constitués de plusieurs composants parmi dont les plus importants sont : **les ordinateurs, les serveurs, des équipements de traitement et de transmission de données, un réseau, un système d'information, des systèmes distribués et le cloud.**

II-3- Les systèmes informatiques de contrôle et de commandes

PDF Compressor Free Version



Ces systèmes sont généralement un peu plus complexe que les autres. En général, ils reçoivent des données relatives à l'état d'un procédé extérieur via des capteurs traite ces données puis en fonction du résultat agissent sur un procédé extérieur via un actionneur. C'est l'exemple des robots

III- les Types deLogiciel

Il existe deux grands types de logiciels pour un ordinateur à savoir : **Les logiciels systèmes (ou logiciel de base)** et **les logiciels d'application (ou logiciel applicatif)**.

III-1- Les logiciels systèmes

Ce sont des logiciels qui sont lancés avant les logiciels d'applications. Ils sont l'intermédiaire entre ceux-ci et le matériel de l'ordinateur.

Exemples : Windows 10, Mac Os, Unix, Ubuntu, Linux, Android, MS DOS

Il existe plusieurs types de système d'exploitation mais les principaux sont :

- **Les systèmes d'exploitation Mono-Utilisateur:**
Ce type de système permet à un seul utilisateur de travailler à la fois dans la même machine
- **Les systèmes d'exploitation Multi-Utilisateur:**
Ce type de système permet à plusieurs utilisateurs de travailler dans la même machine.
- **Les systèmes d'exploitation Mono-tache :**
Ce type de système permet à un seul programme d'être exécuté à la fois à tout instant
- **Les systèmes d'exploitation Multitâche :**
Ce type de système permet à Plusieurs programmes d'être exécutés parallèlement

Les logiciels systèmes ont plusieurs fonctions parmi lesquelles :

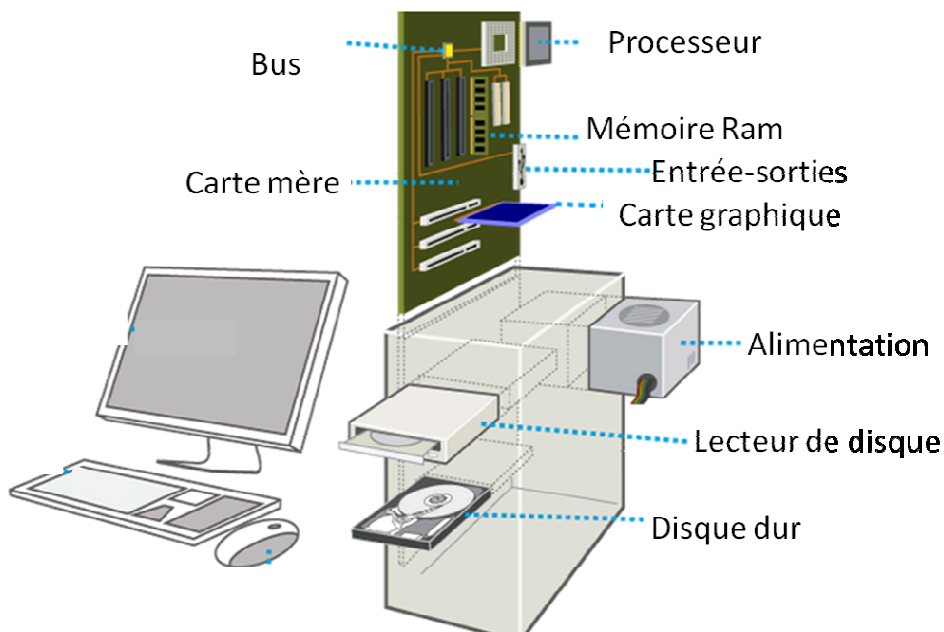
- **La gestion de l'exécution des logiciels d'applications**
- **La gestion des fichiers**
- **La gestion de la mémoire**
- **La gestion des entrées/sorties**
- **La gestion de la sécurité**
- **La gestion des processus**

III-2- Les logiciels d'applications

Ce sont des logiciels qui sont conçus pour l'utilisateur final dans le but de l'aider à réaliser de nombreuses tâches. Ils sont capables de manipuler du texte, des chiffres et des graphiques. Il existe plusieurs types de logiciels d'application à savoir :

- **Les traitements de texte** : ils permettent de saisir et de mettre en forme du texte.
Exemple : Word, Wordpad, Opensuite
- **Les navigateurs** : Ils permettent d'ouvrir des pages web.
Exemple : Firefox, Chrome, Internet explorer.
- **Les tableurs** : Ils permettent de créer des feuilles de calculs.
Exemple : Excel, Lotus, OpenOffice calc
- **Les systèmes de gestion de base de données (SGBD)** : Ils permettent de créer et contrôler des bases de données.
Exemple : Access, MySql, Postgre.
- **Les logiciels multimédias** : Ils permettent de manipuler du son, les images ou les vidéos.
Exemple : VLC, RealPlayer, Windows Media Player,
- **Les logiciels de jeu** : pour se distraire.
Exemple : Zuma, Sudoku, Jewel match.

IV- L'unité centrale et ses composants



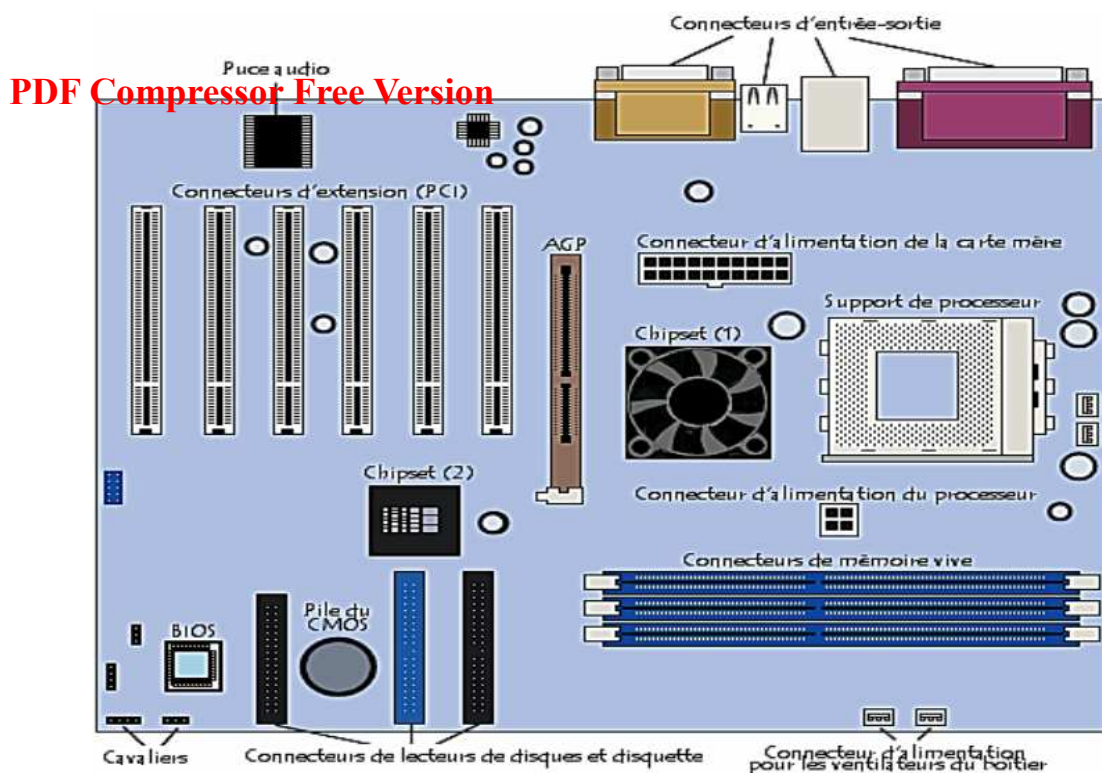
C'est la partie principale d'un ordinateur car elle regroupe le microprocesseur, la mémoire, le disque de stockage et les lecteurs.

Elle se présente sous la forme d'une caisse dans laquelle on retrouve de nombreux éléments.

Les éléments de base sont l'ensemble du matériel physique qui peuvent être vue et touché.

Parmi ces derniers on a :

- **La carte mère** : encore appelé **motherboard** ou **main board**, c'est l'élément fondamental puisque tous les autres éléments sont connectés dessus.



- **Le processeur ou CPU:** c'est l'organe responsable de tous les traitements.
- **La RAM (Random Access Memory) :** elle se présente sous forme d'association de barretteposée sur une petite plaque. C'est une mémoire vive



- **L'alimentation :** L'alimentation se charge de distribuer l'énergieélectriqueindispensable aux composants de la machine. Elleest reliée à la prise de courant par un câble d'alimentation.



- **Le disque dur :** Le disque dur contient les informations (programmes et données) utiles au fonctionnement de la machine et à sa gestion



- **Le lecteur de CD/DVD:** C'est un dispositif qui permet de lire les CD/DVD.



- **La carte vidéo :** responsable de l’affichage du moniteur



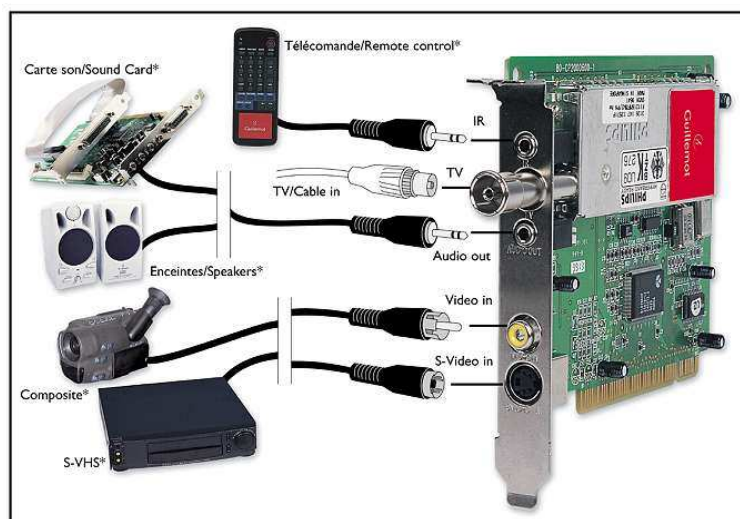
- **La carte son :** qui permet le traitement des sons par l’ordinateur



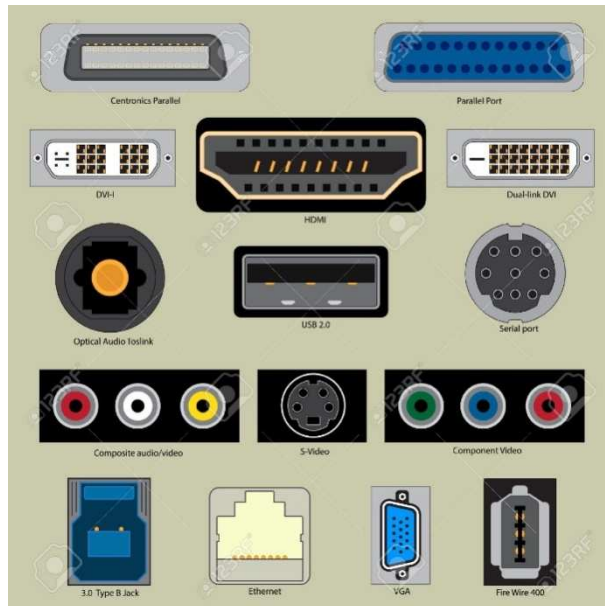
- **La carte réseau :** responsable des communications réseau



- **La carte TV :** qui permet de capter les chaînes TV



- **Les ports :** ce sont des prises au travers duquel les données sont échangées entre l'ordinateur et les périphériques. Les ports permettent de connecter les périphériques à l'unité centrale.



- **Le bios :** C'est un petit programme situé sur la carte mère de l'ordinateur dans une petite puce de type ROM. C'est le premier programme chargé en mémoire lorsque l'ordinateur est mis sous tension. Il assure plusieurs fonctions parmi lesquelles le **POST(Power On Self Test) qui est un ensemble de tests effectués sur l'ordinateur avant le démarrage et également la recherche du disque sur lequel se trouve le système d'exploitation à démarrer.**

Différence entre le bios et le système d'exploitation

Le système d'exploitation permet de dissocier les programmes et le matériel en offrant une interface homme machine à l'utilisateur. Il sert de lien entre les logiciels d'application et le matériel en exécutant des tâches à la demande de l'utilisateur.

Le BIOS permet de contrôler le matériel. Il effectue des opérations de vérification sur ce matériel lors de la mise sous tension avant de passer la main au système d'exploitation. Le BIOS est donc un programme qui est exécuté avant le système d'exploitation.

V-Les utilitaires

Un utilitaire est un petit programme réalisant les tâches d'administration du système mais n'ayant pas suffisamment de fonctions pour être considéré comme un logiciel à part entière. Les utilitaires sont des logiciels qui aident l'utilisateur dans l'exécution de certaines tâches souvent liées à la maintenance, l'optimisation, la manipulation de l'ordinateur. Nombreux d'entre eux-ci sont intégrés au système d'exploitation.

Exemple : Le clavier visuel, la loupe, les antivirus, le magnétophone, powerIso, Winrar.

JEU BILINGUE :

<u>Français</u>	<u>Anglais</u>
Logiciel	Software
Pilote	Pilot
Système d'exploitation	Operating system
interface	Interface

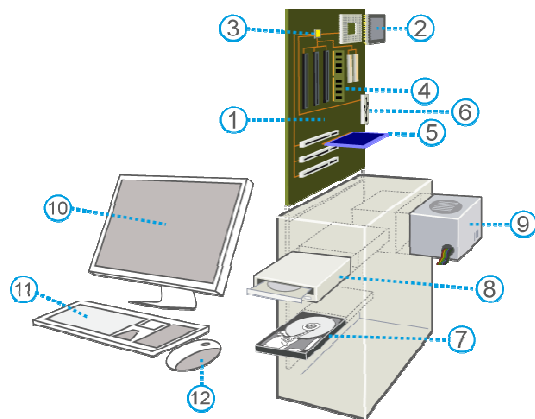
EXERCICES ET CORRIGES

Exercice 1 :

1. Définir : Système d'exploitation, logiciel, pilote
2. Citer le 03 types de systèmes informatiques.
3. Quels sont les 02 grands types de logiciel qui existent ?
4. Citer 02 exemples de chaque types

Exercice 2 :

Identifier par leur nom chacune des parties de l'ordinateur suivant :



Exercice 3 :

Classer les logiciels suivant en fonction de leur type : Word, Firefox, Excel, VLC, Football 2019

CORRIGES

Exercice 1 :

1. Définition :

Système d'exploitation : Ensemble de programmes qui contrôle et coordonne l'utilisation des ressources de l'ordinateur;

Logiciel : Ensemble des programmes et des procédures nécessaires au fonctionnement d'un système informatique ;

Pilote : Programme informatique permettant de faire fonctionner un périphérique ;

2. 03 types de systèmes informatiques :

Les systèmes informatiques personnels, les systèmes informatiques d'organisation, les systèmes informatiques de contrôles et de commandes

PDF Compressor Free Version

3. Les 02 grands types de logiciel qui existent : Les logiciels systèmes et les logiciels d'applications

4. 02 exemples de chaque type :

-02 exemples de logiciel systèmes : Windows 10, Android, Mac Os, Unix, Linux, Ubuntu

-02 exemples de logiciel d'application : Word, Excel, VLC, Firefox, Postgres

Exercice 2 :

1. Carte mère 2. Processeur 3. Bus 4. RAM 5. Carte graphique 6. Entrée-Sortie

7. Disque dur 8. Lecteur de disque 9. Alimentation 10. Moniteur ou écran

11. Clavier 12. Souris

Exercice 3 :

- **Les traitements de texte : Word**
- **Les navigateurs : Firefox**
- **Les tableurs : Excel**
- **Les logiciels multimédias : VLC**
- **Les logiciels de jeu : Football 2019**

MODULE 1: LES SYSTÈMES INFORMATIQUES

UA1: Description de l'architecture d'un système informatique

UE2: Caractéristiques de quelques périphériques et composants internes de l'unité centrale

Compétences Visées:

- ✓ décrire les caractéristiques de quelques périphériques et composants (imprimante, moniteur, processeur, disque dur, RAM
- ✓ Décrire les caractéristiques des principaux composants internes de l'unité centrale (Carte mère, processeur,, RAM, DD, bloc d'alimentation, cartes d'extension)

Contrôle de pré requis :

1. Définissez partie matérielle de l'ordinateur.
2. La partie matérielle de l'ordinateur est divisée en combien de grandes parties ?
3. Citez trois périphériques de l'ordinateur
4. Citez trois éléments qui se trouvent dans l'unité centrale

SITUATION PROBLEME :

Pour votre réussite au probatoire votre père a acheté un ordinateur muni d'une étiquette qui affiche les mentions suivantes:

Unité centrale DELL; processeur Intel ® Quad Core ; Mémoire 32Go – Stockage 500Go ; Intel HD Graphics – lecteur – graveur DVD ; 4 ports USB ; 1 sortie HDMI ; Ethernet ; Windows 8 . 1 64 bits.

Confus et curieux par cette étiquette , il te sollicite tout d'abord pour l'édifier davantage en ce qui concerne le boîtier présent dans le carton, puis sur les composants que renferme ce boîtier, ensuite de lui expliquer les données qui figure sur l'étiquette de cet ordinateur.

Consignes:

1. Que représente ce boîtier? (Réponse: l'unité centrale)
2. Quels sont les composants qu'on retrouve dans l'unité centrale? (Réponse: la carte mère, le processeur, la mémoire vive, le disque dur.,

PDF Compressor Free Version

3. Quel élément faudra t il ajouter pour sortir les données sur un support physique (Réponse: une imprimante)

4. Que représentent les données mentionnés sur l'étiquette ? (Réponse : les caractéristiques de l'ordinateur)

CARACTÉRISTIQUES DE QUELQUES PERIPHERIQUES

1- L'ÉCRAN

Encore appelé moniteur, son efficacité dépend de sa taille, sa résolution ou finesse et sa fréquence maximale

- **La taille de l'écran** : elle se mesure à l'aide de sa diagonale. L'unité de mesure de l'écran est le pouce ("). Un pouce=2,54cm
- **Le pas de masque** (en anglais dot pitch): C'est la distance qui sépare deux luminophores; plus celle-ci est petite plus l'image est précise.
- **La finesse**: la finesse ou résolution de l'écran est le nombre de points par unité de longueur et de largeur de l'écran. Elle s'exprime en Pixel
- **La fréquence maximale de balayage** : c'est le nombre de page et de lignes qu'un écran peut rafraîchir par seconde. Elle s'explique en herz.

2-L'IMPRIMANTE

L'imprimante est un périphérique permettant de faire une sortie(sur papier) des données de l'ordinateur. L'imprimante est généralement caractérisée par les éléments suivants

- **Vitesse d'impression** : exprimée en page par minute (ppm), la vitesse d'impression représente la capacité de l'imprimante à imprimer un grand nombre de pages par minute.

Pour les imprimantes couleurs, on distingue habituellement la vitesse d'impression en monochrome et en plusieurs couleurs.

- **La résolution**: exprimé en point par pouce (ppp) ou dot per inch(dpi) définit la finesse de l'impression. La résolution est parfois différente pour une impression monochrome ou photo.
- **Le format de papier**: selon leur taille, les imprimantes sont capables d'accueillir différentes tailles de documents généralement au format A4 (21*29,7cm) puis rarement au format A3 (29,7*42cm).
- **Le temps de préchauffage** : il représente le temps d'attente nécessaire avant la première impression.
- **La mémoire embarquée** : il s'agit de la quantité de mémoire permettant à l'imprimante de stocker les travaux d'impression.

Certaines imprimantes permettent également d'imprimer sur d'autres types de supports tels que le CD ou DVD.

CARACTÉRISTIQUES DES COMPOSANTS DE L'UNITÉ CENTRALE

L'unité centrale est le boîtier qui contient les principaux composants indispensables au fonctionnement de l'ordinateur. Nous pouvons citer entre autre: la carte mère (la colonne vertébrale), le processeur (le cerveau), la RAM, le disque dur, la boîte d'alimentation (le cœur et les poumons)

➤ **La carte mère:** c'est le composant principal de l'unité centrale. C'est le support sur lequel vient se brancher ou connecter tous les composants d'un ordinateur

➤ **Le processeur:** encore appelé microprocesseur ou CPU (central processing unit) est le cerveau de l'ordinateur. C'est lui qui effectue toutes les opérations permettant à l'utilisateur d'interagir avec l'ordinateur. Il organise les échanges de données entre les composants (mémoire vive, disque dur, carte graphique). Le microprocesseur est une puce électronique qui se présente sous la forme d'une boîte, dont la taille et la forme sont fonction du degré de perfectionnement et de la technologie qui a permis sa fabrication. Ces principales tâches sont : lire les données en mémoire, traiter les données et écrire les données en mémoire. caractérisé par sa fréquence, sa marque et son type:

- Sa fréquence ou vitesse de calcul s'exprime en Hertz . De nos jours les processeurs dépassent les 3 giga hertz (ghz)
- Sa marque (intel, motorola, AMD...)
- Son type (pentium, athlon, céרון...)
- La largeur de ses registres internes de manipulation des données
- La cadence de son horloge exprimée en MHz ou GHz
- Son jeu d'instruction

➤ **La mémoire vive :** La mémoire vive ou RAM (Random Access Memory ou mémoire à accès direct en français) est la mémoire principale de l'ordinateur qui stocke de façon temporaire les informations des programmes et données en cours de fonctionnement. La mémoire vive est volatile. Elle ne stocke pas les données en cas de coupure. Ses particularités sont : sa rapidité d'accès et son aspect temporaire. Ses caractéristiques sont : son type, sa capacité de stockage et sa vitesse d'exécution

- La capacité de stockage de la mémoire s'exprime en octet (ou byte). Aujourd'hui la RAM

d'un ordinateur peut atteindre plusieurs gigas octets (Go)

- Le type de RAM: Il en existe plusieurs types SDRAM, DDRAM
- La vitesse d'exécution: elle se mesure en nano seconds

➤ **Le disque dur:** C'est également la mémoire de l'ordinateur dont le rôle est de stocker les données informatiques de façon permanente. Il contient le système, les programmes installés et les données personnelles de l'utilisateur. Plusieurs disques durs existent : les disques durs classiques, les disques durs HDD (hard disk drive) , les disques durs SSD, les disques durs externes. Les principales caractéristiques du disque sont : sa capacité (est la quantité de données qu'il peut contenir. Elle s'exprime en octet ou byte. Aujourd'hui la capacité de stockage peut atteindre plusieurs Téra Octet (To)) et sa vitesse d'exécution.

- **La boîte d'alimentation:** Il transforme et fournit l'énergie nécessaire à la carte mère.

MOTS FRANCAIS	TRADUCTION EN ANGLAIS
Unité Centrale	Central unit
Disque dur	Hard disk
Processeur	Processor
Carte mere	Motherboard
Moniteur	Monitor
Imprimante	Printer

SITUATION D'INTEGRATION:

Voici un ordinateur exposé dans la vitrine d'un supermarché.

1. Que représente les éléments mentionnés sur la plaque à coté?
Reponse: les caracteristiques de l'ordinateur
2. Identifier les composants internes de l'unité centrale. **Reponse:processeur,RAM,DD,Cartes graphiques,lecteurs.**

Ordinateur de Bureau - Processeur Intel® Quad Core J1900 - Mémoire 4Go - Stockage 500Go - Intel HD Graphics - Lecteur de cartes SD - Lecteur-Graveur DVD - 1 port USB 3.0 - 4 ports USB 2.0 -VGA - 1 sortie HDMI - Ethernet - Clavier AZERTY et Souris USB - Windows 8.1 64 bits –

REINVESTISSEMENT

Choisissez une machine de votre salle informatique, identifiez tous les peripheriques de cet ordinateur en précisant les caractéristiques des composants suivants processeur, ecran, mémoire RAM, disque dur, carte graphique.

MODULE I : LES SYSTEMES INFORMATIQUE

PDF Compressor Free Version

UE2 : PROTECTION D'UN ENVIRONNEMENT DE TRAVAIL

UE3 : TYPES DE MAINTENANCE ET CAUSES DU DYSFONCTIONNEMENT DU MATERIEL ET DU LOGICIEL

❖ Compétences visées

- Définir maintenance et identifier les types de maintenance
- Lister les mesures de protection de matériel et de logiciel informatique
- Lister les causes du dysfonctionnement matériel et du logiciel

❖ Situation – problème

Le père d'Abel vient de lui acheter un ordinateur et une clé USB comme cadeau pour sa réussite à l'examen BEPC.

Abel à l'aide de sa clé USB, passe son temps au cyber à télécharger les logiciels (de jeux), les vidéos et les images qu'il l'installe et copie dans son ordinateur à la maison à tort et à travers. Seulement, depuis quelques temps, son ordinateur devient de plus en plus lent et burg. Depuis la nuit passée, son ordinateur a arrêté de fonctionner et ne s'allume même plus après une coupure et un retour instantané et brusque de l'énergie électrique.

Consigne

- a. Quelles peuvent être les causes de cette lenteur ?
 - La saturation du disque dur DD ou HDD
 - La présence d'un ou des virus informatiques dans l'ordinateur
- b. A quoi est dû le burg du clavier ?
 - Le burg peut être dû à l'infiltration de la poussière, de la sueur et autre débris de petite taille dans le clavier ?
- c. Qu'est ce qui peut être à l'origine de l'arrêt à l'origine de l'arrêt (incapable de s'allumer) de l'ordinateur d'Abel ?
 - La défaillance du bloc d'alimentation suite à une surtension ou une surintensité du courant électrique.

PDF Compressor Free Version

INTRODUCTION

Un système informatique (ensemble formé du matériel et du logiciel informatique) tout comme notre organisme a besoin d'entretien qui peut être préventif ou curatif.

I. LES CATEGORIES DE MAINTENANCE

La maintenance informatique est l'ensemble des tâches effectuées sur un équipement informatique dans le but de le maintenir en bon état.

On distingue généralement deux catégories de maintenance informatique à avoir :

1. La maintenance matérielle ou du Hardware

La maintenance matérielle est l'ensemble des tâches effectuées sur la partie matérielle d'un ordinateur, dans le but de la maintenir en bon état. Elle concerne généralement la souris, le clavier, l'écran, l'unité centrale...

2. La maintenance logicielle ou du Software

C'est l'ensemble des tâches effectuées sur la partie logicielle d'un ordinateur dans le but de permettre son bon fonctionnement. Elle concerne les logiciels, les données....

II. LES TYPES DE MAINTENANCES

On distingue trois types de maintenances à savoir :

- ✓ La maintenance préventive
- ✓ La maintenance curative
- ✓ La maintenance évolutive

1- La maintenance préventive

C'est l'ensemble des tâches effectuées sur un système informatique afin de le prévenir de tout dommage. Exemple :

- Le nettoyage de l'unité centrale et des périphériques
- L'installation d'un antivirus informatique

2- La maintenance curative ou corrective

C'est l'ensemble des tâches effectuées sur un système informatique dans le but de résoudre un problème de dysfonctionnement. Exemple :

- Le changement d'un périphérique défectueux
- Le changement d'une pièce défectueuse
- La désinstallation d'un logiciel incompatible avec les autres
- La libération du disque dur

3- La maintenance évolutive

C'est l'ensemble des tâches effectuées sur un système informatique pour améliorer son fonctionnement. Exemple :

- Le remplacement d'une Ram de capacité moins grande par une Ram de capacité plus grande ;
- Le partitionnement du disque dur ;
- L'ajout d'une carte d'extension
- La mise à jour d'un antivirus ou d'un logiciel

III. LES MESURES DE PROTECTION DU MATERIEL INFORMATIQUE

Protéger le matériel informatique c'est le prévenir de tout risque de dysfonctionnement.

Cette protection se fait par :

- L'achat d'un matériel de bonne qualité ;
- L'utilisation des régulateurs de tensions ou des onduleurs ;
- Le nettoyage des périphériques à l'aide d'une éponge sèche et propre ;
- L'utilisation des CD de nettoyage pour nettoyer les lecteurs DVD, CD ;*
- Couvrir l'ordinateur après le travail à l'aide d'un tissu propre afin d'éviter les dépôts de poussière ;
- Mettre son ordinateur à l'abri de l'humidité

IV. LES MESURES DE PROTECTION DU LOGICIEL INFORMATIQUE

La protection du logiciel informatique passe par :

- La conservation des supports de stockage amovibles (clé USB, CD ROM et DVD ROM, carte mémoire)...

Pour une meilleure conservation, il faut :

Mettre les CD et DVD dans un étui,
PDF Compressor Free Version
 . Evité de mettre les supports de stockage dans les endroits humides et éviter de retirer les clés USB sans éjecter.

- Installer un antivirus puis veiller à sa mise à jour constante
- Toujours scanner un support de stockage amovible lorsqu'on utilise à une machine autre
- Installer un pare feu ou firewall (en anglais)
- N'activer le partage des données que si l'on veut faire une connexion réseau

V. LES CAUSES DE DYSFONCTIONNEMENT DU MATERIEL D'UN ORDINATEUR

Le dysfonctionnement matériel d'un ordinateur peut provenir de :

- Le mauvais branchement des périphériques de base
- Le mauvais entretien
- La défaillance du bloc d'alimentation dû à la surtension ou aux foudres
- La saturation du disque dur
- La défaillance du processeur
- La défaillance de la pile de masse

VI. LES CAUSES DU DYSFONCTIONNEMENT DU LOGICIEL

La principale cause de dysfonctionnement d'un logiciel est laprésence des virus informatiques. Mais on peut aussi noter :

- o L'absence du pilote
- o La maladresse de l'utilisateur (origine des pertes de données)

JEUX BILINGUE

ANGLAIS	FRANÇAIS
Hardware	Partie matérielle
Software	Partie logicielle
Driver	Pilote
firewall	Pare feu

Exercices

PDF Compressor Free Version

1. Quelles différences faites-vous entre un régulateur de tension et un onduleur ?
2. Donnez la différence entre un antivirus et un pare feu
3. Définir virus informatique et donner la différence entre un cheval de Troie et un spiware
4. Citer quelques antivirus

Résolution

- 1- La différence entre un régulateur de tension et un ordinateur est que :
 - Un régulateur de tension a pour rôle de réguler la tension électrique du courant à la norme d'un appareil
 - Un onduleur en plus de réguler la tension du courant électrique à la norme d'un appareil, amagazineaussil'énergie électrique pouvant faire fonctionner cet appareil quelques minutes après interruption brusque du courant électrique.

- 2- La différence entre un antivirus et un pare feu est que :

Un antivirus est un ensemble de programmes qui permet de lutter contre les virus informatiques

Un pare feu est un ensemble de programmes qui lutte contre les spiwares

- 3- Un virus informatique est un programme malveillant qui nuit au bon fonctionnement de l'ordinateur

La différence entre le cheval de Troie et le spiware est que :

- Le cheval de Troie est un virus classique alors que le
- Spiware est un virus espion qui se repend à travers le réseau internet.

- 4- Quelques antivirus

- Norton antivirus
- Avast
- AVG Internet Security
- Kaspersky
- Avira
- F-Secure

MODULE I :SYSTEMES INFORMATIQUES
PDF Compressor Free Version

UA 2: PROTECTION D'UN ENVIRONNEMENT DE TRAVAIL

**UE 4: DÉCRIRE LE RÔLE DES ÉQUIPEMENTS (onduleur et régulateur de tension) ET
LISTER QUELQUES RISQUES LIÉS À LA NON PROTECTION DE L'ENVIRONNEMENT
DE TRAVAIL**

Compétences visées

- Lister quelques outils de protection de l'environnement de travail;
- Décrire le rôle d'un onduleur et d'un régulateur de tension ;
- Lister quelques risques liés à la non protection de l'environnement de travail et donner quelques solutions ;
- Décrire l'utilisation d'un équipement de protection dans un environnement informatique

Situation problème/ situation de vie :

Pendant le montage vidéo, votre grand frère qui utilisait un desktop et qui été à la dernière phase (rendu) a été surpris par la coupure du courant électrique. Le courant une fois revenu, il redémarrera son ordinateur et constate que ses données sont perdues et que son ordinateur redémarre tout seul.

Consigne :

1. Nommer quelques équipements de protection utilisés dans un environnement informatique ;
2. Proposer à votre grand frère un moyen pour éviter la perte de données en cas de coupure de courant ;
3. Identifier un risque lié à la non protection de son environnement de travail.

Résultats attendus :

1. utiliser un onduleur, parasurtenseur, régulateur de tension, parafoudre
2. utiliser un onduleur
3. perte de données

I- Quelques outils de protection de l'environnement de travail

1- outils matériels

Il existe plusieurs outils matériels de protection de l'environnement informatique. Entre autres nous pouvons citer : l'onduleur, le régulateur de tension, le parasurtenseur, le parafoudre etc.



Schéma de quelques matériels de protection

2- outils logiciels

Pour la protection de l'environnement informatique, nous avons comme outils logiciel : l'antivirus, le pare-feu, l'anti spam etc.

II- Rôle d'un onduleur et d'un régulateur de tension

1- rôle de l'onduleur

L'onduleur est un équipement électronique important dans la sécurité des systèmes informatiques.

Il a pour principale rôle de prendre le relais du secteur lorsque des coupures de courant se produisent, laissant ainsi aux utilisateurs le temps de sauvegarder le travail en cours. Il permet également le filtrage, ainsi que la régulation de la tension.

2- rôle d'un régulateur de tension

Un régulateur de tension est un équipement électronique qui maintient à sa sortie, dans certaines limites, une tension constante, indépendamment de la charge et de la tension d'entrée.

Le régulateur de tension a pour rôle : la protection contre les baisses de tensions, la protection contre les courts circuits, une large plage de tension acceptée.

III- Quelques risques liés à la non protection de l'environnement de travail et solution pour chaque

Les risques les plus courants sont :

- **Pannes du matériel** : disque dur, bloc d'alimentation, écran etc. utiliser un outil matériel de protection pour pallier à ces pannes ;

- **Perte de données** liée à une coupure brusque de l'électricité, utiliser un onduleur pour éviter cela ;
- **Des incendies** liés à une surtension électrique, utiliser un régulateur de tension pour éviter ;
- **Dysfonctionnement de certains logiciels** suite à des virus qui sont des logiciels programmés pour nuire qui peuvent s'attaquer physiquement aux postes de travail ou agir en silence, pour récupérer des informations sensibles. La solution ici est l'installation d'un logiciel antivirus professionnel sur tous les postes de travail et Installer régulièrement les mises à jour.
- **Le piratage** qui est l'intrusion dans un système intermédiaire par le biais d'une faille de sécurité au niveau du réseau ou d'un ou plusieurs postes de travail. Pour se protéger, il faut déployer une stratégie de mots de passe complexes, un pare-feu nouvelle génération bien configuré et à jour.

IV- Utilisation d'un équipement de protection dans un environnement informatique : cas d'un onduleur

La procédure d'utiliser d'un onduleur est:

- Connecter le câble d'alimentation de l'appareil à protéger sur l'onduleur ;
- Connecter le câble d'alimentation de l'onduleur sur une prise électrique ;
- Mettre l'onduleur en marche en appuyant son bouton power pour alimenter ton appareil.

Un environnement informatique peut être victime des problèmes dus à l'instabilité du courant électrique, au hacker, au virus et au manque de formation du personnel. Pour pallier à ces problèmes, nous pouvons utiliser des onduleurs, régulateurs de tension tout en mettant en place les techniques de sécurité de données nécessaires.

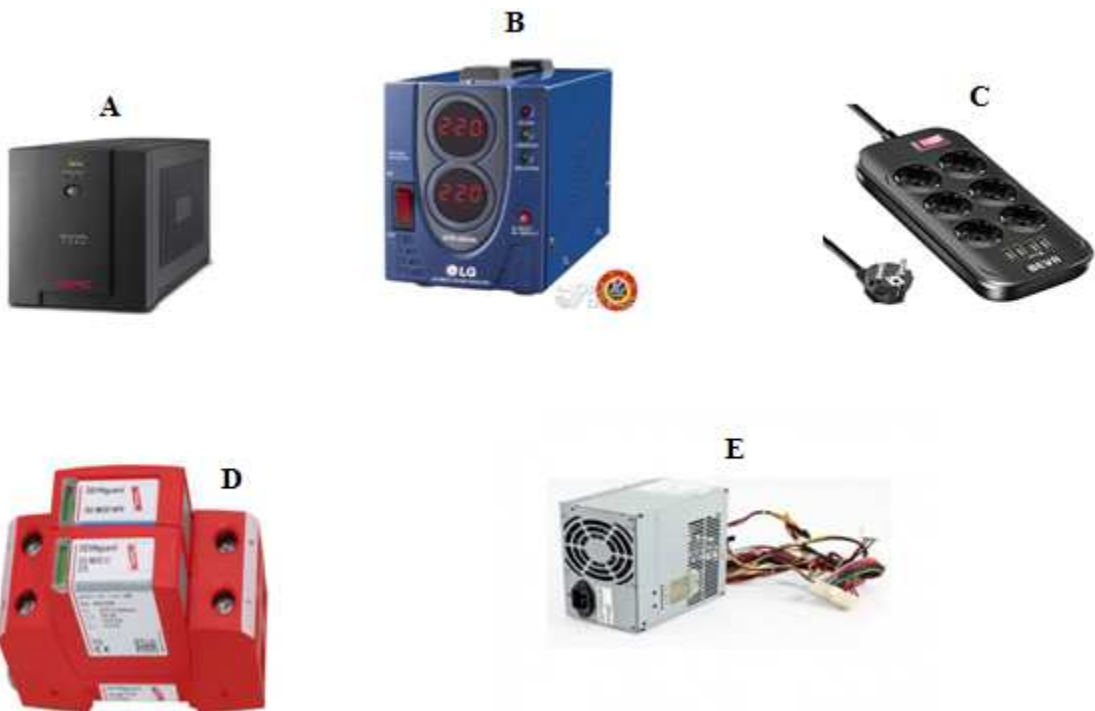
Jeu bilingue :

Français	Anglais
Régulateur de tension	Voltage Stabilizer
Onduleur	UPS (Uninterrupted Power Supply)
Bloc d'alimentation	Power supply

Exercices de consolidation1 :

PDF Compressor Free Version

1. Donner le rôle de chacun des équipements suivants : régulateur de tension et bloc d'alimentation.
2. Identifier par la lettre correspondante les équipements suivants :
 1. Parasurtenseur
 2. Bloc d'alimentation
 3. Onduleur
 4. Parafoudre
 5. Régulateur de tension



3. M. NDJOMO fait face à de nombreuses difficultés liées à de coupures intempestives d'énergie. Il aimerait se procurer d'un équipement qui lui permettra de palier à des situations de coupures brusques d'énergie.
 - a. Proposer un équipement de protection adéquat à la résolution du problème posé par M.NDJOMO.
 - b. Il aimerait installer un onduleur à son ordinateur mais il ne s'y connaît pas. Proposer a M.NDJOMO des étapes qu'il pourrait suivre afin d'y parvenir.

Résolution :

1. **Bloc d'alimentation :** Le bloc d'alimentation est un matériel ayant pour rôle de fournir de l'énergie électrique à un appareil.

Régulateur de tension : a pour rôle de gérer des variations brusques d'énergie (surtension ou sous-tension) afin de la rendre plus stable.

2. 1. C 2. E 3. A 4.D 5.B

a. Onduleur

b. Pour connecter votre onduleur a votre ordinateur, vous pouvez suivre les étapes suivantes :

- Branchez votre onduleur sur l'ordinateur
- Faites un clic droit sur le bureau
- Sélectionnez propriétés
- Choisissez écran de veille
- Cliquez sur gestion d'alimentation
- Sélectionnez alerte et faites le choix entre les deux types d'alertes.

Exercices de consolidation2

Exercice 1

Pendant une séance pratique en salle informatique de votre établissement, survient une coupure brusque du courant électrique qui arrête toutes les machines entraînant ainsi la perte de données et endommageant quelques blocs d'alimentation.

- 1- Identifier la cause de l'arrêt brusque des machines et la perte de donnée après la coupure.
- 2- Donner une cause de la panne du bloc alimentation.
- 3- Nommer un équipement qui pouvait maintenir les ordinateurs allumés permettant ainsi la sauvegarde des données et l'arrêt en toute sécurité des machines.
- 4- Nommer deux équipements qui pouvaient empêcher la panne du bloc alimentation.

Exercice 2

Votre tante travail dans une entreprise où elle garde les documents secrets dans sa machine qui est connectée sur un réseau informatique. En visite dans son bureau, vous constatez que sa machine ne comporte aucune protection.

- 1- Donner le risque encouru par votre tante.
- 2- Proposez-lui une technique pour protéger ses documents contre les pirates.

- 3- Enumérer deux autres risques de la non protection de son environnement de travail et un moyen de protection pour chaque.

Exercice 3

- 1- Identifier l'élément ci-contre
- 2- Donner son rôle
- 3- Décrire son utilisation



COMPETENCES VISEES : à la fin de cette leçon, l'apprenant devra être capable de :

- ❖ Définir tableur et donner quelques exemples de tableurs ;
- ❖ Mettre en forme une feuille de calcul ;

SITUATION PROBLEME



Oncle NGANTCHEU était le comptable de votre père dans son cabinet d'avocat. Pour faciliter la recherche et la compréhension des dossiers reçu il les rangeais dans cet instrument(**figure**) par ordre de réception et par date. Avec l'avènement des TIC, votre père a acheté un ordinateur pour archiver ces travaux de préférence de la même manière que Oncle NGANTCHEU, et surtout parce qu'il a suivi ces collègues dire que c'est plus facile de faire des calculs avec un ordinateur. Il vous fait appel parce

qu'il ne sait pas comment procéder.

1. Donnez un nom à cet instrument.
2. Quel type de logiciel proposerez-vous à votre père ?
3. Citez deux exemples de ce type de logiciel.
4. Définir tableur.

I. NOTIONS DE TABLEUR

A. Définitions Des Concepts

Tableur : est un logiciel permettant d'effectuer automatiquement des calculs sur des données stockées dans un tableau. Il se compose d'une grille de quelques milliers de cases appelées cellules.

Classeur : est un ensemble de feuilles de calcul stockées dans un même fichier d'extension .xls.

Feuille de calcul : est un ensemble de cellules organisées en tableau et repérable par un onglet à son nom.

B. Quelques Exemples De Tableur

On distingue plusieurs types de tableurs, mais les plus connus sont :

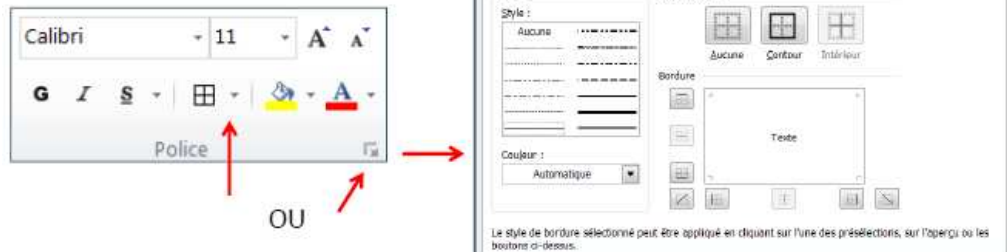
- Microsoft Excel, de la suite bureautique Microsoft Office,
- OpenCalc, de la suite OpenOffice,
- NeoOffice de la suite OpenOffice,
- Sun StarOfficeCalc, de la suite StarOffice,
- IBM/Lotus 1-2-3 de la suite SmartSuite,
- Quattro Pro de la suite Corel WordPerfect Office.

II. MISE EN FORME D'UNE FEUILLE DE CALCUL

On peut effectuer plusieurs mises en forme sur une feuille de calcul et entre autres, nous pouvons élucider:

- ❖ **Ajouter des bordures aux cellules.** Cette mise en forme se fait via la procédure suivante comme illustré par le schéma :

Accueil > Police



- ❖ **Fusionner les cellules.**

Pour faire fusionner des cellules dans Excel, on procède avec méthode comme suit :

Accueil > Alignement > fusionner et centrer. Comme indique cette figure



- ❖ Renvoyer à la ligne automatiquement.

Cette mise en forme selon le schéma suivant :

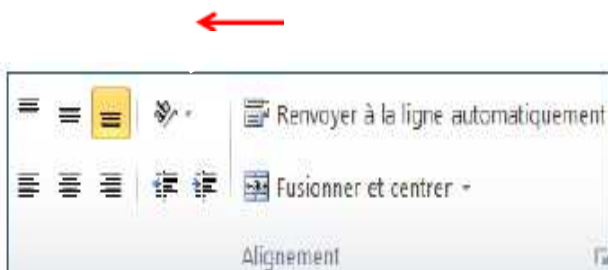
Accueil > Alignement > Renvoyer à la ligne automatiquement



- ❖ Orientation du texte.

Pour donner une orientation a un texte dans Excel, on procède comme suit.

Accueil > Alignement > choisir son type d'orientation sur l'icône (ab)



RESOLUTION DE LA SITUATION PROBLEME

1. Cet instrument est un classeur.
2. Je lui conseillerai un tableur.
3. On peut citer **lotus1,2,3 ; Microsoft Excel**.
4. **Un tableur** est un logiciel permettant d'effectuer automatiquement des calculs sur des données stockées dans un tableau.

ENONCE DES EXERCICES

EXERCICE 1

Mettez une croix dans la case qui convient.

Logiciels	Tableurs
OpenCalc	
Microsoft PowerPoint	
Paint 3D	
Sun StarOfficeCalc	
Microsoft Word	
Quattro Pro	
Microsoft Excel	

EXERCICE 2

Complétez en utilisant les mots ou expressions suivants :

Office ; Word ; .xls ; Microsoft Excel ; cellules ; feuilles de calculs ; rapidité ; calculs.

.....1.....est un logiciel permettant d'effectuer automatiquement les calculs dans un tableau. il est constitué de petites cases rectangulaires appelé.....2.....On peut l'assimiler à un cahier donc les pages sont appelés.....3.....Une fois qu'un fichier issu de ce logiciel est enregistré, on peut le localiser avec son extension qui est.....4.....Son atout majeur réside dans sa faculté à effectuer automatiquement les5.....Il fait partie des logiciels les plus utilisés de la suite.....6.....

CORRIGES DES EXERCICES

CORRIGE EXERCICE 1

Logiciels	Tableurs
OpenCalc	X
Microsoft PowerPoint	
Paint 3D	X
Sun StarOfficeCalc	
Microsoft Word	X
Quattro Pro	
Microsoft Excel	X

CORRIGE EXERCICE 2

1. **Microsoft Excel**

3. **.xls**

2. Cellules

3. Feuilles de calculs

4. Calculs

6. Office

PDF Compressor Free Version

Module 1 : SYSTEME INFORMATIQUE**PDF Compressor Free Version****Unité d'Apprentissage 3 : PRODUCTION D'UN CLASSEUR****UE 6: UTILISATION DES FONCTIONS ET DES REFERENCES ENTRE PLUSIEURS FEUILLES(T&TP)****Indicateurs de compétences :**

- Utiliser les fonctions ;
- Utiliser les références entre plusieurs feuilles.

Vocabulaire : SOMME(), PRODUIT(), MOYENNE(), RANG(), SI(), NB.SI(), SOMME.SI() ;**Prerequis** :Tableur, référence, fonction**Contenu**

Situation d'apprentissage	2
I- Utilisation des fonctions	2
1- Notion de fonction.....	2
2- Utilisation des fonctions mathématiques SOMME, SOMME.SI et PRODUIT	2
a- La fonction SOMME.....	2
b- La fonction SOMME.SI	3
c- La fonction PRODUIT	3
3- Utilisation des fonctions statistiques.....	3
a- La fonction MOYENNE.....	4
b- La fonction RANG	4
c- La fonction SI	4
d- La fonction NB.SI.....	5
II- Utilisation des références de cellules entre plusieurs feuilles	5
1- Notion de référence	5
2- Utilisation de références dans une même feuille de calcul.....	5
3- Utilisations de références entre plusieurs feuilles	6
<i>Jeu bilingue</i>	6
Exercices de consolidations	6

PDF Compressor Free Version

Situation d'apprentissage

Enoncé :

Monsieur TIGA est le directeur de l'école publique de CHOFI. Cette école six (06) salles de classe de la SIL au CM2. Compte tenu du fait que la production des bulletins se fait manuellement, il souhaite numériser le processus de production des bulletins à l'aide d'un tableur. Pour cela vous devez effectuer les tâches suivantes :

- 1) Reporter les notes de chaque classe dans une feuille de calcul ;
- 2) Déterminer pour chaque classe la moyenne de chaque élève ainsi son rang ;
- 3) Calculer la moyenne générale de classe et de l'école ;
- 4) Déterminer le nombre des admis par sexes par classe et pour toute l'école ;

Consignes : Décrire la procédure qui permettra à Monsieur TIGA de concevoir les bulletins pour école à l'aide d'un tableur.

Trace écrite

I- Utilisation des fonctions

1- Notion de fonction

Une **fonction** est une commande qui nécessite des valeurs pour pouvoir produire un résultat donné.

2- Utilisation des fonctions mathématiques SOMME, SOMME.SI et PRODUIT

a- La fonction SOMME

La fonction **SOMME** permet d'additionner des nombres pris comme argument. Sa syntaxe est : **SOMME**(nombre1,[nombre2],...).

Où **nombre1**(obligatoire), **nombre2**, ... sont des arguments qui peuvent être une plage de cellules, une référence, une constante, une formule, une matrice ou le résultat d'une formule.

Remarque :

- Si un argument est une matrice ou une référence alors seules les cellules contenant des nombres seront prises en compte. Les cellules vides, les valeurs logiques ou des textes contenus dans la matrice ou la référence ne seront pas pris en compte ;
- Si des arguments représentent des valeurs d'erreur ou des textes, il est possible de les convertir en nombre.

Exemple : Considérons la feuille de ci-dessous

	A	B	C
1	Données		
2	-8		
3	26		
4	52		

5	"12"		
6	VRAI		
7	Formule	Description	Résultat
8	=SOMME(10;2)	Additionne 10 et 2.	12
9	=SOMME("12";52;VRAI)	Additionne 5, 15 et 1. La valeur de texte « 5 » est convertie en nombre et la valeur logique VRAI est convertie en 1.	65
10	=SOMME(A1:A5)	Additionne les valeurs contenues dans les cellules A2 à A4.	70
11	=SOMME(A1:A3;20)	Additionne les valeurs contenues dans les cellules A1 à A3, puis ajoute 20 au résultat.	38

b- La fonction SOMME.SI

La fonction **SOMME.SI** permet d'additionner des valeurs d'une plage suivant un critère particulier. Sa syntaxe est la suivante : **SOMME.SI(plage ;critère ;[somme_plage])**

L'argument **somme_plage** est facultatif. Il représente les cellules où les valeurs doivent réellement être sommées, si on veut ajouter d'autres cellules que celles qui sont spécifiées dans l'argument **plage**.

Exemple : La feuille de calcul ci-dessous représente une fiche comptable d'un commerçant

	A	B	C
1	Prix article	Bénéfice	Impôts
2	100000	7000	250000
3	200000	14000	
4	300000	21000	
5	400000	28000	
6	Formule	Description	Résultat
7	=SOMME.SI(A2:A5;">160000";B2:B5)	Somme des valeurs de bénéfice excédant 160 000	63 000
8	=SOMME.SI(A2:A5;">160000")	Somme des prix article excédant 160 000	900 000
9	=SOMME.SI(A2:A5;300000;B2:B5)	Somme des bénéfices relatifs aux prix articles égales à 300 000	21 000
10	=SOMME.SI(A2:A5;">"&C2;B2:B5)	Somme des bénéfices relatifs aux prix articles supérieures à la valeur de C2.	49 000

c- La fonction PRODUIT

La fonction **PRODUIT** permet la multiplication des nombres pris en tant qu'argument et renvoyant le produit. Sa syntaxe est la suivante :

PRODUIT(nombre1;[nombre2];...) où :

- **nombre1** (obligatoire) est la première plage ou le premier nombre à multiplier ;
- **nombre2** (facultatif) représente les nombres ou les plages supplémentaires à multiplier, jusqu'à 255 arguments au maximum.

Exemple :

Formule	Description
=PRODUIT(A1 ; B1)	Contenue de A1 fois contenue de B2
=PRODUIT(A1:A4 ;B2:B5)	A1*A2*A3*A4*B2*B3*B4*B5
=PRODUIT(C1:C3 ;10)	C1*C2*C3*10

3- Utilisation des fonctions statistiques

a- La fonction MOYENNE

La fonction **MOYENNE** permet de calculer la moyenne des nombres d'une plage ou sélection de cellule et renvoie le résultat (moyenne des nombres). Sa syntaxe est la suivante :

MOYENNE(nombre1 ;[nombre2] ;...) où

- **nombre1** (obligatoire) est le premier nombre ; la première référence ou plage dont on veut obtenir la moyenne ;
- **nombre2** (facultatif) est un nombre ; une référence ou une plage supplémentaire.

Exemple : Les valeurs ci-dessous représentent les notes d'informatique des élèves de la classe de Terminale CD. Déterminer la moyenne de ses notes dans la cellule N1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	12	09	14	05	07	11	17	10	09	12	15	05	16	10,9230769

La formule est : **=MOYENNE(A1:M1)** et le résultat est : **10,9230769**

b- La fonction RANG

La fonction **RANG** renvoie le rang d'un nombre dans une liste d'arguments. Ce rang est obtenu par comparaison de la valeur du nombre par rapport aux autres valeurs de la liste. Sa syntaxe est : **RANG(nombre,référence,[ordre])** où :

- **Nombre** (obligatoire), représente le nombre dont vous voulez connaître le rang ;
- **Référence** (obligatoire). représente une matrice, ou une référence à une liste de nombres. Les valeurs non numériques dans référence sont ignorées ;
- **ordre**(facultatif)représente un numéro qui spécifie comment déterminer le rang de l'argument nombre.

Exemple : considérons les valeurs de l'exemple précédant. Déterminons le rang de chaque élève.

	A	B	C
1	Note	Formule Rang	Rang
2	12	=RANG(A2;A\$2:A\$14)	5
3	9	=RANG(A3;A\$2:A\$14)	9
4	14	=RANG(A4;A\$2:A\$14)	4
5	5	=RANG(A5;A\$2:A\$14)	12
6	7	=RANG(A6;A\$2:A\$14)	11
7	11	=RANG(A7;A\$2:A\$14)	7
8	17	=RANG(A8;A\$2:A\$14)	1
9	10	=RANG(A9;A\$2:A\$14)	8
10	9	=RANG(A10;A\$2:A\$14)	9
11	12	=RANG(A11;A\$2:A\$14)	5
12	15	=RANG(A12;A\$2:A\$14)	3
13	5	=RANG(A13;A\$2:A\$14)	12
14	16	=RANG(A14;A\$2:A\$14)	2

Remarques :

- La fonction RANG attribue le même rang aux nombres en double. Si un rang apparaît en double le rang immédiatement suivant sera sauté par exemple le rang 12 apparaît deux fois le rang suivant sera donc 14 et il n'y aura pas de rang 13.
- Pour appliquer le rang à une plage, il est préférable d'utiliser les références mixtes ou absolues pour référencer la plage.

c- La fonction SI

La fonction **SI** renvoie une valeur si une condition spécifique est vérifiée ou une autre valeur si la condition n'est pas vérifiée. Sa syntaxe est :

SI(test_logique, [valeur_si_vrai], [valeur_si_faux])

Exemple : la formule **=SI(A1>10,"ADMIS","REFUSE")** renvoie "ADMIS" si A1 est supérieur à 10, et "REFUSE" si A1 est inférieur ou égal à 10.

d- La fonction NB.SI

La fonction **NB.SI** permet de dénombrer les cellules d'une plage qui répondent à un critère spécifique que vous spécifiez. On peut par exemple compter le nombre de cellules où c'est marqué **M** ou **F**. La syntaxe de la fonction **NB.SI** est :

NB.SI(plage; critères)

Exemple : Déterminer le nombre de garçons et de filles dans la liste ci-dessous

	A	B	C
1	Note	Genre	
2	12	M	
3	9	M	
4	14	F	
5	5	M	
6	7	F	
7	11	F	
8	17	M	
9	10	F	
10	9	M	
11	12	F	
12	15	F	
13	5	M	
14	16	F	
15		Formule	Résultat
16	Garçon	=NB.SI(B2:B14;"M")	5
17	Fille	=NB.SI(B2:B15;"F")	8

II- Utilisation des références de cellules entre plusieurs feuilles

1- Notion de référence

Une **référence de cellule** est le fait d'indiquer une cellule ou une plage de cellule d'une feuille de calcul qui pourra être utilisée par un tableur pour manipuler les valeurs ou les données qui contiennent ces cellules dans une formule.

Une référence peut faire allusion à :

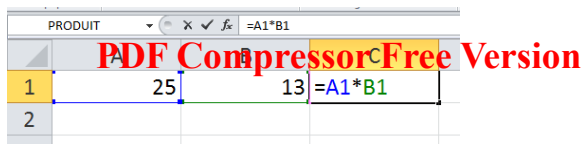
- Une donnée d'une cellule. **Exemple :** A1 = valeur contenu dans la cellule A1 ;
- Des données d'une plage de cellules. **Exemples :** \$A\$1:\$A\$10 contenu de la cellule A1 jusqu'à celui de A10 ou A1:B2;C4 contenus des cellules A1, B2 et C4 ;
- Des données de cellules d'une autre feuille de calcul d'un même classeur. **Exemple :** Feuil4!D7 contenu de la cellule D7 de la Feuille 4.

2- Utilisation de références dans une même feuille de calcul

Pour utiliser les références de cellules d'une même feuille de calcul dans une formule on pourra procéder de la manière suivante :

- Sélectionner la cellule dans laquelle on veut entrer la formule ;
- Taper sur le signe égal(=) pour entrer la formule ;
- Insérer les références de cellules intervenant dans la formule en cliquant sur la cellule ou en saisissant la référence de la cellule.

Exemple :



3- Utilisations de références entre plusieurs feuilles

Il est possible d'utiliser les références de cellules d'une feuille de calcul dans d'autres feuilles de calcul en ajoutant le nom de la feuille de calcul suivi d'un point d'exclamation (!) au début de la référence de cellule.. Pour ce faire, on procédera de la manière suivante :

- Sélectionner la cellule dans la quelle l'on veut entrer la formule ;
- Taper sur le signe égal(=) pour entrer la formule ;
- Insérer le nom de feuille de calcul suivis du point d'exclamation et de la références de cellule intervenant dans la formule en cliquant respectivement sur le nom de feuille de calcul en suite sur la cellule de la formule ou en saisissant le nom et la référence de la cellule.

Exemple :

	A
1	=MOYENNE(Feuil7!A2:Feuil7!A14)
2	

Cette formule permet de déterminer la moyenne de la plage des cellules de A1 à A14 de la Feuil7.

Jeu bilingue

<i>Français</i>	<i>Anglais</i>
SOMME	SUM
SOMME.SI	SUMIF
PRODUIT	PRODUCT
MOYENNE	AVERAGE
RANG	RANK
SI	IF
NB.SI	COUNTIF

Exercices de consolidations

Exercice 1

Le tableau ci-dessous représente les résultats à l'issus des examens de fin d'année en classe de Terminale C. A l'aide d'un tableur, effectuer les tâches suivantes :

N°	Noms et prénoms	Sexe	Note /20	Rang	Obervations
1	Adamou Gilbert	M	12		
2	Alomo Jean Calvin	M	13,5		
3	Badawé Dieudonné	M	09		
4	Bapowa Delphine	F	17		
5	Daïpa Angeline	F	11		
6	Djeukam Sangou	M	07,25		
7	Mariatou	F	06		
8	Maï-awé Silvie	F	08,75		
9	Tchaga Bernard	M	10		
10	Wilba Albert	M	15		
Moyenne des notes					

A l'aide d'un tableur, effectuer les tâches suivantes :

- 1) Reproduire le tableau ci-dessus dans Microsoft Excel.
- 2) A l'aide des fonctions de Microsoft Excel, déterminer :
 - a) La moyenne des notes ;
 - b) Le rang de chaque élève ;
 - c) Le nombre de fille et de garçon de la classe ;
- 3) Ce tableau doit également ressortir en observation les mentions « **ADMIS** » pour les notes supérieures ou égales à 10/20 et « **REFUSE** » dans le cas contraire. Ecrire la formule qui permettra de remplir cette colonne.

Exercice 2

A partir du problème posé dans la situation d'apprentissage, concevoir d'un projet de bulletin numérique pour une école primaire à six classes à l'aide d'un tableur.

MODULE 1 : SYSTEMES INFORMATIQUES

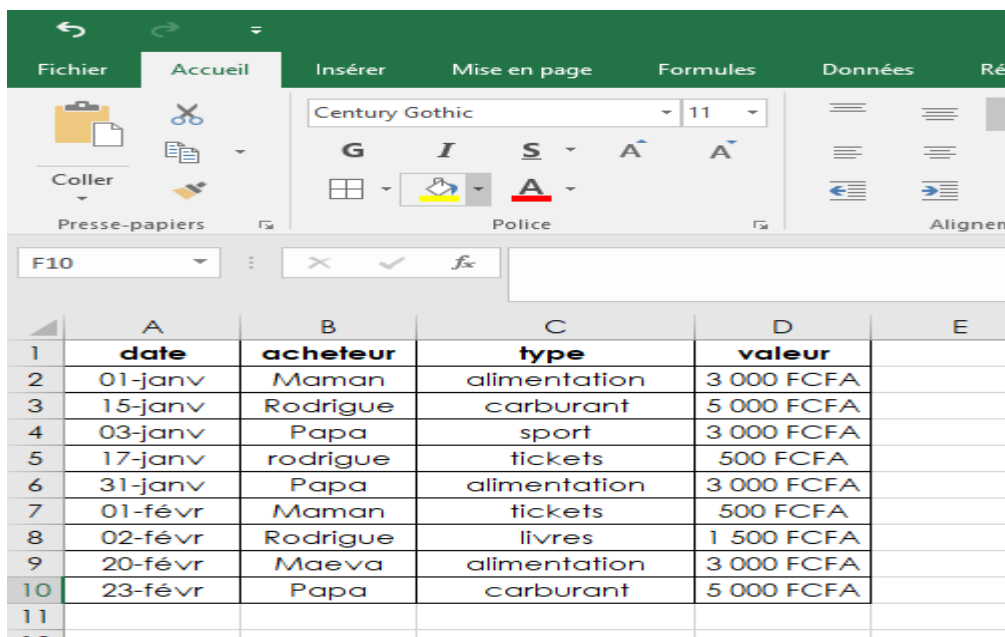
PDF Compressor Free Version

UNITE D'APPRENTISSAGE 3 : PRODUCTION ND'UN CLASSEUR
UE 7: Utilisation d'un tableau croisé dynamique, définir une zone d'impression et protéger l'accès à une feuille de calcul
Compétences visées :

- Utilisation d'un tableau croisé dynamique ;
- Définir une zone d'impression et protéger l'accès à une feuille de calcul

Situation problème :

L'état de livraison de M. Belinga via Ms Excel du 1^{er} janvier au 23 février se présente par le tableau ci-dessous. Ce qui permet à ce dernier de facilement déduire la valeur totale de chaque acheteur par mois, par type et par valeur car, ce fichier ne comporte que 09 enregistrements. Cependant, il se demande : comment il procéderait pour effectuer la même opération lorsque ce fichier aura 100, 200 enregistrements, voire plus; de même pour empêcher l'accès à la feuille de calcul et pouvoir imprimer les livraisons d'un acheteur de son choix



	A	B	C	D	E
1	date	acheteur	type	valeur	
2	01-janv	Maman	alimentation	3 000 FCFA	
3	15-janv	Rodrigue	carburant	5 000 FCFA	
4	03-janv	Papa	sport	3 000 FCFA	
5	17-janv	rodrigue	tickets	500 FCFA	
6	31-janv	Papa	alimentation	3 000 FCFA	
7	01-févr	Maman	tickets	500 FCFA	
8	02-févr	Rodrigue	livres	1 500 FCFA	
9	20-févr	Maeva	alimentation	3 000 FCFA	
10	23-févr	Papa	carburant	5 000 FCFA	
11					

Consignes:

1. Déduire la valeur totale de Maman en illustrant aussi la valeur par mois, type. (Réponse :

Maman	
janv	3000
alimentation	3000
févr	500
tickets	500
Total Maman	3500

2. Proposer une solution à ce dernier pour pouvoir facilement faire la même déduction lorsque ce fichier aura 100, 200 enregistrements, voire plus. (Réponse : **Utilisation d'un tableau croisé dynamique**)
3. Déduire comment il procèdera pour pouvoir respectivement empêcher l'accès à sa feuille de calcul et imprimer les livraisons d'un seul acheteur. (Réponse : **Protéger la feuille de calcul et définir une zone d'impression avant impression**)

Trace écrite

1. Notion de tableau croisé dynamique

a) Définition et rôle

Un tableau croisé dynamique (TCD ou Rapport) est un outil capable de calculer, de synthétiser et d'analyser des données, ce qui permet de voir des comparaisons, des motifs et des tendances dans les données.

Il est dynamique car toute modification de la source entraîne la mise à jour du rapport, lorsque l'actualisation est déclenchée.

b) Création d'un tableau croisé dynamique

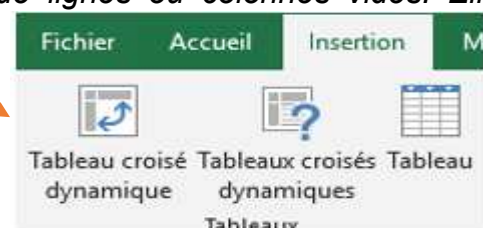
Pour créer un tableau croisé dynamique, l'on procède comme suit :

- Sélectionnez les cellules à partir desquelles vous voulez créer un tableau croisé dynamique.

A	B	C	D
date	acheteur	type	valeur
01-janv	Maman	alimentation	3 000 FCFA
15-janv	Rodrigue	carburant	5 000 FCFA
03-janv	Papa	sport	3 000 FCFA
17-janv	rodrigue	tickets	500 FCFA
31-janv	Papa	alimentation	3 000 FCFA
01-févr	Maman	tickets	500 FCFA
02-févr	Rodrigue	livres	1 500 FCFA
20-févr	Maeva	alimentation	3 000 FCFA
23-févr	Papa	carburant	5 000 FCFA

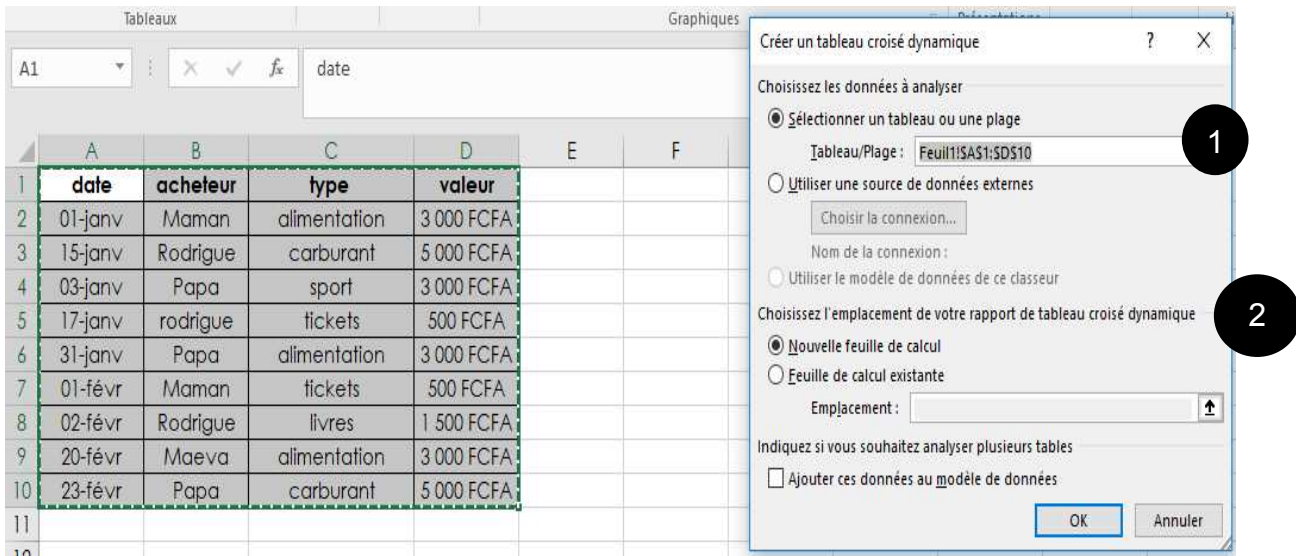
Remarque : Vos données ne peuvent pas comporter de lignes ou colonnes vides. Elles ne peuvent comporter qu'une seule ligne de titre.

- Sélectionnez **Insertion > Tableau croisé dynamique**.



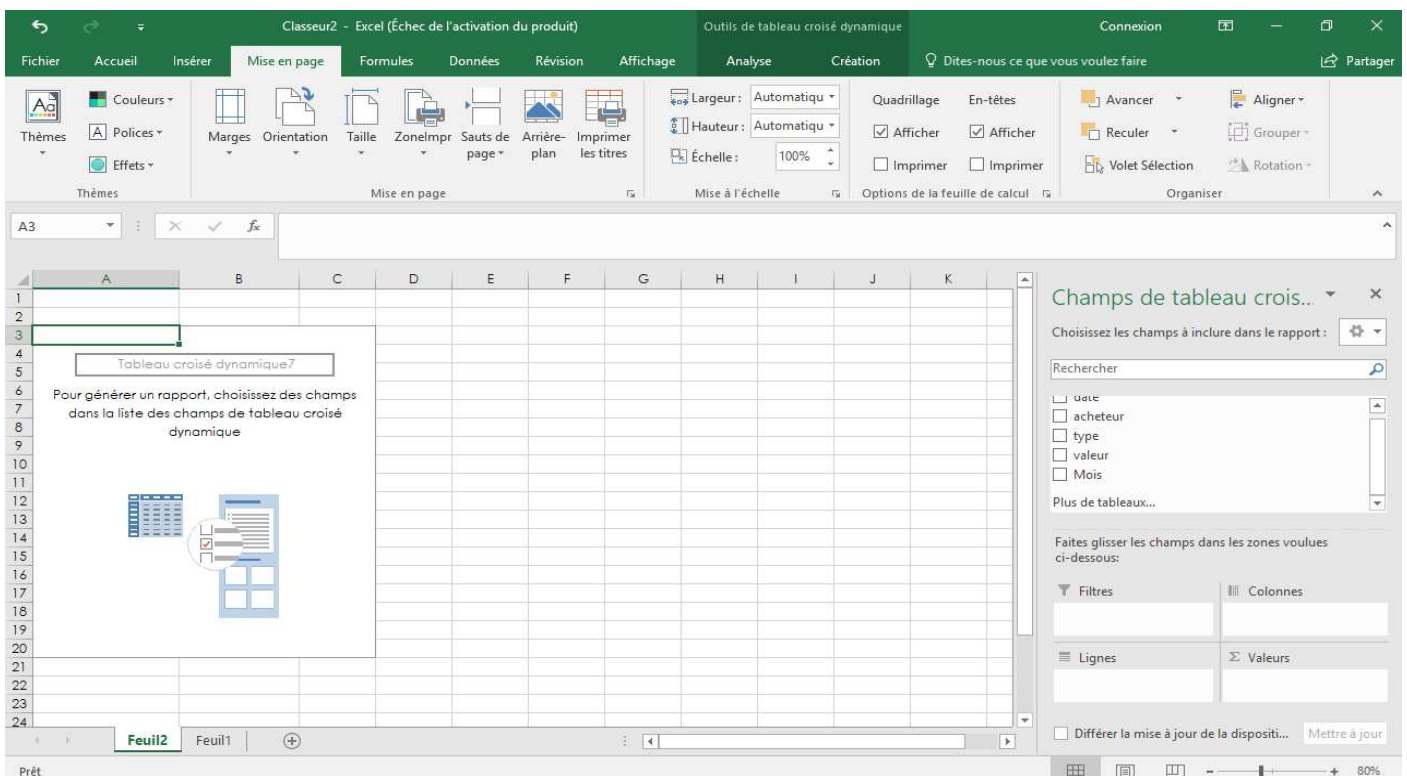
- Sous **Choisissez les données à analyser**, cliquez sur **Sélectionner un tableau ou une plage**.

PDF Compressor Free Version



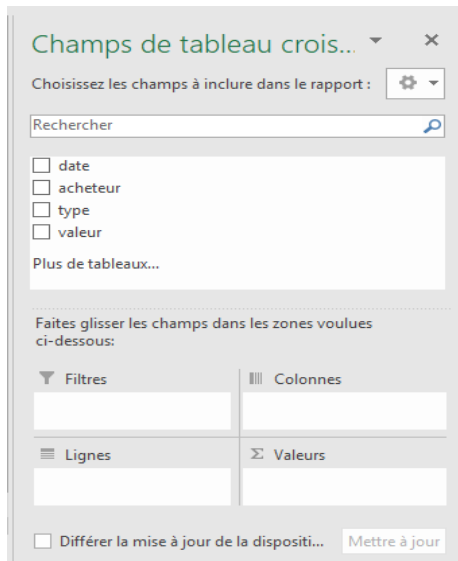
- Dans **Tableau/plage**, vérifiez la plage de cellules. 1
- Sous 2, **sélectionnez l'emplacement de votre rapport de tableau croisé dynamique**, sélectionnez **Nouvelle feuille de calcul** pour placer le tableau croisé dynamique dans une nouvelle feuille de calcul **Feuille de calcul existante**, puis sélectionnez où le tableau croisé dynamique doit apparaître.
- Sélectionnez **OK**.

Un tableau croisé vide se présente de façon similaire à celui indiqué ci-dessous.



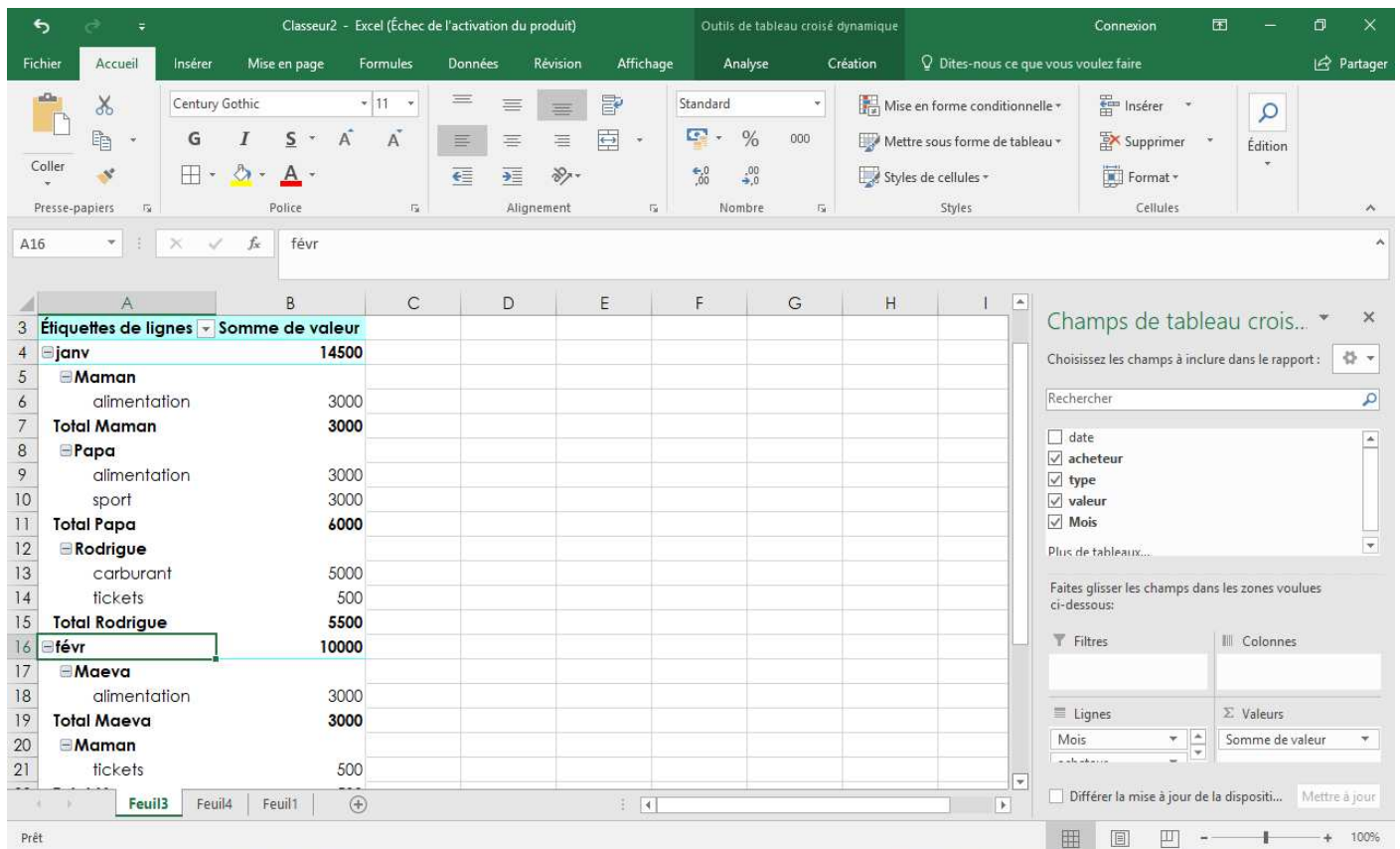
- Pour ajouter un champ à votre tableau croisé dynamique, activez la case à cocher du nom de champ dans le volet **Champs de tableau croisé dynamique**.

PDF Compressor Free Version



Remarque : Les champs sélectionnés sont ajoutés à leur zone par défaut : les champs non numériques sont ajoutés à **Lignes**, les hiérarchies de date et d'heure sont ajoutées à **Colonnes**, et les champs numériques sont ajoutés à **Valeurs**.

- Pour déplacer un champ d'une zone à une autre, faites-le glisser vers la zone cible (**Filtres**, **Colonnes**, **Lignes**, **Valeurs**).



En cochant d'abord mois, ensuite acheteur, puis type et enfin valeur nous avons le résultat suivant par mois.

NB : Pour avoir le visuel de la situation problème, il faut d'abord cocher sur acheter ensuite sur le reste ou bien faire glisser acheteur au-dessus de mois dans zone de champ Lignes.

2. Protection d'une feuille de calcul

Pour empêcher d'autres utilisateurs de modifier, déplacer ou supprimer des données dans une feuille de calcul, vous pouvez verrouiller les cellules de votre feuille de calcul Excel, puis protéger la feuille avec un mot de passe.

La protection d'une feuille de calcul est un processus en deux étapes. La première étape consiste à **déverrouiller les cellules** que les autres utilisateurs peuvent modifier. Vous pouvez ensuite **protéger la feuille de calcul avec ou sans mot de passe**.

Étape 1 : déverrouiller les cellules qui doivent être modifiables

- Dans votre fichier Excel, sélectionnez l'onglet de la feuille de calcul à protéger.
- Sélectionnez les cellules que les autres utilisateurs peuvent modifier.

NB: Pour sélectionner plusieurs cellules non contiguës, vous pouvez appuyer sur **Ctrl+Clic gauche**

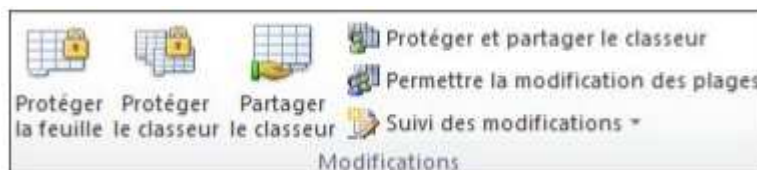
- Cliquez avec le bouton droit n'importe où dans la feuille, puis sélectionnez **Format de cellule** (ou utilisez **Ctrl+1**), puis accédez à l'onglet **Protection** et désactivez **Verrouillé**.



Étape 2 : protéger la feuille de calcul

Pour protéger votre feuille de calcul, vous devez procéder comme suit.

- Dans l'onglet **Révision**, cliquez sur **Protéger la feuille**.



- Dans la liste Autoriser tous les utilisateurs de cette feuille à, **sélectionnez les éléments que les utilisateurs pourront modifier**.

PDF Compressor Free Version



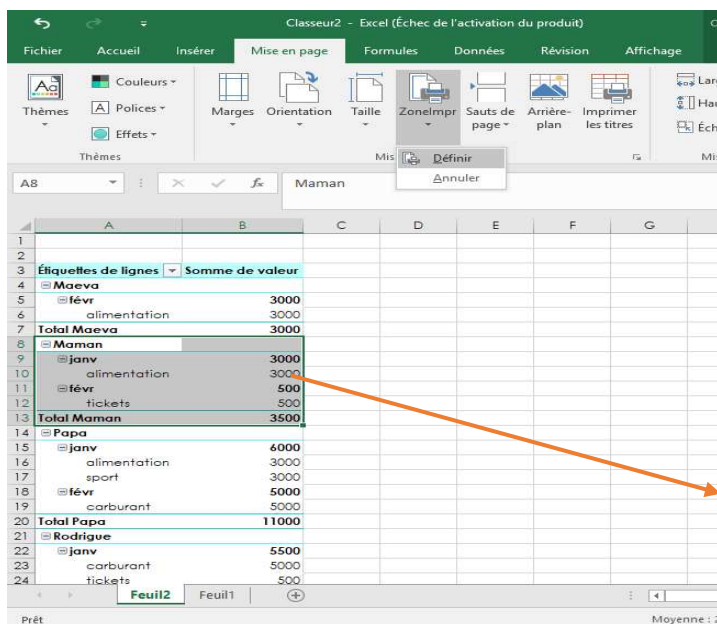
- Vous pouvez également taper un mot de passe dans la zone **Mot de passe pour ôter la protection de la feuille**, puis cliquer sur **OK**. Retapez le mot de passe dans la boîte de dialogue **Confirmer le mot de passe**, puis cliquez sur **OK**.

3. Définition d'une zone d'impression

Une **zone d'impression** correspond à une ou plusieurs plages de cellules que vous spécifiez à imprimer lorsque vous ne souhaitez pas imprimer la feuille de calcul entière.

Pour définir une zone d'impression sous Excel, on procède comme suit :

- Dans la feuille de calcul, sélectionnez les cellules que vous voulez définir comme zone d'impression.
- Sous l'onglet **Mise en page**, dans le **groupe Mise en page**, cliquez sur **Zone d'impression (zoneImpr)**, puis cliquez sur **Définir**.



8	Maman	
9	janv	3000
10	alimentation	3000
11	févr	500
12	tickets	500
13	Total Maman	3500

PDF Compressor Free Version

Jeu Bilingue :

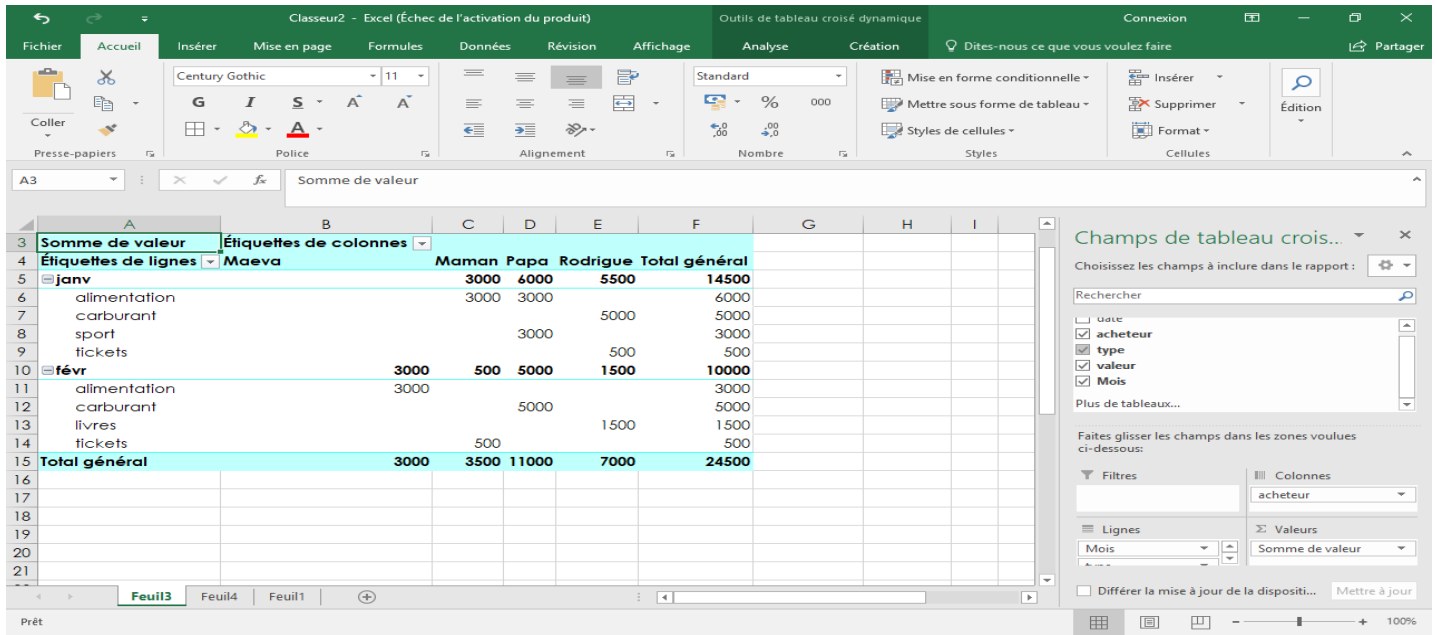
Mots en français	Traduction en anglais
Feuille de calculs	Worksheet
Plages de cellules	Cell ranges
Zone d'impression	Printing area
Tableau croisé dynamique	Pivot table

Exercice 1 :

Reprendre le tableau du fichier de la situation problème pour construire le tableau statistique du chiffre d'affaires par mois et par acheteur pour chaque type et imprimer l'état du mois de janvier.

Résolution exercice 1 :

Suivre les étapes de création d'un tableau croisé dynamique jusqu'à ce letableaucroisé vide se présente, **en suite faire glisser** acheteur dans la zone **colonne** et **terminer en cochant** mois, type et valeur.



Pour imprimer l'état du mois de janvier, deux cas de figure se présente à vous :

- Soit vous définissez une zone d'impression qui couvre la mois de janvier avec les acheteurs avant de lancer l'impression

➤ Soit vous appliquez un filtre sur **Étiquettes de lignes** en cliquant sur l'orientation inférieure

4 **Étiquettes de lignes**, puis décochant **février** avant lancer l'impression.

PDF Compressor Free Version

Exercice 2 :

Un commerçant de vin a regroupé dans une liste le chiffre d'affaires réalisé par mois, par région et par appellation sur ces ventes de vin.

Il vous demande de protéger son fichier, d'imprimer les états des régions du Centre et de l'EST, de même construire les tableaux statistiques suivants :

- Chiffre d'affaires par mois et par région pour chaque appellation
- Chiffre d'affaires par appellation et par région

	A	B	C	D	E
	Mois	Region	Appellation	C.A.	
2	Janvier	Sud	Vins de Loire	20000	
3	Janvier	Sud-Ouest	Bordeaux	15000	
4	Janvier	Nord-Ouest	Alsace	30000	
5	Janvier	Centre	Bordeaux	30000	
6	Fevrier	Sud	Bordeaux	25000	
7	Fevrier	Centre	Alsace	15000	
8	Fevrier	Sud-Ouest	Vins de Loire	20000	
9	Fevrier	Nord-Ouest	Vins de Loire	15000	
10	Fevrier	Sud	Bordeaux	15000	
11	Mars	Est	Vins de Loire	30000	
12	Mars	Nord-Ouest	Bordeaux	15000	
13	Mars	Sud	Alsace	20000	
14	Mars	Centre	Vins de Loire	20000	
15	Avril	Sud-Ouest	Bordeaux	30000	
16	Avril	Est	Vins de Loire	14000	
17	Avril	Centre	Alsace	10000	
18					
19					

Résolution exercice 2 :

- Chiffre d'affaires par mois et par région pour chaque appellation

Suivre les étapes de création d'un tableau croisé dynamique jusqu'à ce le tableau croisé vide se présente, **en suite faire glisser Mois dans la zone colonne et terminer en cochant Région et CA**

	Janvier	Fevrier	Mars	Avril	Total général
Centre	30000	15000	20000	10000	75000
Est			30000	14000	44000
Nord-Ouest	30000	15000	15000		60000
Sud	20000	40000	20000		80000
Sud-Ouest	15000	20000		30000	65000
Total général	95000	90000	85000	54000	324000

Pour imprimer l'état des régions du Centre et de l'EST, deux cas de figure se présente à vous :

➤ Soit vous définissez **une zone d'impression qui couvre les régions du Centre et de l'EST avec les différents mois avant de lancer l'impression**

➤ Soit vous appliquez un filtre sur **Étiquettes de lignes** en cliquant sur l'orientation inférieure

4 **Étiquettes de lignes**, puis décochant les Régions du Nord-Ouest, Sud et Sud-Ouest **avant** lancer l'impression.

- Chiffre d'affaires par appellation et par région

PDF Compressor Free Version

Suivre les étapes de création d'un tableau croisé dynamique jusqu'à ce le tableau croisé vide se présente, en suite faire glisser Appellation dans la zone colonne et terminer en cochant Région et C.A.

The screenshot shows an Excel worksheet with a PivotTable and the PivotTable Fields task pane on the right.


PivotTable Data:

Étiquettes de lignes	Alsace	Bordeaux	Vins de Loire	Total général
Centre	25000	30000	20000	75000
Est			44000	44000
Nord-Ouest	30000	15000	15000	60000
Sud	20000	40000	20000	80000
Sud-Ouest		45000	20000	65000
Total général	75000	130000	119000	324000

PivotTable Fields Task Pane:

- Champs de tableau croisé...
 - Choisissez les champs à inclure dans le rapport :
 - Rechercher
 - ☐ Mois
 - ☒ Région
 - ☒ Appellation
 - ☒ C.A.
 - Plus de tableaux...
- Faites glisser les champs dans les zones voulues ci-dessous:
 - Filtres
 - Colonnes: Appellation
 - Lignes: Région
 - Valeurs: Somme de C.A.
- ☐ Différer la mise à jour de la dispositi... Mettre à jour

Pour imprimer l'état des régions du Centre et de l'EST, deux cas de figure se présente à vous :

- Soit vous définissez **une zone d'impression qui couvre les régions du Centre et de l'EST avec les différentes appellations** avant de lancer l'impression
- Soit vous appliquez un filtre sur **Étiquettes de lignes** en cliquant sur l'orientation inférieure , puis décochant les Régions du **Nord-Ouest, Sud et Sud-Ouest** avant lancer l'impression.

Pour protéger votre feuille de calcul, appliquez la procédure (cf trace écrite)

MODULE 1 : SYSTEMES INFORMATIQUES
PDF Compressor Free Version
UA 3 : PRODUCTION D'UN CLASSEUR

UE 8 : PRODUIRE UNE FACTURE, UN BULLETIN DE NOTE OU DE PAIE

Compétences visées :

- Reproduire une facture, un devis, un bulletin de paie ou de notes ;
- Définir une zone d'impression ;
- Imprimer une feuille de calculs.

Situation problème : lors des longs congés du covid-19, vous aidez votre papa à la boutique. Dans le but de réduire le constat avec les factures papiers, il vous demande de reproduire la facture papier dans l'ordinateur en ajoutant des formules pour ne plus utiliser de calculatrice.

	A	B	C	D	E	F	G	H
1								
2		FACTURE					N°	
3		M.				DATE		
4		QTE	DESIGNATION			P. U.	P. TOTAL	DETAILS
5								
6								
7								
8								
9								
10								
11								
12					NET A PAYER			
13								

1. Quel logiciel peux-tu utiliser pour reproduire cette facture ? (Microsoft Excel)
2. Quels sont les éléments qui ont l'alignement centre, gauche ou droite ? (centre : facture, QTE, désignation, P.U., P.TOTAL et détails ; gauche : N° et M.; droite : date et net à payer)
3. Comment faire apparaître les bordures sur une partie de facture ? (sélectionner les cellules et cliquer bordures puis tous les bordures)
4. Quels sont les formules mathématiques pour avoir le P.TOTAL et le NET A PAYER ? (QTE * P.U. ; SOMME des P.TOTAL)
5. Donner la formule Excel qu'on doit retrouver dans la cellule G5. (=B5*F5)
6. Comment faire pour reproduire cette formule dans les autres cellules du P. Total ? (faire la recopier)
7. Donner la formule Excel pour le NET A PAYER. (SOMME (B5 : B11))
8. Donner la procédure pour définir la zone d'impression sur une page; (cliquer sur mise en page et choisir le nombre de pages en horizontal et en vertical)
9. Donner la procédure pour imprimer cette facture. (cliquer sur fichier puis sur imprimer)

Trace écrite

Grâce à Excel, nous pouvons créer ou reproduire facilement une facture, un devis, un bulletin de paie ou de notes, et avec certaines formules bien placées, nous pourrions automatiser plusieurs actions.

I. LA SAISIE DES DONNEES

1. Saisie

- Cliquer sur la cellule
- Taper le texte, le nombre ou la date
- Valider avec « entre » ou se déplacer dans la prochaine cellule à saisir

2. Modifier le contenu d'une cellule

- Cliquer deux fois sur la cellule à modifier ou cliquer sur la cellule et cliquer dans la barre de formule
- Apporter les modifications
- Valider

3. Calcul sur cellules

- Etre dans la cellule résultat
- Taper =
- Cliquer (ou se placesur) la 1^{er} cellule
- Taper un signe arithmétique
- Cliquer la 2^{ème} cellule
- Procéder ainsi pour chaque cellule
- Valider

4. Saisie d'une fonction

- Etre sur la cellule résultat
- Taper =
- Taper le nom de la fonction et ouvrir la parenthèse
- Sélectionner les cellules concernées
- Fermer la parenthèse
- Valider

Quelques fonctions

- ✎ Fonction pour la date : =AUJOURDHUI ()
- ✎ Fonction pour l'addition : =SOMME (X ; Y)
- ✎ Fonction pour la multiplication : =PRODUIT (X ; Y)
- ✎ Fonction pour la moyenne : =MOYENNE (X ; Y ; ...)
- ✎ Fonction pour le rang : = RANG (nombre ; plage ; ordre)
- ✎ Fonction pour choix d'une opération parmi plusieurs : = SI (question ; traitement si vrai ; traitement si faux)
- ✎ Formule pour le maximum : = MAX (X ; Y ; ...)

II. MISE EN FORME DES CELLULES

1. La barre d'outils

Après avoir sélectionné les cellules, on fait le choix de la police, la taille et la couleur du texte, activer ou désactiver Gras, Italique ou souligne, choisir l'alignement gauche, centre ou droit

2. Fusionner des cellules

- Sélectionner les cellules
- Cliquer sur le bouton fusionner dans le ruban alignement

3. Bordure

- Sélectionner les cellules à encadrer
- Cliquer la flèche à droite de l'icône bordures
- Cliquer un style de bordure

4. couleur de fond

- Sélectionner les cellules
- Cliquer la flèche à droite de l'icône couleur de remplissage
- Cliquer une couleur de fond

5. Reproduire une mise en forme

- Sélectionner une cellule contenant la mise en forme à reproduire
- Cliquer l'icône reproduire la mise en forme
- Sélectionner les cellules qui seront dotées des mêmes formats
- Relâcher

III. MISE EN PAGE ET IMPRESSION

1. Marges – papier – orientation

- Etre dans la feuille à imprimer
- Cliquer sur mise en page
- Cliquer sur marges (pour modifier le pourtour de la page), orientation (portrait ou paysage) ou taille (pour le choix du papier à imprimer)
- Cliquer OK

2. Mise à l'échelle

- Etre dans la feuille à imprimer
- Cliquer sur mise en page
- Dans le ruban de mise à l'échelle rentrer le nombre de pages en largeur et en hauteur

3. Créer un entête ou pied de page

- Cliquer sur mise en page
- Cliquer l'onglet entête/ pied de page (utiliser un entête ou un pied de page intégré)
- Cliquer OK

4. Aperçu avant impression

- Cliquer sur fichier
- Cliquer sur aperçu avant impression
- Cliquer sur fermer

5. Impression

- Cliquer sur fichier
- Cliquer sur imprimer
- Modifier ou non le nombre de copies
- Cliquer sur ok

IV. DESCRIPTION DE QUELQUES DOCUMENTS

1. Une facture

La facture est le document établi par le vendeur et adressé au client à la suite d'une vente, qui détaille le type de marchandise vendu, la quantité, le décompte du prix et la somme totale due par le client. Les informations minimales qui doivent apparaître sont :

- Les éléments d'identification du vendeur : raison sociale, adresse, numéro de registre du commerce et des professions...
- Les éléments d'identification de la facture : date, numéro...
- Les références de la commande ayant déclenché la vente.
- Le rappel des conditions de vente (transport, livraison, paiement...).

- La description détaillée des marchandises : désignation, quantité, prix unitaire hors taxe(HT), montant total hors taxe, réduction de prix, TVA, total toutes taxes comprises(TTC) et net à payer

PDF Compressor Free Version

2. Un bulletin de paie

Le bulletin de paie est le document qui atteste du versement du salaire à un employé à une date précise. Les informations minimales qui doivent apparaître dans un bulletin de paie sont :

- le nom du salarié et le poste qu'il occupe ;
- le nom et les coordonnées de la structure « employeur » (nom, adresse, N° d'immatriculation sociale et fiscale) ;
- la période de paie concernée (ex : du 1er au 31 janvier 2019) ;
- le salaire de base, les éventuelles indemnités additionnelles au salaire de base (allocations familiales, indemnités de transport...), les retenues faites et le salaire net à verser ;
- le coût salarial total (salaire net + charges employeur + charges employé) ;
- la signature de l'employé ;
- la signature et le cachet de l'employeur ;
- la date d'émission de la fiche de paie ;
- le numéro matricule de l'employé ;
- la situation matrimoniale/nombre d'enfant(s) à charge.

3. Un bulletin de note

Le bulletin de note est un document périodique portant mention des notes d'un élève. Les informations minimales qui doivent apparaître sont :

- L'entête : coordonné de l'établissement, l'année scolaire, la classe et l'identité de l'élève
- Le corps : les matières, les moyennes, les appréciations, le nom et signature des enseignants
- Le pied de page : la moyenne finale, appréciation globale, vie scolaire, les décisions et le cachet de l'établissement

JEU BILINGUE

Mot en français	Traduction en anglais
Bulletin de note	Report card
Facture	bill
Bulletin de paie	payslip
Impression	printing

EXERCICES

Exercice 1 : votre petit frère vous demande de l'aide pour confessional son bulletin de la 1^{ère} évaluation car dans son lycée on ne remet que les bulletins du trimestre.

1. Quel sont les informations que doit fournir votre petit frère ? (coordonné de l'établissement, l'année scolaire, la classe et l'identité de l'élève ; les matières, les moyennes, les appréciations ; la moyenne finale)
2. Sur forme d'une feuille de calcul, dessiner la structure du bulletin de votre frère. (6 matière)

LECON 8 : PRODUIRE UNE FACTURE , UN BULLETIN DE NOTE OU DE PAIE

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								

3. Donner les formules à l'aide des référence des cellules pour :
 - 3.1. La somme des coefficients(=SOMME (E6 :E11))
 - 3.2. La note total (note * coef) (=D6*E6)
 - 3.3. La somme des notes total (=SOMME (F6 : F11))
 - 3.4. La moyenne (=F12/E12)
4. Donner la procédure pour imprimer le bulletin de note (Cliquer sur fichier, Cliquer sur imprimer, Modifier ou non le nombre de copies, Cliquer sur ok

Exercice 3 : voici le bulletin de paie d'un employé de la station-service BOCOM

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

1. Lister les informations qu'on retrouve dessus.
2. Expliquer en quelque ligne la conception de ce bulletin de paie.

Module I : SYSTEMES INFORMATIQUES

PDF Compressor Free Version

Unité d'apprentissage 4 : CONFIGURATION D'UN RESEAU INFORMATIQUE

Situation problème

Stéphane a rencontré son ami Paul et lui raconte que son papa qui est informaticien est sorti de la maison en disant que les employés de la société se plaignent qu'ils n'arrivent pas à communiquer sur le LAN et se demande bien de quoi il s'agit. Paul lui dit qu'il s'agit d'un type de réseau informatique.

1. Qu'est-ce qu'un réseau ? un réseau informatique ?
2. Quels sont les différents types de réseaux ?

UE 9: TYPES DE RESEAUX ET AVANTAGES DES RESEAUX SANS FILS

Compétences visées :

- Distinguer les différents types de réseaux informatiques (selon l'étendue géographique, selon le type de la liaison, selon l'utilisation) ;
- Citer les avantages des réseaux sans fils.

Trace écrite :

Introduction

Un réseau informatique est un ensemble d'équipements informatiques interconnectés dans le but d'échanger les informations. On distingue deux types de réseaux informatiques : les réseaux filaires et les réseaux sans fil. Il est important de noter que les réseaux sans fils ont plus d'avantages que les réseaux filaires.

I. Généralités sur les réseaux informatiques

Donner définition des réseaux informatiques : ensemble d'équipements interconnectés entre eux dans le but de partager les informations et communiquer.

Donner les avantages des réseaux informatiques :

- Partage et transmission des ressources ;
- Diminution des coûts ;
- Organisation et communication efficace des données ;
- Unicité de l'information.

II. Types de réseaux informatiques

Les réseaux informatiques sont classifiés selon les critères suivants :

- En fonction du type de la liaison
 - Réseaux sans fils

Les équipements sont reliés à l'aide des ondes (infrarouge, bluetooth, wifi, etc...)

- Réseaux filaires

Les équipements sont reliés à l'aide des câbles (câbles à paires torsadées, câbles coaxial, fibre optique).

- En fonction de l'étendue géographique
 - PAN (personal area network) /WPAN (wireless personal area network): il est utilisé pour relier les équipements électroniques entre eux.
La distance entre les équipements est de quelques mètres (10m).
Les appareils mis en réseau doivent être dans la même pièce. Il existe plusieurs technologies utilisées pour les WPAN: Le bluetooth, le HomeRF, l'infrarouge
Exemple : trois ordinateurs connectés entre eux, deux Smartphones connectés entre eux à l'aide du Bluetooth
 - LAN (local area network)/WLAN (wireless local area network) : on l'utilise pour relier des équipements relativement proches.
La distance entre les équipements est d'une centaine de mètres maximum (max 1km). Les appareils mis en réseau sont dans un immeuble. Il existe plusieurs technologies utilisées pour le WLAN : le WIFI (Wireless Fidelity), HiperLAN (High Performance Radio LAN)
Exemple : réseau des ordinateurs d'un lycée, des équipements reliés entre eux à l'aide du wifi
 - MAN (metropolitan area network)/WMAN (wireless metropolitan area network) :
La distance entre les équipements est d'une dizaine de kilomètres maximum. Le réseau métropolitain sans fil est connu sous le nom de Boucle Locale Radio (BLR) et offre un débit utile de 1 à 10 Mbit/s pour une portée de 4 à 10 kilomètres. Les technologies utilisées sont : le WIMAX (Worldwide Interoperability for Microwave Access), LMDS (Local Multipoint, Distribution Service)
Exemple : interconnexion des lycées dans la ville de Yaoundé.
 - WAN (wide area network)/WWAN (Wireless wide area network) :
La distance entre les équipements d'une centaine de km. Le réseau étendu sans fil est également connu sous le nom de réseau cellulaire mobile. Il s'agit des réseaux sans fil les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil. Les principales technologies sont les suivantes : GSM (2G) (Global System for Mobile communication): 9,05 Kbps GPRS (2.5G) (General Packet Radio Service): 171,2 Kbps UMTS (3G) (Universal Mobile Telecommunication System) : 1.9 Mbps LTE-Advanced (4G) : 1Gbps / 100Mbps

PDF Compressor Free Version

- En fonction de l'utilisation

- Internet : interconnexion de tous les réseaux
- Intranet : interconnexion des machines dans un réseau local
- Extranet : connexion à un réseau local étant extérieur à la structure ; réseau externe d'une entité organisationnelle)

III. Avantages des réseaux sans fils

Mobilité : les déplacements sont facilités

Facilité : dès que l'on est dans une zone couverte par le réseau et que l'on dispose des autorisations nécessaires, la connexion est instantanée

Souplesse : l'adaptation de la zone d'action en fonction des besoins

Faible cout : le cout de la main d'œuvre pour la mise en place est réduit

Evolutivité : il est facile d'étendre ou de réduire le réseau

Conclusion

Les réseaux informatiques facilitent la communication entre les individus quelque soit l'endroit où ils se trouvent. Ainsi nous avons les réseaux filaires et les réseaux sans. Nous pouvons constater que les réseaux sans fil sont plus avantageux que les réseaux filaires

Jeu bilingue

FRANÇAIS	ANGLAIS
Réseau informatique	Network
Réseaux sans fil	Wireless network
Internet	Internet
Intranet	Intranet
Extranet	Extranet

Exercices

Exercice 1

1) Définir réseau informatique

Réponse : un réseau informatique est un ensemble d'équipements interconnectés dans le but d'échanger les informations

2) Quels sont les 2 types de réseaux informatiques ?

Réponse : il existe les réseaux filaires et les réseaux sans fil

3) Quels sont les media utilisés dans chaque type de réseau ?

Réponse : les média utilisés sont : les câbles, le Bluetooth, l'infrarouge, les ondes radio, le satellite

4) Donner la différence entre l'intranet et l'extranet ?

Réponse : la différence entre l'intranet et l'extranet est que : l'intranet est un réseau propre à une entreprise et inaccessible aux utilisateurs extérieurs. Alors que l'extranet est un réseau d'une entreprise qui peut être accessible aux personnes extérieures après authentification

5) Enumérez 2 différences entre le réseau filaire et le réseau sans fil

Réponse : la mobilité du réseau sans fil par rapport au réseau filaire. La simplicité de l'installation d'un réseau sans fil

Exercice 2

Le proviseur de votre établissement voudrait connecter les trois ordinateurs de son service.

1) Quel type de réseau sera mis en place ?

Réponse : Le réseau local

2) Pour un échange facile d'informations avec le lycée voisin, le proviseur voudrait savoir quel type de réseau utilisé ?

Réponse : le réseau métropolitain

Exercice 3 :

M. Gauthier possède 5 ordinateurs dans son entreprise et il voudrait que les différentes machines puissent communiquer entre elles sans utiliser une connexion internet. Les locaux de son entreprise se trouvent dans un immeuble.

1. Est-il possible de communiquer sans utiliser Internet ? Si oui que doit-il mettre en œuvre ?
2. Quel est le type de réseau à mettre en place en tenant compte de l'étendue géographique ?
3. en tenant compte de l'utilisation, quel type de réseau sera mis en place ?

Exercice 4 :

La société ARPA Plus possède un réseau informatique de type LAN d'environ 15 machines. Le directeur général se déplace régulièrement et souhaite continuer à envoyer des messages dans le réseau de l'entreprise lorsqu'il se trouve à l'étranger.

Est-il possible que le DG puisse communiquer sur le réseau de l'entreprise ? Si oui que faut-il faire ?

PDF Compressor Free Version

Exercice 5 :

Nael en regardant dans la galerie du téléphone de son ami Yanis a retrouvé de vieilles photos d'eux et souhaite que Yanis lui envoie ces photos en se servant de l'application Xender.

1. Quel est le type de réseau utilisé en tenant compte de la nature de la liaison ?
2. Citer deux avantages de ce type de réseau.

MODULE N° 1 : ENVIRONNEMENT INFORMATIQUE ET HUMANITE NUMERIQUE

PDF Compressor Free Version

UA 4 : CONFIGURATION D'UN RESEAU INFORMATIQUE

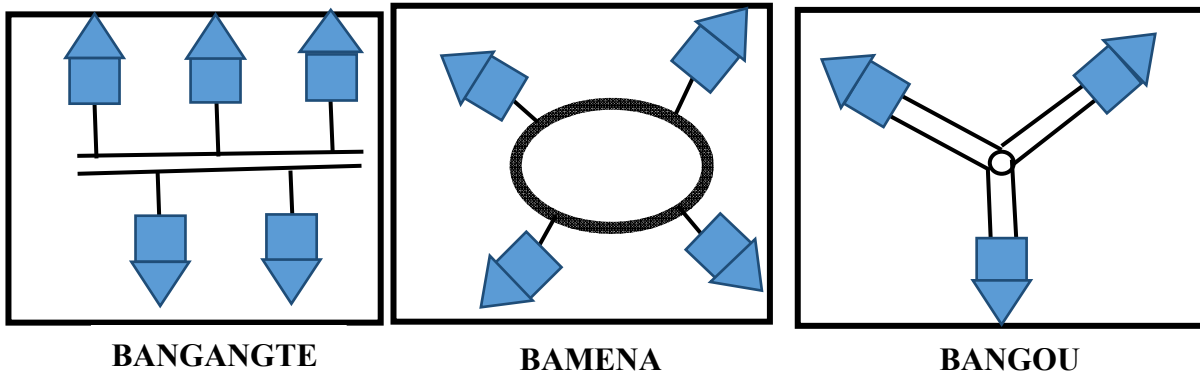
UE10 : TOPOLOGIES ET ARCHITECTURES RESEAUX

COMPETENCES VISEES : à la fin de cette leçon, l'apprenant devra être capable de :

- ❖ Identifier, nommer et donner le rôle des équipements réseaux
- ❖ Décrire les principaux types de câble

SITUATION PROBLEME

NGANTCHEU est un commerçant qui se balade de villages en villages pour proposer ses produits aux différents habitants. Il choisit de vendre dans trois villages : BANGANGTE, BAMENA et BANGOU, donc les dispositions des maisons sont les suivantes :



1. Par quel canal joint-il les habitants de chaque maison ?
Le canal utilisé est la route (voie secondaires, pistes).
2. En remplaçant les maisons par les ordinateurs, par quel nom générique peut-on désigner la disposition physique des ordinateurs dans chaque figure ?
Le nom générique est topologie.
3. Comment est régie la communication entre les différents ordinateurs ?
La communication est régie par les différents câbles réseaux qui interconnectent les ordinateurs entre eux.
4. Comment appelle-t-on la manière dont les informations circulent dans ces machines ?
La manière dont les informations circulent entre les machines est appelé topologie logique.
5. Définir topologie physique et architecture réseau.
Topologie réseau physique : Elle désigne le mode d'interconnexion physique des différents éléments du réseau. Architecture réseau : elle définit le rôle que jouent les ordinateurs dans un réseau.
6. Citer deux topologies et deux architectures de votre choix. *Voir cours*

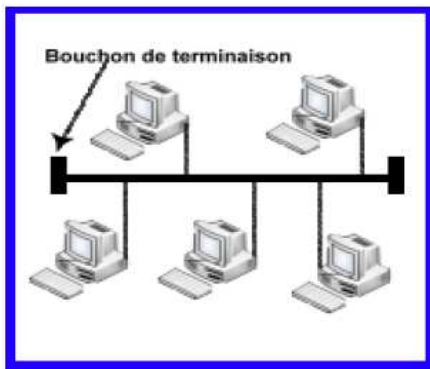
I. LES TOPOLOGIES RESEAUX

A. Les topologies Physiques

Elle désigne le mode d'interconnexion physique des différents éléments du réseau (structure des chemins de câbles, type de raccordement...). C'est tout simplement la disposition physique des équipements dans un réseau. On distingue diverses topologies physiques à savoir :

❖ **La topologie en Bus**

Dans cette topologie, les ordinateurs sont disposés et reliés de part et d'autre d'un câble principal appelé **bus(cable)**. Le support de transmission utilisé dans ce cas est le câble coaxial (**coaxial cable**). Lorsqu'un ordinateur envoie un message, chaque ordinateur situé sur le chemin intermédiaire reçoit le message et l'ouvre s'il est le destinataire. La **figure 3** illustre cette topologie.



fréquentes

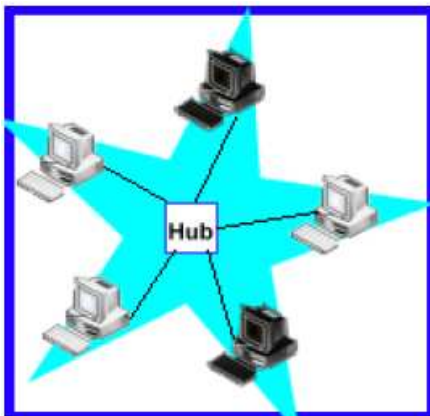
Avantages :

- Tous les ordinateurs sont connectés entre eux et peuvent donc communiquer directement
- Facile à mettre en œuvre et possède un fonctionnement simple

Inconvénients :

- Les ordinateurs sont déconnectés s'il se produit une rupture de câble
- Les collisions (rencontre entre deux informations) sont fréquentes
- Les performances se dégradent avec l'ajout de nouvelles machines

❖ **Topologie en Etoile**



pas le réseau

Dans cette topologie, les ordinateurs sont tous reliés à un équipement central. Ce dernier joue le rôle d'intermédiaire, car tous les messages passent d'abord par lui. C'est lui qui dirige les messages à destination. Le support de transmission dans ce cas est le câble à paires torsadées (**twisted paire cable**). L'équipement central peut être un concentrateur (**hub**) ou un commutateur (**switch**). Dans la **figure 2** suivante, c'est le **hub**.

Avantages :

- Tous les ordinateurs communiquent entre eux aisément
- Un ordinateur peut tomber en panne sans que cela ne paralyse pas le réseau
- L'ajout facile de nouveaux ordinateurs

Inconvénients :

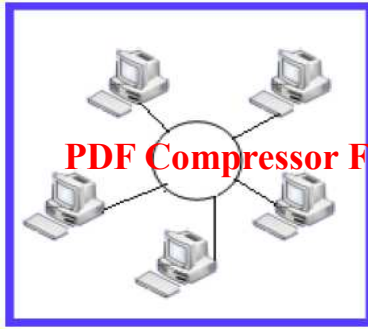
- Plus coûteux qu'un réseau en bus avec l'achat d'un nouveau matériel
- La panne du nœud central entraîne la destruction du réseau

❖ **Topologie en Anneau**

Dans cette topologie, les ordinateurs sont connectés sur un câble en forme de boucle ou anneau. C'est cet anneau qui achemine les messages suivant un sens bien défini. Le support de transmission dans ce cas est le câble coaxial. Un ordinateur n'accepte l'information que si elle lui est destinée. Dans le cas contraire, l'ordinateur en question fait passer l'information à l'ordinateur suivant. La figure 3 ci-dessous illustre cette topologie.

Avantages :

- Nombre de câble réduit



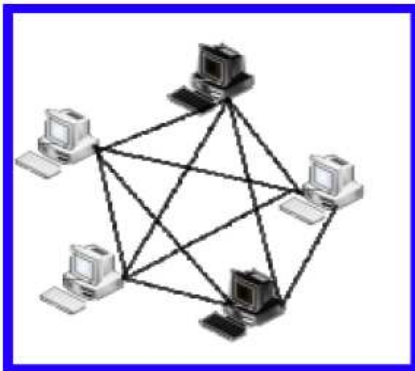
- La simplicité du protocole de communication
- L'absence de collision

Inconvénients :

- La panne d'un ordinateur paralyse le réseau
 - Difficile à mettre en œuvre
- L'ajout ou la suppression d'un ordinateur peut perturber le fonctionnement de l'ensemble du réseau

❖ **Topologie Maillée**

Dans cette topologie, chaque ordinateur est relié aux autres par un câble. Lorsqu'un ordinateur veut envoyer un message, ce dernier transite directement à travers le câble qui mène au destinataire, comme à la figure 4 ci-dessous.



Avantages :

- L'ajout facile de nouveaux ordinateurs
- La panne d'un ordinateur ou d'un câble n'influence pas le réseau

Inconvénients :

- Très coûteux avec l'achat de nombreux câbles trop élevés
- Difficile à mettre en œuvre.

On peut ajouter à cette liste la topologie en arbre. Une topologie arborescente est une combinaison des différentes autres topologies ; elle peut reposer à la fois sur des topologies en bus, en étoile et en anneau.

B. LES TOPOLOGIES LOGIQUES

Une fois la topologie physique construite, il est nécessaire de définir le procédé qu'utilisent les ordinateurs pour transmettre les informations sur le support de transmission ainsi que la manière avec laquelle ces informations vont circuler sur ce support. Pour ce faire, il existe plusieurs topologies logiques :

▪ **Ethernet**

Lorsqu'un ordinateur veut émettre un message, il vérifie d'abord qu'aucun message ne circule sur le réseau. Si c'est le cas, il envoie le message sur le support. Dans le cas contraire, c'est un autre ordinateur qui est en train d'émettre : on dit qu'il y a **collision**. Cette topologie logique est utilisée dans les topologies physiques en bus et en étoile.

▪ **FDDI**

La topologie logique FDDI (Fiber Distributed Data Interface) est une topologie qui utilise la fibre optique comme support de transmission pour transmettre les informations. Elle est basée sur la topologie physique en anneau. Pour émettre un message sur l'anneau, chaque ordinateur a besoin d'un jeton qui donne le droit d'envoyer les données sur le support. Le jeton fait le tour de l'anneau. Lorsqu'un ordinateur le reçoit, il le garde jusqu'à ce qu'il ait fini de transmettre son message. S'il n'a pas de message à transmettre, il laisse passer le jeton qui continue son bout de chemin sur l'anneau.

▪ **Token ring**

Token Ring (en français anneau à jeton) est une topologie logique qui fonctionne également sur la base d'un jeton pour l'accès au support. Elle est utilisée dans la topologie physique en étoile pour l'envoi des messages.

II. LES ARCHITECTURES RESEAUX

Leçon 10 : TOPOLOGIES ET ARCHITECTURES RESEAUX

Une architecture réseau définit le rôle que jouent les ordinateurs dans le réseau. Chaque ordinateur peut jouer deux rôles :

- **Client**: c'est l'ordinateur qui initie les requêtes puis attend les réponses du serveur
- **Serveur**: c'est l'ordinateur qui est à l'écoute, en attente des requêtes des clients. Dès qu'une requête arrive, il traite et renvoie la réponse au client.

Il existe plusieurs architectures réseaux : les architectures d'égal à égal, infrastructure et client-serveur.

❖ L'architecture d'égal à égal

L'architecture d'égal à égal (en anglais Peer-to-Peer) est un type d'architecture réseau dans lequel chaque ordinateur est à la fois client et serveur. Autrement dit, chaque ordinateur possède des ressources qu'il met à la disposition des autres. Dans le langage des réseaux sans fil, cette architecture est appelée **ad hoc**. Cette architecture a pour inconvénients :

- Une sécurité faible
- Une architecture non centralisée rendant difficile l'administration

❖ L'architecture client /serveur

L'architecture **client-serveur** est un mode de communication dans lequel on distingue un serveur parmi plusieurs clients. Chaque logiciel client peut formuler des requêtes à destination du serveur.

Les avantages de cette architecture sont :

- La centralisation des ressources au niveau du serveur ;
- Une meilleure sécurité ;
- Le caractère évolutif du réseau : l'ajout ou la suppression des clients n'affecte pas le fonctionnement du réseau.

EXERCICES

EXERCICES IVRAI OU FAUX

Dire si les affirmations suivantes sont vrai ou fausses.

1. HUB est uniquement utilisé dans la topologie réseau anneau
2. Lorsqu'un switch reçoit une information, il le redistribue à tous les ordinateurs du réseau
3. Dans l'architecture post a post, les ressources sont centralisées
4. Dans la topologie FDDI, le support transmission est la fibre optique.
5. La défaillance de l'équipement centrale dans une topologie en étoile paralyse le réseau

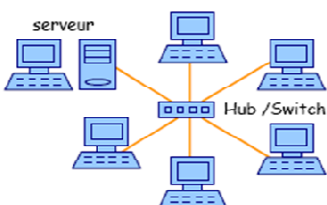
EXERCICE 2 ENIGMES

Résolvez les énigmes suivantes.

1. Je suis un équipement réseau utilisé pour échanger les informations sur internet, mon nom est formé par cinq lettres qui commence par **M** et se termine par **M**.
2. Je suis une disposition physique d'ordinateur dans laquelle chacun communique directement avec son destinataire.
3. Je suis un mode de communication réseau caractérisée par une forte sécurité, je centralise les ressources et je n'ai de problèmes selon qu'on m'ajoute ou retire un communicant(ordinateur).
4. Je suis un équipement réseau d'interconnexion, je suis caractérisé par ma forte confidentialité. Quand je suis dans un réseau, toutes les informations passent par moi. Quand je reçois une information, je l'envoie directement au destinataire.

EXERCICE 3

Dans le but de partager vos tâches de travaux pratiques avec votre enseignant, les ordinateurs de votre salle informatiques sont interconnectés entre eux suivant la disposition donnée par le schéma ci-dessous.



1. Définir topologie physique
2. Donner la topologie correspondante à ce réseau.
3. Donner un avantage et un inconvénient de ce réseau.
4. Quelle architecture réseau est mise en exergue ici ?
5. que choisirez-vous comme équipement central ? pourquoi ?

PDF Compressor Free Version
CORRIGES DES EXERCICES

Résolution exercice 1.

- | | | |
|----------|----------|---------|
| 1. Faux. | 3. Faux. | |
| 2. Faux. | 4. Vrai | 5. Vrai |

Résolution exercice 2

1. Modem
2. Topologie maillée
3. Architecture client-serveur.
4. Le switch.

Résolution exercice 3.

1. **Topologie réseau physique** :Elle désigne le mode d'interconnexion physique des différents éléments du réseau
2. Topologie en étoile
3. Son avantage est que la défaillance d'un ordinateur ne paralyse pas le réseau et son inconvénient est que la défaillance de l'équipement central paralyse le réseau.
4. Architecture client-serveur
5. Le switch, car il garantit la confidentialité des informations dans le réseau.

Module1 : SYSTEMES INFORMATIQUES

PDF Compressor Free Version

UA4 : CONFIGURATION D'UN RESEAU INFORMATIQUE

UE11 : EQUIPEMENTS ET PROTOCOLES RESEAUX

Compétences visées :

- Donner le rôle des équipements réseaux suivants : modem, hub, switch, routeur ;
- Donner le rôle de chacun des protocoles d'internet suivants : http, IP, FTP, SMTP, POP3, DHCP.

Contrôle de prérequis :

Situation problème :

Cinq élèves équipés chacun d'un ordinateur fixe se retrouvent chez les grands parents pour y passer les vacances. Pour s'échanger les informations, ils se baladent de chambre en chambre avec des clés USB. Etant entendu que vous faites les cours d'informatique

- 1) Prodiguer de façon à limiter voir même annuler ces déplacements?

REPONSE attendue : Mettre les différents ordinateurs individuels en réseau

- 2) Nommer les éléments matériels permettant de relier ces ordinateurs.

REPONSE attendue : chaque ordinateur devra Posséder la carte réseau à l'achat et si tel n'est le cas ils doivent en acheter installer et il devrait ajouter les câbles réseaux et les connecteurs comme RJ45

- 3) Une fois les ordinateurs en réseau, ils décident de connecter leur réseau formés à celui du lycée qu'auront-ils besoin pour la circonstance ?

REPONSE attendue : un équipement d'interconnexion comme le **modem**

- 4) pour communiquer les ordinateurs ont besoin de respecter un ensemble de règles et de procédures. donner un nom à ces ensembles de règles et procédures.

REPONSE attendue : le protocole

- 5) énumérer quelques exemples de protocoles que vous connaissez

REPONSE attendue : HTTP, FTP, DHCP etc...

INTRODUCTION

PDF Compressor Free Version

On entend par **équipements réseaux** l'ensemble de matériels d'un réseau qui sont interconnectés ou qui servent à l'interconnexion. Il peut s'agir soit des hôtes (ordinateurs, imprimante réseau, serveur, camera...) soit des équipements d'interconnexion. Un hôte a besoin au moins d'une carte réseau, des câbles et des connecteurs.

I. SUPPORTS PHYSIQUES OU DE TRANSMISSION (Filaires)

Il s'agit ici des câbles et de leurs connecteurs. Ils permettent d'assurer la liaison entre les machines du réseau. Il peut s'agir :

1) La carte réseau ou NIC (Network Interface Card)

Tout équipement qui se connecte à un réseau utilise une carte réseau. Il s'agit d'une carte d'extension que l'on peut enficher sur la carte-mère et qui sert d'interface pour l'échange des données entre un hôte et le réseau local : c'est la porte d'entrée et de sortie dans un réseau.

2) Le câble réseau et connecteurs associés.

C'est le support de transmission permettant de relier les ordinateurs du réseau. Parmi ces câbles on distingue :

2-1) Le câble coaxial et le connecteur BNC

Le câble coaxial est le premier type de câble utilisé dans les débuts du réseau. C'est un fil rigide très souvent utilisé par les distributeurs de télévision par câble, il est généralement fait de métal ou d'aluminium et peut fournir des images TV et internet.

2-2) Le câble à paire torsadée et le connecteur RJ-45

La paire torsadée est le câble le plus utilisé dans les réseaux locaux comme votre salle informatique. Elle est, comme son nom l'indique, faite de 8 fils groupés par paire et entrecroisés (torsadés) afin de réduire les interférences. Les **connecteurs RJ45** sont fixés aux bouts des câbles et permettent de connecter ces derniers à la carte réseau à travers le port RJ45.

PDF Compressor Free Version

CONNECTEURS RJ45



Câble RJ45



Port RJ45 sur une carte réseau

2-3) La fibre optique

La fibre optique c'est un câble en verre ou en plastique. On parle de **fibre** à cause de son diamètre à peu près égal aux cheveux humains et **d'optique** parce qu'elle transporte les informations grâce à la lumière plutôt que par l'électricité comme le font la paire torsadée et le câble coaxial. Elle permet de transporter de très grandes quantités d'informations sur de longues distances en numérique et à des vitesses inégalables.

Remarque : Il existe aussi des supports de transmissions sans fils tels que : **Bluetooth, Infrarouge, WIFI, WIMAX etc....**

II. Les équipements d'interconnexion d'un réseau informatique.

Pour réaliser certains réseaux, seuls les équipements de base ne suffisent pas. L'on fait généralement recours aux équipements d'interconnexion. Ce sont :

- ✓ Le **concentrateur** : appelé **hub** en anglais est un équipement physique à plusieurs ports. Il sert à relier plusieurs ordinateurs entre eux. Son rôle c'est de prendre les données reçues sur un port et les diffuser sur l'ensemble des ports.
- ✓ Le **répéteur** : appelé **repeater** en anglais, est un équipement qui sert à régénérer le signal entre deux nœuds pour le but d'étendre la distance du réseau.
- ✓ Le **commutateur** : appelé **switch** en anglais, est un équipement multiport comme le concentrateur. Il sert à relier plusieurs équipements informatiques entre eux. Sa seule différence avec le hub, c'est sa capacité de connaître la machine à qui est destiné le message (donnée).
- ✓ La **passerelle ou Gateway** : permet de relier des réseaux qui utilisent des moyens de communication (protocole) et/ou des architectures différents.
- ✓ Le **modem** (modulateur-démodulateur): il permet de transformer les données numériques en données analogiques et inversement.
- ✓ Le **routeur** : c'est un matériel permettant l'interconnexion des sous-réseaux informatiques.
- ✓ Le **pont ou bridge** : Il permet de relier deux réseaux de même type (topologie physique).

- ✓ **Le pare-feu (firewall)** : c'est un système (matériel et logiciel) qui permet de protéger un réseau local d'intrusions de personnes en provenance d'internet. On l'appelle parfois garde-barrière, gate-keeper, coupe-feu.
- ✓ **Serveur Proxy ou mandataire** : c'est une machine dont le rôle principale est de permettre aux ordinateurs du LAN d'accéder à Internet. Il peut servir de firewall

III. Protocole réseau.

1. Notion de protocole

Entre deux ou plusieurs machines connectées qui s'échangent des informations il existe un ensemble de règles appelé **protocole**.

Un **protocole** désigne un ensemble de règles et procédures à respecter pour permettre à un ensemble d'équipements d'émettre et de recevoir des informations.

Un protocole est dit fiable quand il s'assure que les données qui ont été envoyées ont bien été reçues.

2. Exemples de protocoles.

Il existe une multitude de protocoles. Donc la liste ci-dessous est loin d'être exhaustive.

- ✓ **TCP/IP** (Transmission Control Protocol/Internet Protocol) permet l'interconnexion des machines hétérogènes ;
- ✓ **Appel Talk** est un protocole réseau spécifique et intrinsèque aux machines Appel ;
- ✓ **FTP** (File Transfert Protocol) est un protocole internet pour le transfert de fichiers ;
- ✓ **SMTP**(Simple Mail Transfert Protocol) est un protocole internet pour le transfert de messagerie électronique ou encore permet d'envoyer des courriers électroniques;
- ✓ **TCP** (Transmission control Protocol) est une partie du protocole internet TCP/IP qui assure que les connexions entre deux ordinateurs sont établies et maintenues ;
- ✓ **POP3** (Post Office Protocol version 3) permet de se connecter à un serveur de messagerie ;
- ✓ **IMPAP4** (Internet Message Advertising) permet de récupérer un courrier électronique ;
- ✓ **IP** (Internet Protocol) gère les adresses logiques des nœuds (stations...) ;

- ✓ **HTTP** (Hyper Text Transfert Protocol) protocole de communication permettant un **PDF Compressor Free Version** transfert de fichiers (essentiellement au format HTML) localisés grâce à une chaîne de caractères appelée URL entre client (navigateur) et serveur web ;
- ✓ **DHCP** (Dynamic Host Configuration Protocol) permet d'assurer la configuration automatique des paramètres IP d'une station ou d'une machine, notamment en lui en lui attribuant automatiquement une adresse IP et un masque sous réseau.

Conclusion

Communiquer dans un réseau nécessite un ensemble de règles et de procédure que l'on doit respecter ceci à travers un certain nombre d'équipements de base d'une part et d'équipements d'interconnexion d'autre part.

Jeu bilingue :

Mot en français	Traduction en Anglais
Réseau	Network
commutateur	switch
concentrateur	hub
protocole	protocol
passerelle	Gateway
pont	bridge
répéteur	repeater
câble	cable
Fibre optique	Optical fiber

Exercice 1

Mr Herman Raoul possède une bureautique pour faciliter le travail veut créer un réseau de 07 ordinateurs et une imprimante dont la topologie est en étoile. Ne pouvant pas seul réussir cette tâche, il fait donc appel à vous dans le but de leur aider en répondant aux questions suivantes :

- 1) Donner deux éléments d'interconnexion que Mr Herman Raoul peut utiliser pour relier les ordinateurs de son réseau.
- 2) Présenter la différence entre les éléments cités à la question 1.
- 3) Quel est le support de transmission qui convient le mieux pour ce réseau local ?

4. Mr Herman Raoul souhaitent joindre son réseaux avec celui du Collège situé à quelque dizaine de mètre de sa bureautique pour former un seul. Quels équipements d'interconnexion pourrait – il utiliser pour résoudre ce problème ?

Correction

- 1) Deux éléments d'interconnexions : le Switch et le Hub
- 2) Différence entre le Hub et le Switch : le Switch c'est le Hub intelligent.
- 3) Pour ce réseau le support qui convient le mieux est le Switch
- 4) Il pourra utiliser le Pont pour reliés les deux réseaux

Exercice 2:

Votre frère TAGNE est un technicien qui travaille sur les réseaux informatiques. De passage chez-lui, tu aperçois une facture sur laquelle figurent les appareils suivants :

- ✓ 1 serveur
- ✓ 5 cartes réseau
- ✓ 5 modems
- ✓ 1 routeur
- ✓ 1 hub
- ✓ 1 switch

Ta maman qui ne sait rien en informatique mais qui veut savoir un peu plus sur les réseaux, te demande de lui expliquer ce qu'il y a dans cette liste.

- 1- Définir chacun de ces matériels et dire à quoi ils servent.
- 2- Ta maman suite aux explications que vous lui aviez faites plus haut veut faire communiquer plusieurs machines. Mais son fils TAGNE lui fait comprendre qu'il ya un ensemble de règles et de procédures à respecter.
 - a) Proposer le nom correspondant à un ensemble de règles et procédures utilisées par les ordinateurs pour les échanges d'information dans un réseau.
 - b) Nommer le protocole permettant de s'envoyer des messages
 - c) En outre du protocole cité ci-dessus, énumérer quatre autres protocoles tout en donnant leur rôle.

Module1 : SYSTEMES INFORMATIQUES**PDF Compressor Free Version****UA4 : CONFIGURATION D'UN RESEAU INFORMATIQUE****UE12 : LES ADRESSES IP (IPV4) ET IPV6****Compétences visées :**

- ✓ Lister les deux versions d'adresse IP
- ✓ Décrire les adresses IPV4;
- ✓ Lister les classes d'adresses IPv4 ;
- ✓ Identifier une adresse IPV4 de classe A, B et C ;
- ✓ Donner le principal avantage de IPV6 sur IPV4

SITUATION PROBLEME :

Vous petit frère a créé un réseau informatique de 04 ordinateurs. Mais à grande surprise, il constate que ce réseau ne fonctionne pas ceci parce qu'il n'a pas été configuré. Ne pouvant pas tout seul résoudre ce problème, il fait appel à vous dans le but de l'aider.

CONSIGNE :

1. Quels sont les paramètres de configuration d'un ordinateur sur le réseau informatique ? (**Réponse attendue:** adresse IP et Mac, le protocole, domaine)
2. Définir adresse IP (**Réponse attendue:** nombre qui permet d'identifier de manière unique un ordinateur du réseau)
3. Quel sont les types d'adressage d'un réseau informatique? (**Réponse attendue :** adressage dynamique et statique)
4. En quoi consiste chacun de ce type d'adressage? (**Réponse attendue:** adressage dynamique est l'attribution automatique des adresses IP aux équipements du réseau et l'adressage statique consiste à attribuer manuellement les adresses aux équipements réseau)

RESUME

Pour pouvoir communiquer sur un réseau, chaque ordinateur doit avoir une adresse IP unique. L'adresse IP, encore appelée adresse logique, permet d'identifier le réseau et peut changer pour une même machine tandis que L'adresse physique ou MAC identifie un hôte dans un réseau, elle est unique et attribuée par le fabricant de la carte réseau. Deux machines

ne peuvent donc avoir la même adresse MAC. Exemple d'adresse MAC : 70-5A-B6-C4-A2-94

1. Notion d'adresse IPV6

Il existe les adresses IP de version 4 (IPV4) sur 32 bits et les adresses IP de version 6 (IPV6) sur 128 bits la version 4 est actuellement la plus utilisée.

IPV6 est un protocole réseau sans connexion de la couche 3 du modèle OSI¹. Contrairement à l'IPV4, dont l'espace d'adressage ne faisait que 32 bits, celui de l'IPV6 s'élève à 128 bits. Comme pour dire le nombre total d'adresse IPV6 est de 2^{128} correspondant à $3,4 \times 10^{38}$ adresse IP unique.

Comme principale avantages d'adresse IPV6, c'est le nombre quasi infini d'adresse IP globale.

2. Le découpage d'adresse

Une adresse IP se décompose aussi en deux parties. Une partie servant pour l'identification du réseau (Net Id) et une autre servant pour l'identification de l'adresse de la machine (Host Id). Il existe cinq classe d'adresse IP, chaque classe étant identifiée d'une lettre allant de A à E. pour identifier à quelle appartient une adresse IP, il faut examiner ses premier bit. Comme illustrer dans le tableau ci-dessus.

Classes	Premier bits	Premier octet	Début	Fin	Masque sous réseau
A	0	0 – 126	0.0.0.0	126.255.255.255	255.0.0.0
B	10	128 – 191	128.0.0.0	191.255.255.255	255.255.0.0
C	110	192 – 223	192.0.0.0	223.255.255.255	255.255.255.0
D	1110	224 – 239	224.0.0.0	126.255.255.255	Non défini
E	1111	240 – 255	240.0.0.0	240.255.255.255	Non défini

N.B : Les adresses IP commençant par 127 sont réservées à des procédures de tests (127.0.0.1=localhost)

Tableau classification des adresses IP

Classes	Champ 1	Champ 2	Champ 3	Champ 4	Maximum de réseau	Maximum d'hôtes
Classe A	Identificateur réseau	Identificateur d'hôte			126	16777214
Classe B	Identificateur réseau		Identificateur d'hôte		16382	65534
Classe C	Identificateur réseau			Identificateur d'hôte	2097150	254

¹ Les couches du modèle OSI : ("couches matérielle" 1) **Physique** « transmission binaire numérique ou analogique » 2) **Liaison** « adressage physique » 3) **Réseau** « détermine le parcours et l'adressage logique ip » "couches hautes" 4) **Transport** « connexion de bout en bout et contrôle de flux 'TCP' » 5) **Session** « communication interhost », 6) **Présentation** « conversion et chiffrement des données » 7) **Application** « point d'accès aux services réseau »)

PDF Compressor Free Version

Classification des identificateurs des adresses par classe

3. Masque de réseau

Le masque de réseau sert à séparer les parties réseau et hôte d'une adresse. Il est obtenu en mettant à 1 tous les bits de la partie réseau de l'adresse IP. On retrouve l'adresse du réseau en effectuant un ET logique bit à bit entre une adresse complète et le masque de réseau en respectant la table suivante (0 et 1=0 ; 1 et 0= 0 ; 0 et 0 =0 ; 1 et 1 =1). En général, le masque se présente sous la forme W.X.Y.Z. Il a pour rôle de vérifier si une adresse de machine appartient ou pas à un réseau.

4. L'adresse de diffusion

Chaque réseau possède une adresse particulière dite de diffusion. Tous les hôtes du réseau «écoutent» cette adresse en plus de la leur. Certaines informations telles que les annonces de service ou les messages d'alerte sont utiles à l'ensemble des hôtes du réseau.

Prenons un exemple d'adresse IP pour en identifier les différentes parties :

Exemple : adresse IP 192.168.1.1

Adresse complète 192.168. 1. 1

Masque de réseau 255.255.255.0

Classe d'adresse : Classe C

Partie réseau(Net Id) :192.168. 1.

Partie hôte (HostID) : 1

Adresse Réseau 192.168. 1. 0

Adresse de diffusion 192.168. 1.255

5. Les différents modes d'adressage

La configuration des adresses dans un réseau consiste à attribuer les adresses IPV4 et un masque de sous réseaux. Elle peut se faire de façon manuelle (adressage statique) ou dynamique (adressage automatique).

✓ **L'adressage statique** : consiste à définir manuellement une adresse IP statique ou fixe à chaque ordinateur. Chaque fois qu'un ordinateur se connecte au réseau, il aura la même adresse IP. Lorsque deux ordinateurs ont la même adresse IP, on parle de conflit d'adresse ;

✓ **L'adressage dynamique** : elle se fait à l'aide du protocole **DCHP (Dynamic Host Configuration Protocol)**. Le protocole DHCP permet l'attribution automatique des

informations d'adressage, telles que l'adresse IP, le masque de sous-réseau et d'autres paramètres.

JEU BILINGUE :

Mots en français	Traduction en anglais
Diffusion	Broadcast
Adresse réseau	Network adress
Masque de réseau	Network mask
Sous – réseau	Subnet

EXERCICES CORRIGES

EXERCICE 1

- Classer les adresses suivantes par classes :
 - 143.25.67.89
 - 12.15.5.45
 - 192.23.67.123
 - 221.45.67.123
 - 172.12.56.78
 - 123.56.78.23
 - 126.9.76.23
- Parmi les adresses suivantes une seule est erronée laquelle ? Justifier votre réponse
 - 126.0.0.213
 - 126.0.0.099
 - 126.0.0.317

EXERCICE 2

- 204.160.241.93 (adresse IP de www.javasoft.com);
- 18.181.0.31 (www.mit.edu)
- 138.96.32.3 (www.inria.fr);
- 226.192.60.40

EXERCICE 3

Soit les adresse IP ci-dessus :

- 10.21.125.32 ;
- 155.0.0.78 ;
- 192.168.25.69 ;

Pour ces adresses sus – citées, donnez :

- Leurs classes ;
- L'ID réseau et l'ID de l'hôte ;
- Dites si ces adresses sont privées ou publiques ;
- Leurs traductions binaires ;
- L'adresse réseau et l'adresse machine.

EXERCICE 4

Pour l'ensemble des adresses IP des machines suivantes présentes sur le même réseau physique, et ayant un masque par défaut, donner les adresses des machines qui communiqueront ensemble. Justifiez vos réponses.

- a) 10.0.0.1 g) 10.2.3.6 m) 10.254.254.254
 b) 122.0.25.38 h) 11.23.69.87 n) 1.1.1.1
 c) 122.25.25.39 i) 200.25.89.56 o) 156.0.54.2
 d) 200.25.48.69 j) 156.25.12.36 p) 201.25.48.1
 e) 156.54.23.1 k) 156.54.69.2
 f) 156.25.69.12 l) 200.25.48.12

CORRECTION

EXERCICE 1

1. Classifications les adresses suivantes par classes :

Classe A	Classe B	Classe C
c) 12.15.5.45	a) 143.25.67.89	d) 192.23.67.123
f) 123.56.78.23	b) 172.12.56.78	e) 221.45.67.123
g) 126.9.76.23		

2. L'adresse erronée est c) 126.0.0.317 parce que le nombre 317 n'est pas compris entre 0 et 255

EXERCICE 2

Classons les adresses réseaux :

- a- 204.160.241.93 (adresse IP de www.javasoft.com): **Classe C** ; d- 226.192.60.40 :
Classe C
 b- 138.96.32.3 (www.inria.fr): **classe B** c- 18.181.0.31 (www.mit.edu): **Classe A**

EXERCICE 3

Soit les adresse IP ci-dessus :

- b) **10.21.125.32** ;
 ➤ Classe **A** ; L'ID réseau : **10**, et l'ID de l'hôte : **21.125.32** ; Adresses publiques ;
 ➤ Traductions binaires² : **00001010.00010101.01111101.00100000**
 ➤ L'adresse réseau est : **10.0.0.0** et l'adresse machine³ est **357896**. En effet, on aura
 $2^5 + 2^8 + 2^{10} + 2^{11} + 2^{12} + 2^{13} + 2^{14} + 2^{16} + 2^{18} + 2^{20} = 357896$.
 c) **155.0.0.78** ;
 ➤ Classe **B** ; l'ID réseau : **155.0** et l'ID de l'hôte : **0.78** ; Adresses publiques ;

²Pour la traduire une adresse IP en binaire, il suffit de convertir chaque partie de l'adresse IP en base 2, et on l'écrit sur 8 bits en ajoutant les zéros (0) à droite du bit quand il n'atteint pas.

³ Pour trouver l'adresse machine il suffira de prendre la traduction binaire de l'adresse hôte en un bloc et le convertir en base 10.

➤ Leurs traductions binaires : **10011011.00000000.00000000.01001110** ;

PDF Compressor Free Version

➤ L'adresse réseau est **155.0.0.0** et l'adresse machine est **78** ($2^1 + 2^2 + 2^3 + 2^6 = 78$.)

d) 192.168.25.69 ;

Pour ces adresses sus – citées, donnez :

➤ Classe **C** ; l'ID réseau est **192.168.25** et l'ID de l'hôte est **69** ; Adresse publiques ;

➤ Leurs traductions binaires : **11000000.10101000.00011001.01000101**

➤ L'adresse réseau est : **192.168.25.0** et l'adresse machine est : **69** ($2^0 + 2^2 + 2^6 = 69$.)

EXERCICE 12

Les machines qui communiqueront ensemble sont :

A ; G ; M	B ; C	E ; F ; J ; K ; O	D ; I ; L
-----------	-------	-------------------	-----------

➤ Justification : les ordinateurs communiquent directement lorsqu'ils appartiennent au même réseau c'est ainsi qu'en application le masque par défaut que l'on retrouve aux différentes classes de réseaux auquel ils appartiennent.

Module1 : Systèmes Informatiques
PDF Compressor Free Version

UA4 Configuration d'un réseau informatique

UE 13 : MODES D'ADRESSAGES ET COMMANDES RÉSEAUX

Compétences visées :

- ✓ Définir adresse IP
- ✓ Décrire les modes d'adressage
- ✓ Utiliser quelques commandes réseau (ping, ipconfig)

Contrôle de prérequis :

- ✓ Définir réseau informatique
- ✓ Donner le rôle de la carte réseau
- ✓ Donner le rôle du protocole DHCP

Situation Problème

L'ordinateur de la secrétaire de Proviseur de votre établissement est connecté au réseau local du lycée mais les autres utilisateurs de ce réseau n'arrivent jamais à accéder aux fichiers contenus dans cet ordinateur. Le technicien a laissé entendre qu'il s'agirait d'un problème d'adresse IP...

Consignes :

1. Définir Adresse IP. (Réponse : est un numéro unique permettant à un ordinateur de communiquer dans un réseau.)
2. Donner une méthode utilisable pour faire communiquer des ordinateurs distants. (Réponse : **les connectés à un réseau**)

Trace écrite

I-Définition

Une adresse IP est un numéro unique permettant à un ordinateur de communiquer dans un réseau.

Il existe deux versions d'adresses IP à savoir IPV4(Internet Protocol version 4) codée sur 32 bits et IPV6(Internet Protocol version 6) codée sur 128 bits.

Une adresse IPV4 se présente sous la forme de quatre champs numériques W.X. Y.Z (Ex : 192.168.1.1), chaque champ représentant un octet pouvant donc prendre une valeur allant de 0

à 255. Une adresse IP se décompose aussi en deux parties : une partie des nombres à gauche désigne le réseau (partie réseau) et les nombres de droite désignent les machines (partie machine ou hôte)

II- Les modes d'adressage

Il existe deux manières d'adresser (donner une adresse IP) à un équipement réseau : adressage statique et adressage dynamique.

1- Adressage statique

Dans ce mode, les adresses sont attribuées manuellement par un individu à travers une interface prévue à cet effet. Il faut au préalable cocher l'option de configuration

2- Adressage dynamique

Dans ce mode, c'est un ordinateur du réseau appelé serveur DHCP (Dynamic Host Configuration Protocol) qui va attribuer de façon automatique les adresses à tous les ordinateurs qui vont se connecter au réseau.

Le protocole DHCP permet l'attribution automatique des informations d'adressage, telles que l'adresse IP, le masque de sous-réseau et d'autres paramètres. La configuration du serveur DHCP nécessite un bloc d'adresses, appelé pool d'adresses, soit défini de manière à être attribué aux clients DHCP d'un réseau.

III- Quelques commandes réseau

Les commandes réseau doivent être saisies dans un terminal ou invite de commande.

1. La commande IPCONFIG

La commande réseau ipconfig permet d'afficher un résumé des propriétés IP de la carte réseau. Pour chaque carte réseau on retrouve l'adresse IP, le masque de sous réseau, la passerelle par défaut. Pour l'utiliser il faut juste saisir ipconfig dans l'invite de commande Windows ou dans un terminal linux par exemple. La figure suivante illustre un exemple d'utilisation de cette commande nous affichant les propriétés IP des différentes cartes de la machine sur laquelle elle est exécutée.

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Clément>ipconfig

Configuration IP de Windows

Carte Ethernet Connexion réseau sans fil 4:
    Suffixe DNS propre à la connexion : home
    Adresse IP. . . . . : 192.168.1.18
    Masque de sous-réseau . . . . . : 255.255.255.0
    Passerelle par défaut . . . . . : 192.168.1.1

Carte Ethernet Connexion au réseau local:
    Suffixe DNS propre à la connexion : home
    Adresse IP. . . . . : 192.168.1.12
    Masque de sous-réseau . . . . . : 255.255.255.0
    Passerelle par défaut . . . . . : 192.168.1.1

C:\Documents and Settings\Clément>
  
```

2. La commande PING

La commande tracert affiche toutes les adresses IP intermédiaire par lesquelles passe un paquet entre la machine locale et l'adresse IP spécifiée. teste la connexion réseau avec une adresse IP distante. Sa syntaxe est **ping destination**. **NB : destination** peut être l'adresse IP de la destination, son nom d'hôte ou son nom de domaine. **Exemple** ping 192.168.10.1 ou ping www.google.cm. La capture ci-dessous montre le résultat de la commande ping www.youtube.com indiquant que l'hôte www.youtube.com est accessible. On peut ainsi voir les réponses de cet hôte de même que les délais de réponse.

```

Microsoft(R) Windows DOS
(C) Copyright Microsoft Corp 1990-2001.

C:\DOCUME~1\CHAD\DESKTOP>ping www.youtube.com

Pinging youtube-ui.l.google.com [74.125.127.113] with 32 bytes of data:

Reply from 74.125.127.113: bytes=32 time=53ms TTL=247
Reply from 74.125.127.113: bytes=32 time=55ms TTL=247
Reply from 74.125.127.113: bytes=32 time=54ms TTL=247
Reply from 74.125.127.113: bytes=32 time=53ms TTL=247

Ping statistics for 74.125.127.113:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 53ms, Maximum = 55ms, Average = 53ms

C:\DOCUME~1\CHAD\DESKTOP>
  
```

3. La commande TRACERT

La commande tracert affiche toutes les adresses IP intermédiaire par lesquelles passe un paquet entre la machine locale et l'adresse IP spécifiée. Sa syntaxe est **tracert[@ip ou nom d'hôte] destination**. **Exemple** tracert 192.168.10.1 ou tracert www.google.cm.

La figure suivante illustre un exemple d'utilisation de la commande tracert montrant l'itinéraire suivi par les paquets pour atteindre la destination google.COM dans ce cas précis.

```

Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\technoblogical>tracert google.com

Tracing route to google.com [74.125.45.100]
over a maximum of 30 hops:
  0  0 ms  0 ms  0 ms  dellmini.mshome.net [192.168.0.1]
  1  4 ms  2 ms  2 ms  192.168.1.100
  2  *  *  *  Request timed out.
  3  11 ms  11 ms  9 ms  68.86.151.117
  4  12 ms  11 ms  11 ms  68.85.174.21
  5  14 ms  54 ms  88 ms  ts-8-2-as01.nashville.tn.nash.comcast.net [68.85.174.171]
  6  27 ms  26 ms  26 ms  68.86.98.73
  7  29 ms  27 ms  25 ms  xe-7-2-0.edge1.Chicago2.Level3.net [4.71.248.25]
  8  28 ms  26 ms  27 ms  vlsn52.sbr2.Chicago2.Level3.net [4.69.138.198]
  9  42 ms  42 ms  48 ms  ae-2-2.sbr2.Washington1.Level3.net [4.69.133.70]
 10  54 ms  53 ms  48 ms  ae-62-62.csw1.Washington1.Level3.net [4.69.134.146]
 11  42 ms  48 ms  48 ms  ae-1-69.edge1.Washington1.Level3.net [4.68.17.16]
 12  43 ms  42 ms  42 ms  GOOGLE-INC.edge1.Washington1.Level3.net [4.79.231.6]
 13  44 ms  43 ms  45 ms  209.85.248.136
 14  44 ms  43 ms  97 ms  72.14.238.136
 15  45 ms  43 ms  42 ms  209.85.254.243
 16  46 ms  52 ms  54 ms  209.85.253.137
 17  44 ms  56 ms  43 ms  yx-in-f100.google.com [74.125.45.100]

Trace complete.

C:\Users\technoblogical>
  
```

Jeu Bilingue :

Mots en français	Traduction en anglais
Réseau	Network
Protocole internet	Internet Protocol
Adresse	Adress

Exercice1

1. Citer les modes d'adressage réseau que vous connaissez
2. Décrire brièvement les modes d'adressage réseau cités plus haut

Solution Exercice1

1. Adressage statique et dynamique
2. Confère cours

Exercice2

Vous disposez d'une machine connecter à un réseau. L'on désire connaître la configuration IP de cette machine ainsi que d'autres informations.

1. Donner la commande permettant d'afficher l'adresse IP de cette machine dans le réseau.
2. Donner la commande permettant de tester l'accessibilité de ce poste à l'adresse

www.yahoo.fr

3. Donner la commande permettant de déterminer le chemin parcouru par les paquets pour atteindre la destination 128.15.10.1.

Solution Exercice2

1. ipconfig
2. pingwww.yahoo.fr
3. tracert128.15.10.1

MODULE2: SYSTÈMES INFORMATIONS ET BASES E DONNEES. 16hTitre de U.A. **CONCEPTION DES MODELES DE DONNEES D'UN SI 6h****U.E 14: METHODES DE CONCEPTIONS D'UN SI (2h)**Compétences visées

- ✚ Déterminer l'importance du système d'information et sa position dans l'entreprise
- ✚ Énumérer les méthodes de conception des SI et leurs domaines d'utilisations
- ✚ Déterminer les cycles de développement de SI et les différents modèles existants

Situation problème : Pendant les vacances, vous êtes admis à faire un stage de vacance dans l'entreprise d'import/export de votre oncle. Il vous exprime le besoin d'informatiser sa structure pour s'arrimer aux technologies mondiales et prendre des décisions en temps réel. Il vous demande de vous en occuper durant votre stage.

CONSIGNE :

- 1) Pensez-vous qu'il a raison ? Rep : il a complètement raison
- 2) Donner Les méthodes que vous pourriez utiliser pour atteindre votre objectif. Et préciser enfin les étapes que vous suivrez. Rep : MERISE, UML, SART et méthode CASCADE ou en V

DEFINITIONS :

Un modèle informatique est une représentation simplifiée de la réalité en vue de réaliser un traitement avec un ordinateur.

La modélisation est la conception d'un modèle

la modélisation des données est l'analyse et la conception de l'information contenue dans le système afin de représenter la structure de ces informations et de structurer le stockage et les traitements.

Un système d'information (SI) : est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information¹, en général grâce à un réseau d'ordinateurs .

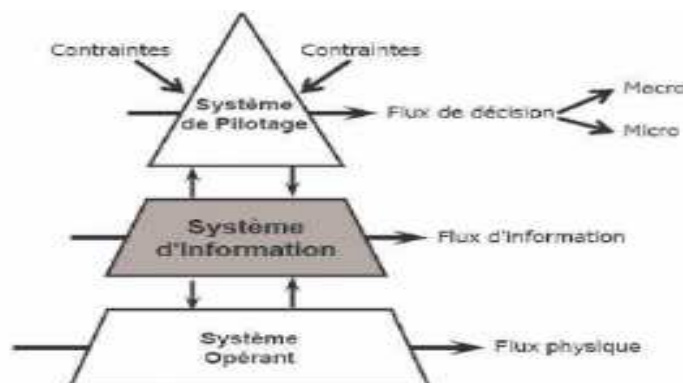
I) La notion de système d'information et sa position dans l'entreprise

Le SI a pour objectif de fournir des informations nécessaires à des utilisateurs qui en ont besoin au moment voulu sous forme convenable afin de les aider à accomplir leurs rôles respectifs au sein d'une organisation.

1) Position d'un système d'information dans l'entreprise

Une organisation est composée d'un ensemble de systèmes. Il y a le système **opérant**, de **pilotage** et d'**information**, ce sont trois sous-systèmes qui interagissent entre eux :

- le **système de pilotage ou de décision** Il se situe à la tête du système d'information fixant les objectifs et prenant les décisions. *Ex recruter une nouvelle personne ; commander la marchandise...*
- le **système d'information** : C'est ce qui intervient entre les deux autres systèmes.
- le **système opérant** : Il correspond aux différents services d'une entreprise. c'est lui qui permet la transformation de l'information dont l'objectif est de la restituer à la bonne personne. *Ex facturer un client, fabriquer les produits...*



2) Les fonctions d'un système d'information dans l'entreprise

Il existe donc 4 fonctions principales d'un SI :

- ✚ **Collecter** : c'est à partir de là que naît la donnée, qu'on acquière les informations provenant de l'environnement interne ou externe à l'entreprise.
- ✚ **Stocker** : dès que l'information est acquise, le système d'information la conserve. Elle doit pouvoir être disponible et doit pouvoir être conservée dans le temps.
- ✚ **Transformer/traiter** : cette phase permet de transformer l'information et choisir le support adapté pour traiter l'information. Ici on construit de nouvelles informations en modifiant le fond ou la forme.
- ✚ **Diffuser** : le SI transmet ensuite l'information dans son environnement interne ou externe.

En Gros, le SI va donc restituer une information au sein d'une organisation directement exploitable par les différents acteurs et **faciliter la prise de décision**.

II) les méthodes de conception d'un système d'information et leurs domaines d'utilisation.

Plusieurs méthodes de conception de S.I. coexistent et sont exploitées différemment selon les objectifs les domaines pays, dont Merise, UML, SART.

- 1) **MERISE** méthode de modélisation procédurale adaptée pour la gestion des projets internes aux organisations, se limitant à un domaine précis. Il se divise en modèle de données et modèle de traitement. Ex gestion des stocks
- 2) Le **Langage de Modélisation Unifié**, de l'anglais *Unified Modeling Language (UML)*, est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.
- 3) **Booch**, **OMT** et **OOSE**, sont des méthodes d'analyse et de conception orientées objet à l'origine d'UML
- 4) L'Analyse fonctionnelle pour le temps réel (en anglais *Structured Analysis for Real Time* ou **SART**) est une méthode permettant de modéliser un système temps réel complexe.

III) les cycles de développement d'un SI

Il existe différents types de **cycles de développement** (modèle) entrant dans la réalisation d'un SI. Ces cycles prennent en compte toutes les étapes de la conception d'un logiciel.

1) Modèle en cascade

Processus de nature plutôt linéaire, centré sur **activités** avec un ordonnancement prédéfini des activités, dans lesquels il est toujours possible de revenir en arrière. Les étapes sont les suivantes :

Initiation et étude de faisabilité : Évaluer la faisabilité technique, économique, opérationnelle.

Spécification des besoins : Analyser le domaine d'application, Identifier les exigences, Élaborer un glossaire de termes

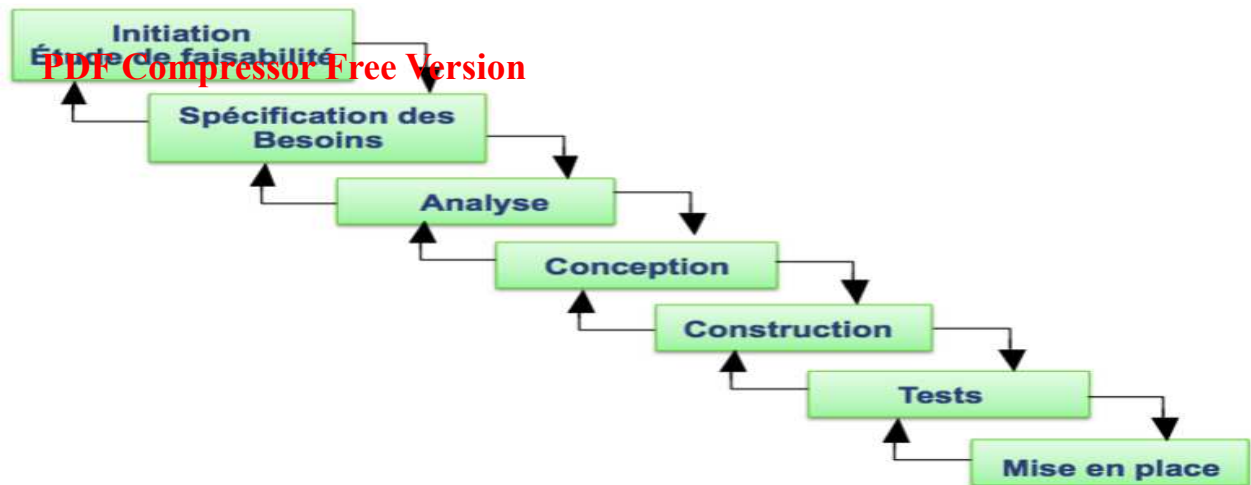
Analyse : Analyser les exigences, Proposer une ou plusieurs solutions conceptuelles,

Conception : Spécification détaillée de la solution choisie

Construction : Créer les programmes, Créer et remplir les bases de données

Test : Tester le système global

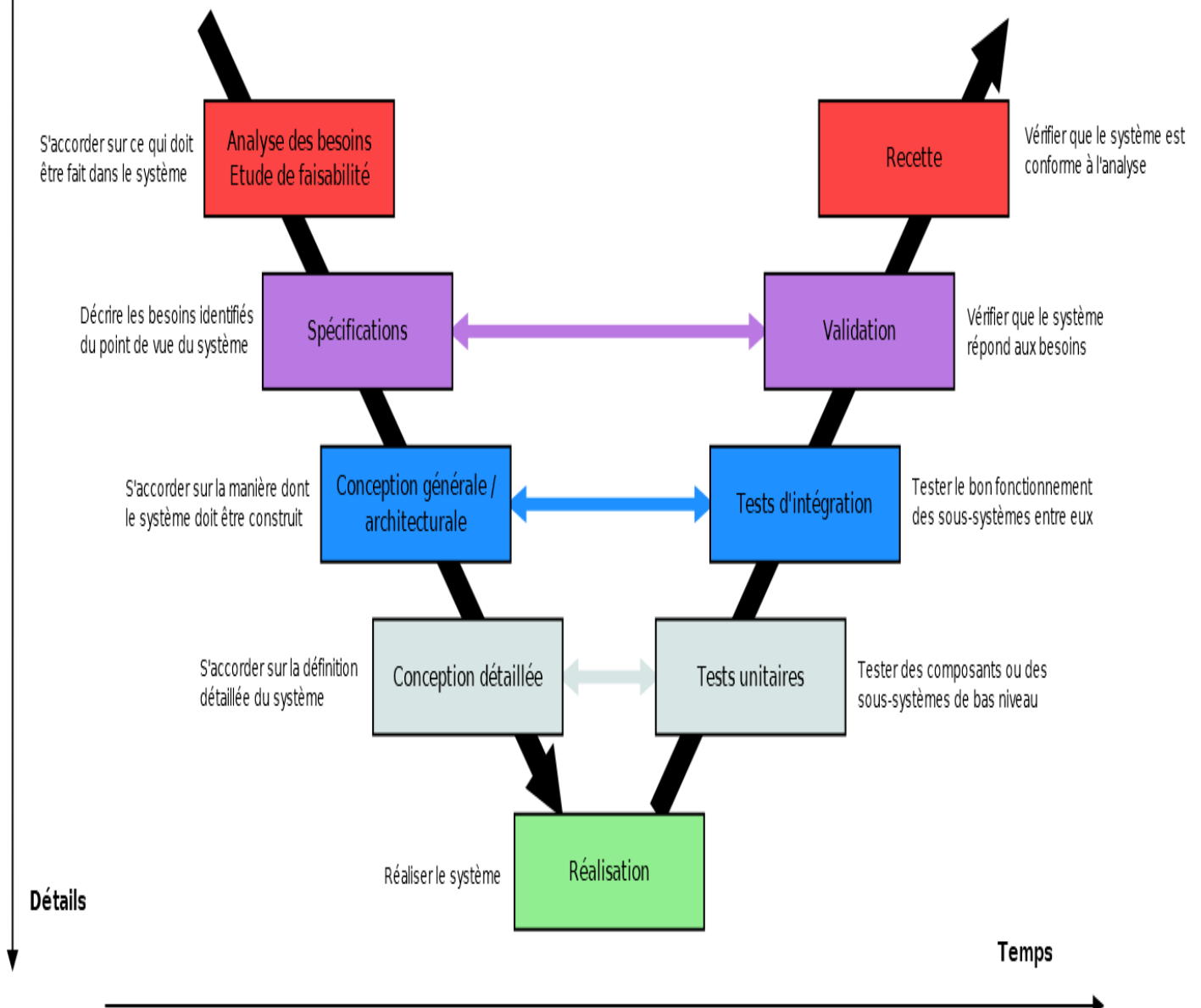
Mise en place : S'assurer que tout le matériel et les réseaux nécessaires pour le nouveau SI sont mis en place, Installer le système, Former l'utilisateur



2) Model en V

Décomposition du système en composants qui peuvent être développés de manière indépendante. Ce modèle permet d'avoir une meilleure réactivité que le modèle en cascade : il permet de limiter les retours en arrière en cas d'anomalie.

PDF Compressor Free Version



JEUX BILINGUE

Mots en français	Traduction en anglais
Système d'information	Information system
Modélisation	Modeling
Modèle	model

EXERCICES

Exercice 1 savoir :

- 1) **Définir** :Modèle Informatique, Modélisation, Systèmes D'information
- 2) **PDF Compressor Free Version** Nommez 3 sous-systèmes dans la gestion d'une entreprise, en précisant le rôle de chacun.
- 3) *Enumérer 4 fonctions principale d'un SI*
- 4) Citez 3 méthodes de conception des SI
- 5) Nommez 2 modèles de développement d'un SI
- 6) Déterminer 2 différences entre ses 2 modèles

Exercice 2 : savoir faire

Soit les taches de suivantes : **Construction - Spécification des besoins - Mise en place - Analyse - Conception - Initiation et étude de faisabilité - Test**

- 1) Déterminer le modèle de conception de SI auquel vous pourriez les associer
- 2) Identifier la première étape et donnez son rôle.
- 3) Identifier la dernière étape et donnez son rôle.
- 4) Ranger ces taches en ordre pour un modèle en cascade
- 5) Déterminer l'étape à laquelle ont n'analyse les spécifications de l'utilisateur final
- 6) Déterminer l'étape à laquelle ont Créé et rempli les bases de données
- 7) Enumérer les étapes du model en V

IV) CORRIGE

PDF Compressor Free Version
Exercice 1

1) Définir : *Modèle Informatique, Modélisation, Systèmes D'information*

Un modèle informatique est une représentation simplifiée de la réalité en vue de réaliser un traitement avec un ordinateur.

La modélisation est la conception d'un modèle

Un système d'information (SI) : est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information¹, en général grâce à un réseau d'ordinateurs .

2) Nommez 3 sous-systèmes dans la gestion d'une entreprise, en précisant le rôle de chacun.

- le **système de pilotage ou de décision** Il se situe à la tête du système d'information fixant les objectifs et prenant les décisions. *Ex recruter une nouvelle personne ; commander la marchandise...*
- le **système d'information** : C'est ce qui intervient entre les deux autres systèmes.
- le **système opérant** : Il correspond aux différents services d'une entreprise. c'est lui qui permet la transformation de l'information dont l'objectif est de la restituer à la bonne personne. *Ex facturer un client, fabriquer les produits...*

3) Enumérer 4 fonctions principales d'un SI en précisant le rôle pour chacun d'eux

- ✚ **Collecter** : c'est à partir de là que naît la donnée, qu'on acquière les informations provenant de l'environnement interne ou externe à l'entreprise.
- ✚ **Stocker** : dès que l'information est acquise, le système d'information la conserve. Elle doit pouvoir être disponible et doit pouvoir être conservée dans le temps.
- ✚ **Transformer/traiter** : cette phase permet de transformer l'information et choisir le support adapté pour traiter l'information. Ici ont construit de nouvelles informations en modifiant le fond ou la forme.
- ✚ **Diffuser** : le SI transmet ensuite l'information dans son environnement interne ou externe.

En Gros, le SI va donc restituer une information au sein d'une organisation directement exploitable par les différents acteurs et **faciliter la prise de décision**.

4) Citez 3 méthodes de conception des SI

- **MERISE** (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise.)

- UML(Unified Modeling Language)
- **PDF Compressor Free Version**
SART(Structured Analysis for Real Time)

5) Nommez 2 modèles de développement d'un SI

Modèle en cascade -- Modèle en V

6) Déterminer la différences entre ses 2 modèles

- la méthode en cascade ne permet pas de retour aux étapes précédentes. Par contre, le cycle en V rend possible le retour en arrière et ce, même lorsque l'étape a été validée.
- Dans le Modèle en V ont décompose le système en composants qui peuvent être développés de manière indépendante. Ce qui n'est pas le cas dans le Modèle en cascade

Exercice 2 : savoir faire

Soit les taches de suivantes : **Construction - Spécification des besoins - Mise en place - Analyse - Conception - Initiation et étude de faisabilité - Test**

1) Déterminer le modèle de conception de SI auquel vous pourriez les associer

Il s'agit du **Modèle en cascade**

2) Identifier la première étape et donnez son rôle.

Initiation et étude de faisabilité : Évaluer la faisabilité technique, économique, opérationnelle.

3) Identifier la dernière étape et donnez son rôle.

Mise en place : S'assurer que tout le matériel et les réseaux nécessaires pour le nouveau SI sont mis en place, Installer le système, Former l'utilisateur

4) Ranger ces taches en ordre pour un modèle en cascade

Initiation et étude de faisabilité -- Spécification des besoins -- Analyse -- Conception -- Construction -- Test -- Mise en place

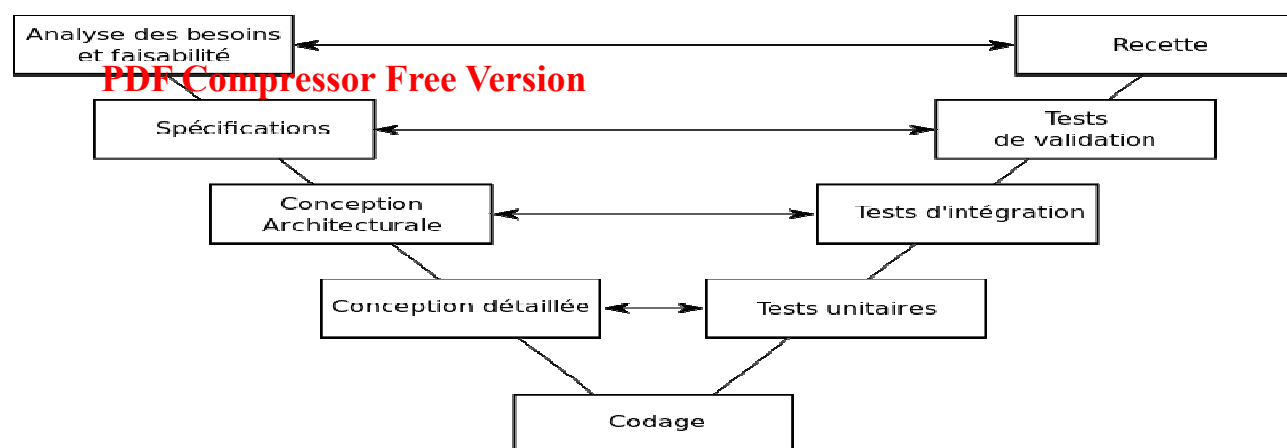
5) Déterminer l'étape à laquelle ont n'analyse les spécifications de l'utilisateur final

Analyse

6) Déterminer l'étape à laquelle ont Créé et rempli les bases de données

Construction

7) Enumérer les étapes du model en V



MODULE 2: SYSTÈMES D'INFORMATION ET BASE DE DONNEES

PDF Compressor Free Version

UA5: Conception des modèles de données d'un S.I

UE15: ELABORATION D'UN MCD

Compétences visées :

- ✓ Utiliser les règles de gestion pour élaborer un MCD
- ✓ Élaborer un MCD

Situation problème : Dans l'optique de concevoir une application qui va gérer l'aspect commercial de l'entreprise dans laquelle vous avez effectué votre stage, vous êtes emmené sortir un modèle de donnée qui permettra aux décideurs de l'entreprise de mieux comprendre comment le futur système doit fonctionner en voyant clairement les relations existantes entre les différentes entités pertinentes du système. comment procéder pour ressortir un tel modèle?

INTRODUCTION

Le MCD (modèle conceptuel de données) est un modèle abstrait de la méthode Merise permettant de représenter l'information d'une manière compréhensible aux Différents services de l'entreprise. Il permet une description statique du système d'information à l'aide d'entités et d'associations.

I. DÉFINITION

Modèle: Représentation abstraite et simplifiée d'un problème permettant de le comprendre facilement et de faire des simulations.

Modélisation: La modélisation consiste à créer une représentation simplifiée d'un problème: le modèle.

Règles de gestion: Ensembles de contraintes que doivent respecter les données du système d'information pour refléter la parfaite réalité du fonctionnement de l'organisation ou de l'entreprise.

II. NOTIONS DE BASES

1) LA PROPRIÉTÉ

La propriété est une information élémentaire, c'est-à-dire non déductible d'autres informations, qui présente un intérêt pour le domaine étudié. Par exemple, si l'on considère le domaine de gestion des commandes d'une société de vente par correspondance, les données : « référence article », « désignation article », « prix unitaire HT », « taux de TVA » sont des propriétés pertinentes pour ce domaine. La donnée « prix unitaire TTC » n'est, d'après la

définition, pas une propriété car ses valeurs peuvent être retrouvées à partir des propriétés «prix unitaire HT» et «taux de TVA».

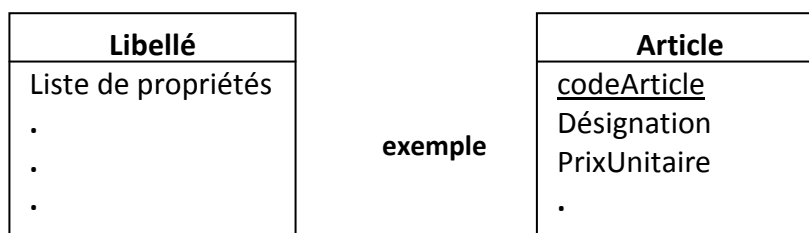
Chaque valeur prise par une propriété est appelée occurrence. Des occurrences de la rubrique « désignation article » sont par exemple : « râteau », « bêche », « scie », ...

Une propriété est dite simple ou encore atomique si chacune des valeurs qu'elle regroupe n'est pas décomposable. La propriété « Adresse », n'est pas élémentaire car elle peut être décomposée en plusieurs propriétés : la rue, le code postal et la ville etc.

2) ENTITÉ

L'entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire. Au premier abord, on peut définir l'entité comme étant un regroupement sensé, de plusieurs propriétés. Par exemple, on considère l'entité ARTICLE qui regroupe les propriétés : Référence, Désignation et PrixUnitaireHT.

Les entités sont représentées par un rectangle. Ce rectangle est séparé en deux champs: le champ du haut contient le libellé. Le champ du bas contient la liste des propriétés de la classe d'entité.



3) OCCURRENCE D'ENTITÉ

D'après la définition d'une entité, on sait que la connaissance d'une valeur de la rubrique identifiant détermine la connaissance des valeurs des autres rubriques de l'entité. L'ensemble de ces valeurs est appelé occurrence d'entité. Le tableau suivant présente des exemples d'occurrences de l'entité ARTICLE.

352GT Stylo 5F	134ER RATEAU 150F	354TY Cahier 100F
----------------------	-------------------------	-------------------------

4) LES IDENTIFIANTS

Un identifiant est une propriétés (ou plusieurs) permettant de désigner une et une seule entité. L'identifiant est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cet objet pour lesquelles cette propriété pourrait prendre une même valeur. Le modèle conceptuel des données propose de souligner les identifiants (parfois

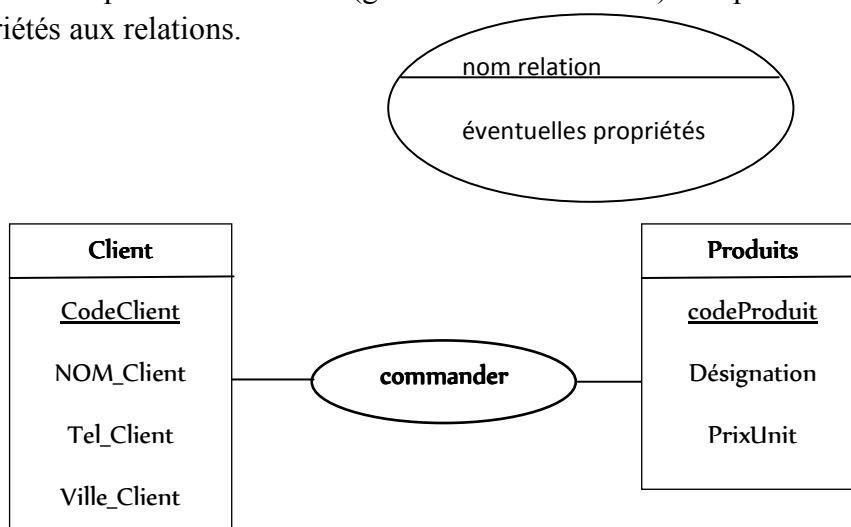
de les faire précéder d'un #). Ainsi, chaque entité doit posséder au moins un attribut identifiant.

5) L'ASSOCIATION

Une association (appelée aussi parfois relation) est un lien sémantique entre plusieurs entités. Une relation peut lier plus de deux entités. Selon le nombre d'entités liés on distingue :

- Relation récursive (ou réflexive) relie une entité à elle-même.
- Relation binaire relie deux d'entité.
- Relation ternaire relie trois entités.
- Relation n-aire relie n entités.

Les relations sont représentées par des hexagones (parfois des ellipses) dont l'intitulé décrit le type de relation qui relie les entités (généralement un verbe). On peut éventuellement ajouter des propriétés aux relations.

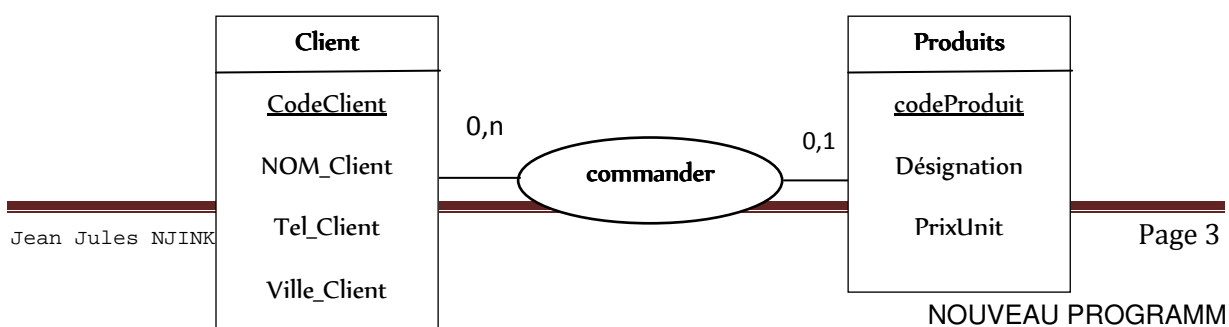


6) CARDINALITÉ

Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité représente le nombre minimal et maximal d'occurrences de l'entité qui peuvent participer à une occurrence de l'association. La cardinalité d'une relation est composé d'un couple comportant une borne minimale et une borne maximale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur :

- La borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une occurrence de l'entité peut participer à une relation.
- La borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une occurrence de l'entité peut participer à une relation.

Les cardinalités sont mentionnées par couple du côté de l'entité à considérer. La cardinalité minimale est représentée en premier, et la maximale en second.



PDF Compressor Free Version

NB: les cardinalités doivent obéir aux règles de gestion de l'entreprise.

III. CONSTRUCTION DU MCD

Le MCD est une représentation des données, facilement compréhensible. Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur les concepts « entité-association ».

La Construction du MCD doit passer par un ensemble d'étapes dont les principales sont:

- Repérer les concepts pertinents pour la résolution du problème et déduire les entités
- Rattacher à ces entités leurs propriétés
- Identifier les associations entre entités. (reconnu en général par les verbes d'action)
- Grâce aux règles de gestion, étudier les cardinalités de chaque couple Entité – Relation

Exemple d'application

Dans un super marché il est donné la possibilité aux clients d'effectuer des commandes sur plusieurs produits. Par souci de prudence, un produit peut être livré par plusieurs fournisseurs. Chaque Client du super marché est identifié par un identifiant, son nom, et son contact; Chaque produit est repéré par un code, une désignation et son prix unitaire. Les fournisseurs également sont représentés par un identifiant, un nom, la ville de résidence et le contact.

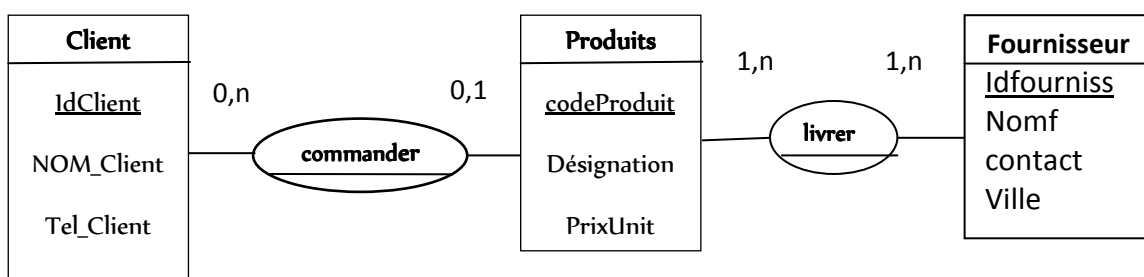
Élaborer le MCD correspondant au système d'information de ce super marché.

1- Recherche des Concepts pertinents ou entités

On a les concepts suivants: **Client-Produit-fournisseur**

2- Recherche des associations entre entités

On a les associations suivantes: **commander**(entre client-Produit) et **livrer**(entre fournisseur-produit)



PDF Compressor Free Version

IV. NORMALISATION DU MCD

L'objectif de la normalisation est de construire un schéma de Base de données cohérent. Un mauvais schéma conceptuel et relationnel peut conduire à un certains nombres d'anomalies pendant la phase d'exploitation de la base de données.

La normalisation du MCD permet donc de garantir une base de donnée cohérente et optimale. Les règles de normalisation sont les suivantes :

1. Normalisation des entités et association

→ Le nom d'une entité/association doit être concis et parlants et en rapport avec le domaine de gestion étudié ;

→ Le nom d'une entité, d'une association ou d'un attribut doit être unique ;

Éliminer les associations fantômes c'est-à-dire les associations ayant pour cardinalité maximale 1 de chaque coté de l'association.(Ce jeu de cardinalité traduit généralement une erreur de conception.)

2. Normalisation des identifiants

→ Chaque entité doit posséder un identifiant ;

3. Normalisation des attributs

→ Toutes les propriétés d'une entité ou d'une association doivent être atomique ou élémentaires (non décomposables);

→ Les attributs calculables doivent être retirés des entités;

→ Les attributs d'une association doivent dépendre directement des identifiants de toutes les entités en association

4. Normalisation des cardinalités :

Une cardinalité minimale est toujours 0 ou 1(et pas 2, 3 ou n) et une cardinalité maximale est toujours 1 ou n (et pas 2,3, ...)

CONCLUSION

Le modèle conceptuel des données est une représentation statique du système d'information de l'entreprise qui met en évidence sa sémantique. Il a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible. Cet aspect recouvre les mots qui décrivent le système ainsi que les liens existants entre ces mots. Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur les concepts « entité-association ».

jeu bilingue

conception = design

modèle = model

Règles de gestion = Rules Management

Modèle Conceptuel de Donnée = Conceptual Data Model

entité = entity

Identifiant = Identifier

EXERCICES1

Définir les concepts suivant:

Modélisation, Entité, Association, Cardinalité,

EXERCICE 2:

On voudrait informatiser la gestion des examens officiels au Cameroun. En effet, un candidat caractérisé par son matricule, son nom, son prénom, sa date de naissance peut s'inscrire à plusieurs examen (no_examen, nom_examen). Un examen se déroule dans un centre d'examen (code_centre, nom_centre) ; Pendant les examens, chaque candidat est affecté dans une salle de classe ayant un numéro, et une capacité définie. Il vous est demandé de :

- 1- Identifier les concepts pertinents pour la résolution de ce problème.
- 2- Identifier Trois règles de gestion dans le fonctionnement des Examens
- 3- Dresser le MCD pour la résolution du problème ci-dessus.

solution

1- concepts pertinents

Examen, candidat, CentreExamen, SalleClasse.

2- Trois règles de gestion

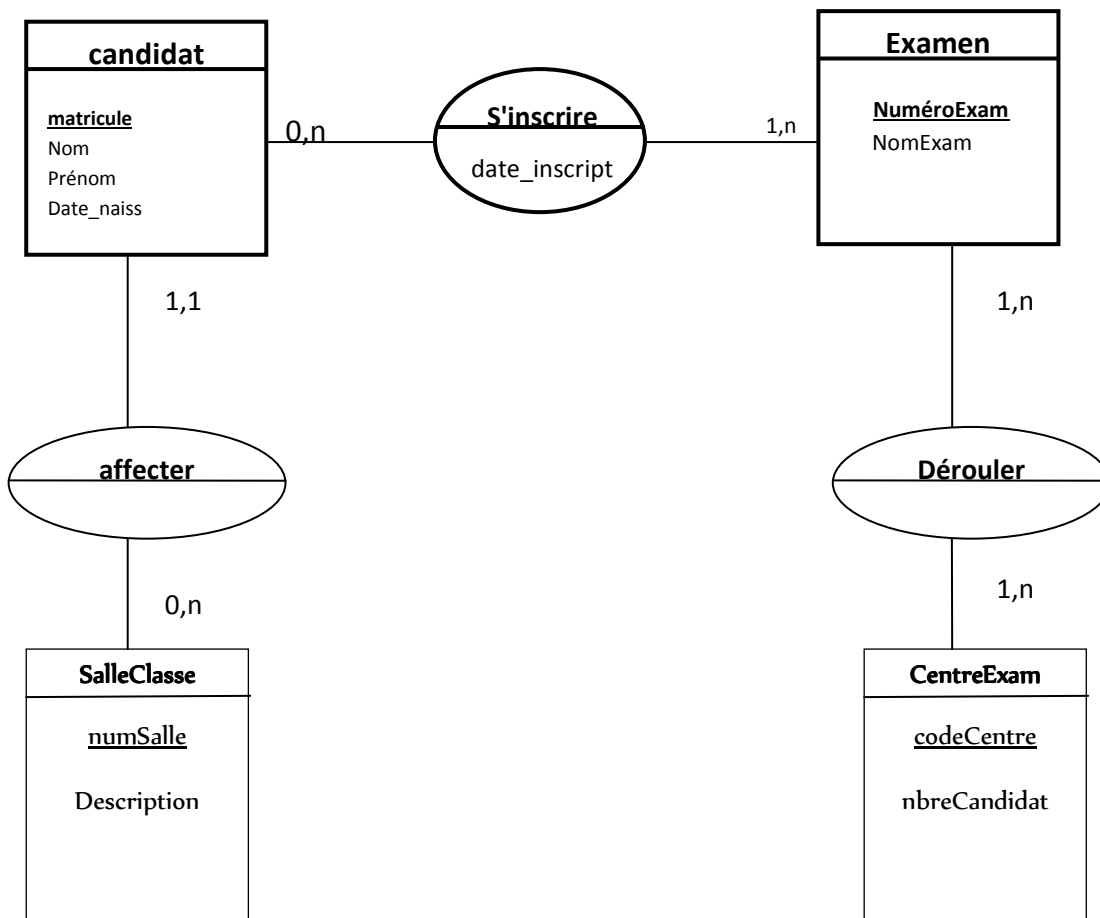
RG1: un candidat peut s'inscrire à plusieurs examen

RG2: un candidat ne peut être affecté que dans une seule classe d'examen

RG3: Un Examen peut se dérouler dans plusieurs centre d'examen

PDF Compressor Free Version

3- Élaboration du MCD



EXERCICE3:

On voudrait concevoir un mini système d'information pour gérer la situation matrimoniale dans une société. Le Système d'information devra prendre en considération les règles de gestion suivantes :

- la polygamie est autorisée dans cette société comme au Cameroun,
- la polyandrie (une femme ne peut avoir plusieurs maris) est catégoriquement interdite,
- nous avons des célibataires et des mariés dans cette société
- nous avons des chômeurs et des travailleurs
- certains hommes dans cette société travaillent dans plusieurs entreprises.

Un homme (numeroH, nom, prénom) peut épouser une femme (numeroF, nom, prénom) et travailler dans une entreprise (numeroE, type, nomE)

TAF

- Identifier les concepts pertinents pour la résolution de ce problème
- Ressortez le Modèle conceptuel de données correspondant au problème posé.

MODULE 1 : SYSTEMES INFORMATIQUES

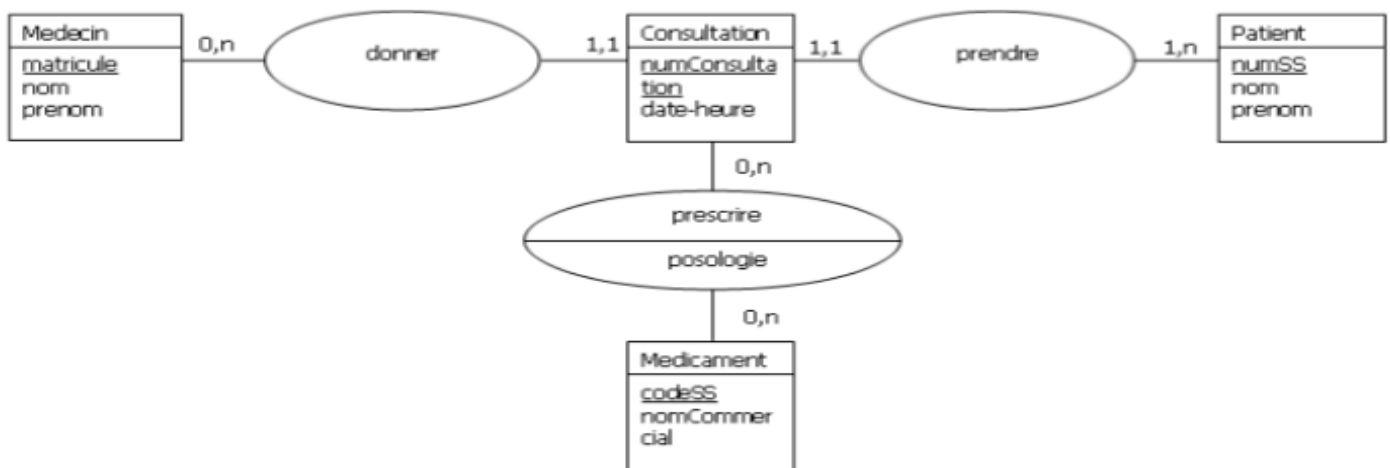
PDF Compressor Free Version

UNITE D'APPRENTISSAGE 5 : CONCEPTION DES MODELES DE DONNEES D'UN SI**UE 16 :RÈGLES DE PASSAGES DU MCD AU MLD ET CONSTRUCTION D'UN MLD****Compétences visées :**

- Enoncer les règles de passage du MCD au MLD;
- Utiliser les règles ci-dessus pour déduire le MLD.

Situation problème :

On vous donne le MCD représentant des visites dans un centre médical.

**Consignes:**

1. Identifier, puis relever les liens des cardinalités de ce MCD en précisant l'association en question. (Réponse : 0, n-1,1 de l'association donner, 1,1-1, n de l'association prendre et 0,n-0,n pour l'association prescrire)
2. Donner le nombre de type de lien que comporte ce MCD. (Réponse : deux car 0, n-1,1 de l'association donner, 1,1-1, n de l'association prendre correspondent au même type)
3. Déduire comment construire le MLD à partir du MCD. (Réponse : en appliquant les règles de passages du MCD au MLD et en normalisant les clés primaires et étrangères)

Trace écrite**1. Notion de MLD**

Un **modèle logique de données** (MLD) est la représentation des données d'un système d'information. Les données sont représentées en prenant compte le modèle technologique qui sera utilisée pour leur gestion. Cette représentation résulte d'un MCD, en appliquant certaines règles en fonctions des types de liens entre entités.

2. Les notations des relations entre les entités

En notant uniquement les cardinalités maximums c'est à dire à l'ordre n, on distingue 3 types de liens :

- 1 : 1 pour les cardinalités max 1 de chaque côté, c'est-à-dire **1,1-1,1 ou 0,1 - 0,1 ou 0,1-1,1**
- 1 : n si l'une et une seule des cardinalités max est n, c'est-à-dire **1, n-0, 1 ou 1,n - 1,1 ou 0, n-0, 1 ou 0,n - 1,1**

- $n : m$ pour les cardinalités max n de chaque côtés, c'est-à-dire $1, n-1, n$ ou $0, n - 0, n$ ou $0, n-1, n$

3. Les règles de passage du MCD au MLD

Le passage du MCD au MLD se fait via 4 règles en fonction du (ou des) type(s) de lien(s) que comporte(nt) le MCD :

- **Règle 1 : Transformation des entités en tables**, c'est-à-dire que chaque entité du modèle conceptuel devient une table dans le modèle logique et conserve les mêmes propriétés qui deviennent des attributs. L'identifiant de l'entité devient la clé primaire de la table.
- **Règle 2 : Destruction de la relation 1 : n**, c'est-à-dire que dans les associations de type « un à plusieurs » l'association disparaît et l'identifiant de l'entité forte migre dans l'entité faible pour devenir une clé étrangère.

Rappel : association « un à plusieurs » est une association où la cardinalité maximale d'un côté est 1 et de l'autre N.

- **Règle 3 : Transformation de la relation $n : m$ en table**, c'est-à-dire que dans les associations de plusieurs à plusieurs, l'association devient une table, sa clé primaire sera la concaténation (combinaison) des identifiants de toutes les entités qui interviennent dans l'association et les propriétés de l'association deviennent des attributs.
- **Règle 4 : Destruction de la relation 1:1 et application de la contrainte d'unicité sur la clé étrangère.**

Cas particuliers : Dans les associations dont les cardinalités maximales sont à 1

- ✓ $0, 1-1, 1$ équivaut à « un à plusieurs » ($x, 1-x, n$) : L'association disparaît et l'identifiant de l'entité forte « $0, 1$ » migre vers entité faible « $1, 1$ » pour devenir une clé étrangère.
- ✓ $1, 1-1, 1$ ou $0, 1 - 0, 1$: Les identifiants migrent de part et d'autre de l'association. Dans ce cas, on privilégie une des références en supprimant une colonne.

NB : Etablir les liaisons entre les clés primaires et étrangères.

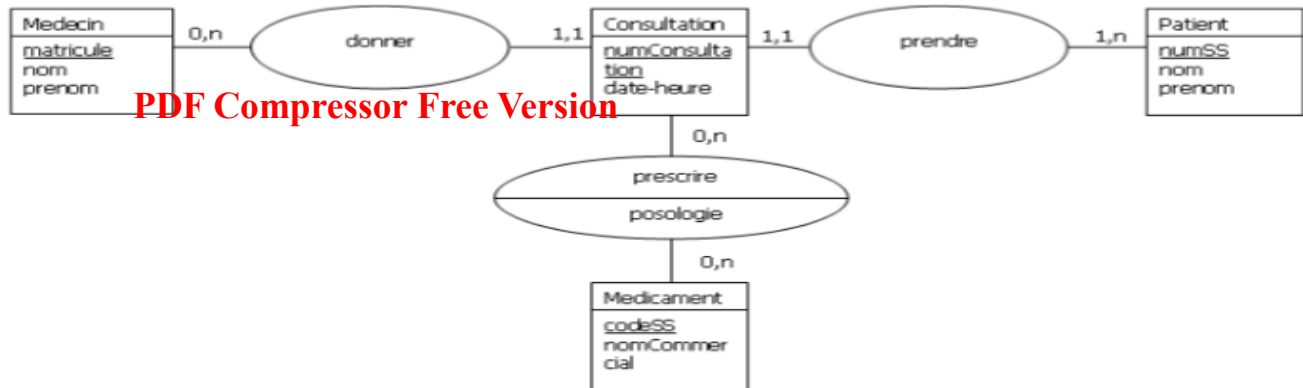
4. Normalisation des clés primaires et étrangères

La normalisation des clés primaires et clés étrangères se fait avec :

- La clé primaire soulignée ;
- La clé étrangère précédée ou suivie de dièse (#).

5. Application

Construire le MLD à partir du MCD représentant des visites dans un centre médical en utilisant les règles de passage du MCD au MLD.



Résolution :

Règle 1 : Transformation des entités en tables ;

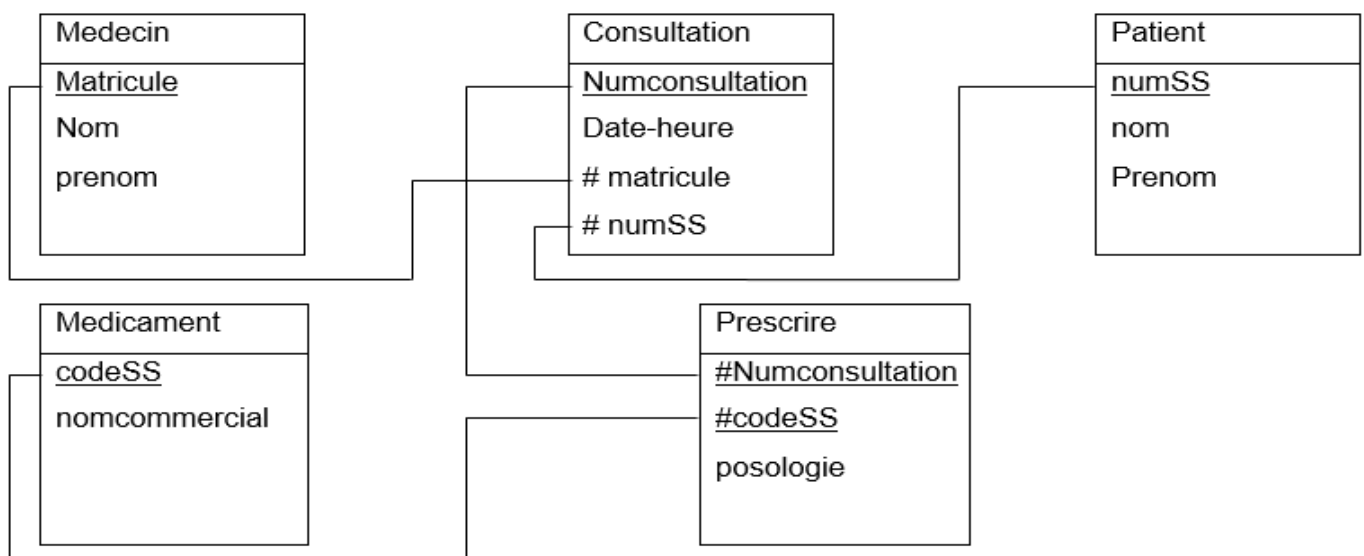
Règle 2 : Destruction de la relation 1 : n

- Pour l'association « donner » (0,n - 1,1) ainsi l'association disparaît et l'identifiant (matricule) de l'entité (Medecin) forte migre dans l'entité faible (Consultation) pour devenir une clé étrangère (#matricule).
- Pour l'association « prendre » (1,1-1,n) ainsi l'association disparaît et l'identifiant (numSS) de l'entité (Patient) forte migre dans l'entité faible (Consultation) pour devenir une clé étrangère (#numSS).

Règle 3 : Transformation de la relation n :m en table, (associations de plusieurs à plusieurs)

- Pour l'association « prescrire » (0,n-0,n) ainsi l'association (prescrire) devient une table, sa clé primaire sera la concaténation (combinaison) des identifiants (**Numconsultation et codeSS**) de toutes les entités (**consultation et Medicament**) qui interviennent dans l'association et la propriété de l'association (posologie) devient attribut de la table.

NB: Les clés primaires soulignées, Les clés étrangères précédées de dièse (#), Etablir les liaisons clé primaire clé étrangère.



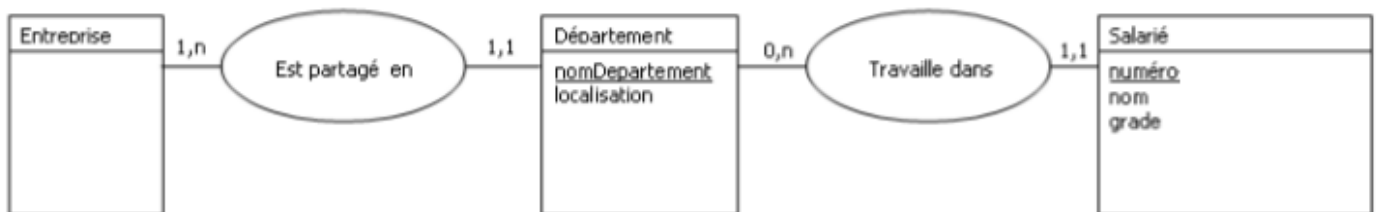
Jeu Bilingue :

PDF Compressor Free Version

Mots en français	Traduction en anglais
Modèle	model
Système d'information	Information system
Règle de gestion	Rules management
Attribut	attribute
Identifiant	username
Clé primaire	Primary key
Clé étrangère	Foreign key
Association	Association
Cardinalité	Cardinality

Exercice 1 :

Soit le MCD d'une entreprise de la place :



1. Compléter les attributs de l'entité Entreprise avec comme caractéristiques son code, nom et téléphone.
2. Construire le MLD en utilisant les règles de passage du MCD au MLD.

Résolution exercice 1 :

1. Compléter les attributs de l'entité Entreprise avec comme caractéristiques son code, nom et téléphone.

Entreprise
<u>Code</u>
Nom
Téléphone

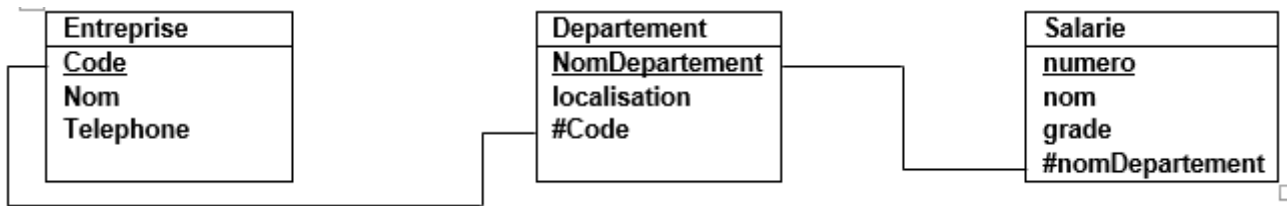
2. Construire le MLD en utilisant les règles de passage du MCD au MLD.

Règle 1 : Transformation des entités en tables ;**Règle 2 : Destruction de la relation 1 : n**

- Pour l'association « est partagé en » (1,n - 1,1) ainsi l'association disparaît et l'identifiant (code) de l'entité (Entreprise) forte migre dans l'entité faible (Département) pour devenir une clé étrangère (#code).
- Pour l'association « travaille dans » (0,n-1,1) ainsi l'association disparaît et l'identifiant (nomDépartement) de l'entité (Département) forte migre dans l'entité faible (Salarie) pour devenir une clé étrangère (#nomDeparment).

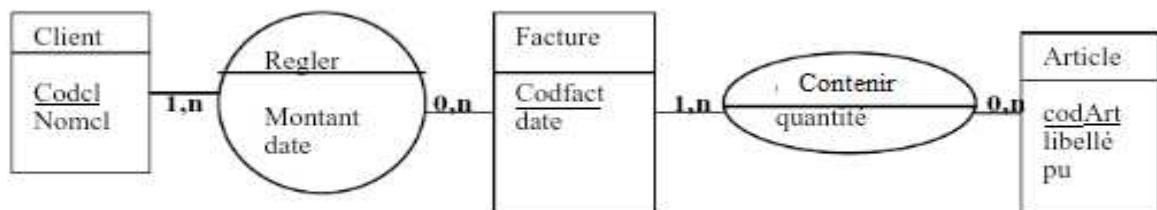
NB :

- Les clés primaires soulignées ;
- Les clés étrangères précédées de dièse (#).
- Etablir les liaisons clé primaire clé étrangère.



Exercice 2 :

Construire le MLD à partir du MCD suivant en utilisant les règles de passage du MCD au MLD.



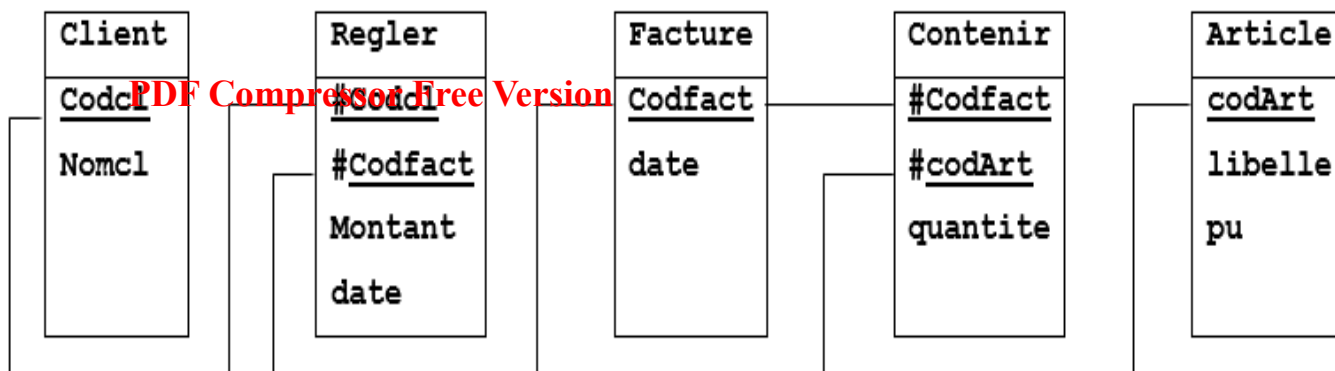
Résolution exercice 2 :

Règle 1 : Transformation des entités en tables ;

Règle 3 : Transformation de la relation n :m en table, (associations de plusieurs à plusieurs)

- Pour l'association « Regler » (1,n-0,n) ainsi l'association (Regler) devient une table, sa clé primaire sera la concaténation (combinaison) des identifiants (Codcl et Codfact) de toutes les entités (Client et Facture) qui interviennent dans l'association et les propriétés de l'association (Regler) devient attribut de la table.
- Pour l'association « Contenir » (1,n-0,n) ainsi l'association (Contenir) devient une table, sa clé primaire sera la concaténation (combinaison) des identifiants (Codfact et codArt) de toutes les entités (Facture et Article) qui interviennent dans l'association et la propriété de l'association (Contenir) devient attribut de la table.

NB: Les clés primaires soulignées, Les clés étrangères précédées de dièse (#), Etablir les liaisons clé primaire clé étrangère.



MODULE 1 : SYSTEMES INFORMATIQUES

UNITE D'APPRENTISSAGE 6 : IMPLEMENTATION D'UNE BASE DE DONNEES (4H)

Situation problème :

Après avoir créé une base donnée en mode graphique via le SGBD Accesset y inséré des informations, votre enseignant M. NJOUONANG vous fait savoir qu'à présent pour interagir avec cette BD il faut utiliser un langage bien précis.

Consigne : en utilisant vos connaissances acquises sur les leçons précédentes, identifier de quel langage il s'agit.

UE 17 : INTRODUCTION AU LANGAGE SQL

Compétences visées :

- Donner la signification du sigle SQL
- Enumérer les sous-langages ou parties du SQL
- Enumérer les types de données manipulées en SQL



Comme dans la vie, pour que des personnes puissent se comprendre, elles doivent parler le même langage et bien en informatique c'est pareil.

Pour que les différents logiciels et le moteur de base de données puissent se comprendre, ils utilisent un langage appelé SQL.

I- GENERALITES SUR LE SQL

Le SQL (Structured Query Language) en français (langage de requête structuré) est un langage Standard permettant à un client de communiquer des instructions à la base de données. Créé au début des années 1970 par Donald Chamberlin et Raymond Boyce, deux ingénieurs de chez IBM, le langage SQL est aujourd'hui reconnu comme une norme internationale. Il se décline en 04 parties et comme tout langage possède une syntaxe bien précise avec des instructions permettant (types de données, contraintes d'intégrité) permettant de décrire les données qui seront manipulées.

II- LES SOUS-LANGAGE DU SQL

Ainsi donc se décline en 04 parties qui sont :

- **LDD (Langage de définition des données)** : cela veut dire qu'il permet de créer des bases de données ainsi que des tables dans une Base de données (avec les mots-clés *CREATE*, *ALTER*, *DROP*, *RENAME*... et sont généralement suivis de *DATABASE*, *TABLE*...)
- **LMD (Langage de Manipulation des données)** : cela signifie qu'il permet de sélectionner, insérer, modifier, ou supprimer des données dans une base de données par les mots-clés *SELECT*, *INSERT*, *UPDATE* ou *DELETE*

- **LCD (Langage de contrôle de données)** : qui permet d'autoriser des actions à certaines personnes, d'ajouter ou de supprimer des autorisations (*GRANT* et *REVOKE*)
- **LCT (Langage de contrôle de transaction)** : permettant de confirmer ou annuler l'exécution d'une transaction (*COMMIT*, *ROLLBACK*).

Dans le cadre de ce cours nous nous limiterons aux LDD et LMD.

III- LES TYPES DE DONNEES UTILISEES EN SQL

Comme n'importe quel autre langage SQL propose les types de données suivants :

- Numériques : INTEGER (32 bits), SMALLINT (16 bits), FLOAT
- Chaines de caractères : CHAR(n), VARCHAR(n)
- Date : DATE (pour représenter une date sous le format XX/XX/XXXX)
- Heure : TIME (pour représenter l'heure)

IV- LES CONTRAINTES D'INTEGRITE EN SQL

Ce sont des règles qui édictent les propriétés que doivent respecter les champs de la base de données afin de garantir leur **cohérence**, leur **pertinence** et leur **validité**. Ces contraintes doivent être exprimées dès la création de la table grâce aux mots clés suivants :

- **CONSTRAINT** : permet de spécifier une clé primaire ou clé étrangère
- **DEFAULT** : Permet de garantir qu'un champ ne sera pas vide
- **NOT NULL** : permet de spécifier qu'un champ doit être saisi
- **UNIQUE** : cette clause permet de vérifier que la valeur saisie pour un champ n'existe pas déjà.
- **CHECK ()** : permet de faire un test sur un champ comportant une condition logique portant sur une valeur entre les parenthèses.

Jeu Bilingue :

Mots en français	Traduction en anglais
SQL(StructuredQueryLanguage)	Langage de Requêtes Structurées(LRS)

Exercice

- 1- Donner en français la signification du sigle SQL
- 2- Quel sous-langage du SQL permet de créer les bases de données ?
- 3- Enumérer les types de données manipulés en SQL et proposer un exemple de données pour chaque type
- 4- Citer 02 contraintes d'intégrité de votre choix tout en expliquant leurs rôles.

BDE Compressé Version 1.0

MODULE 9 : SYSTEMES D'INFORMATION ET BASES DE DONNEES

UNITE D'APPRENTISSAGE 6 : IMPLEMENTATION D'UNE BASE DE DONNEES

UE 18 : ÉCRITURE ET EXECUTION DES REQUETES SQL DE DEFINITION ET DE MANIPULATION DES DONNEES

Compétences visées :

- ❖ Écrire les requêtes SQL de définition des données sur une seule table (création, modification, suppression) ;
- ❖ Écrire/exécuter les requêtes SQL de manipulation des données sur une seule table (insertion, modification, sélection, suppression) ;
- ❖ Utiliser les opérateurs logiques (AND et OR) et de comparaison (=, <, >, <=, >=) ;

Situation problème :

Vous êtes informaticiens au sein de votre établissement. Le fondateur vous demande de mettre sur pied une base de donnée pour la gestion des élèves de la classe de terminale. Ils souhaitent pouvoir Enregistrer (nom, prénom, âge, sexe) ; en sachant que chaque élève est identifié de façon unique avec son matricule, le fondateur doit être à mesure de modifier, supprimer, de lister les élèves par ordre alphabétique, être capable de savoir le nombre de filles et de garçon, rechercher un élève en fonction de son matricule.

Consignes

1. Créer la base de donnée nommer ecole
2. Créer la table élève
3. Ecrire la commande qui permet au fondateur de vous enregistrer en tant qu'élève.
4. Sachant qu'un élève procèdent une date de naissance, écrire la requête qui modifie la structure de la table en ajoutant l'attribut datenaiss
5. Donner la requête permettant de :
 - ✓ Lister toutes les élèves de la table client
 - ✓ D'afficher les élèves de sexe masculin
 - ✓ D'afficher les filles donc la tranche d'âge est supérieur à 20 ans
6. Supprimer le champs datenaiss
7. Modifier votre âge
8. Afficher les prénoms de toutes les élèves toutes en éliminant les doublons.

I- DEFINITION DES DONNEES**PDF Compressor Free Version****A. CRÉATION ET SUPPRESSION D'UNE BASE DE DONNEE**

Pour créer une base de données qui sera appelé « ma_base » il suffit d'utiliser la requête suivante qui est très simple :

CREATE DATABASE ma_base ;

Pour supprimer la base de données « ma_base », la requête est la suivante :

DROP DATABASE ma_base ;

Attention: cela va supprimer toutes les tables et toutes les données de cette base.

B. CRÉATION DE TABLES

La commande **CREATE TABLE** permet de créer une table en SQL. Un tableau est une entité qui est contenu dans une base de données pour stocker des données ordonnées dans des colonnes. La création d'une table sert à définir les colonnes et le type de données qui seront contenus dans chacun des colonne (entier, chaîne de caractères, date, valeur binaire ...).

Syntaxe

La syntaxe générale pour créer une table est la suivante :

```
CREATE TABLE nom_de_la_table
(
    colonne1 type_donnees(n) contrainte,
    colonne2 type_donnees(n) contrainte,
    colonne3 type_donnees(n) contrainte,
    colonne4 type_donnees(n) contrainte
);
```

❖ Principaux types de données

CHAR(n) : Chaîne de caractères de longueur fixe

SMALLINT : Entier signé de 16 bits (-32768 à 32757)

INTEGER : Entier signé de 32 bits (-2E31 à 2E31-1)

DECIMAL(n,m) : Nombre de n chiffres [optionnellement m après la virgule]

DATE : Date sous la forme 16/07/99

VARCHAR(n): Chaîne de caractères de n caractères maximum

FLOAT/ : Nombre à virgule flottante

❖ **Contraintes d'intégrité**

NOT NULL empêche d'enregistrer une valeur nulle pour une colonne.

UNIQUE unicité d'un attribut

PRIMARY KEY indiquer si cette colonne est considérée comme clé primaire pour un index.

FOREIGN KEY clé étrangère

CHECK plage ou liste de valeurs

DEFAULT attribuer une valeur par défaut si aucune donnée n'est indiquée pour cette Colonne lors de l'ajout d'une ligne dans la table.

Une contrainte qui ne fait référence qu'à une seule colonne de la table peut faire partie intégrante de la définition de colonne

Exemple

Imaginons que l'on souhaite créer une table utilisateur, dans laquelle chaque ligne correspond à un utilisateur inscrit sur un site web. La requête pour créer cette table peut ressembler à ceci :

```
CREATE TABLE utilisateur
(  

  id INT PRIMARY KEY NOT NULL,  

nom VARCHAR(100),  

prenom VARCHAR(100),  

email VARCHAR(255),  

date_naissance DATE,  

pays VARCHAR(255),  

ville VARCHAR(255),  

code_postal VARCHAR(5),  

nombre_achat INT  

);
```

C. MODIFICATION DU SCHÉMA

La plupart des systèmes permettent la suppression ou la modification d'une table à l'aide des commandes :

DROP TABLE et **ALTER TABLE**

❖ **DROP TABLE**

La commande **DROP TABLE** en SQL permet de supprimer définitivement une table d'une base de données. Cela supprime en même temps les éventuels index, trigger, contraintes et permissions associées à cette table.

Attention : il faut utiliser cette commande avec attention car une fois supprimée, les données sont perdues. Avant de l'utiliser sur une base importante il peut être judicieux d'effectuer un backup (une sauvegarde) pour éviter les mauvaises surprises.

Syntaxe

Pour supprimer une table « nom_table » il suffit simplement d'utiliser la syntaxe suivante :

DROP TABLE nom_table ;

A savoir : s'il y a une dépendance avec une autre table, il est recommandé de les supprimer avant de supprimer la table. C'est le cas par exemple s'il y a des clés étrangères.

❖ ALTER TABLE

La commande ALTER TABLE en SQL permet de modifier une table existante. Il est ainsi possible d'ajouter une colonne, d'en supprimer une ou de modifier une colonne existante, par exemple pour changer le type.

Syntaxe de base

D'une manière générale, la commande s'utilise de la manière suivante :

ALTER TABLE nom_table

Instruction ;

Le mot-clé « instruction » ici sert à désigner une commande supplémentaire, qui sera détaillée ci-dessous selon l'action que l'on souhaite effectuer : ajouter, supprimer ou modifier une colonne

➤ Ajouter une colonne (ADD)

L'ajout d'une colonne dans une table est relativement simple et peut s'effectuer à l'aide d'une requête ressemblant à ceci :

ALTER TABLE nom_table

ADD nom_colonne type_donnees ;

Exemple

Pour ajouter une colonne qui correspond à une rue sur une table utilisateur, il est possible d'utiliser la requête suivante :

ALTER TABLE utilisateur

ADD adresse_rue VARCHAR(255) ;

➤ Supprimer une colonne (DROP COLUMN)

Une syntaxe permet également de supprimer une colonne pour une table. Il y a 2 manières totalement équivalentes pour supprimer une colonne :

ALTER TABLE nom_table

DROP nom_colonne ;

Ou (le résultat sera le même)

ALTER TABLE nom_table

DROP COLUMN nom_colonne ;

➤ Modifier une colonne(MODIFY)

Pour modifier une colonne, comme par exemple changer le type d'une colonne, il y a la syntaxe suivante :

```
ALTER TABLE nom_table  
MODIFY nom_colonne type_donnees ;
```

Ici, le mot-clé « type_donnees » est à remplacer par un type de données tel que INT, VARCHAR, TEXT, DATE ...

➤ Renommer une colonne

Pour renommer une colonne, il convient d'indiquer l'ancien nom de la colonne et le nouveau nom de celle-ci.

```
ALTER TABLE nom_table  
CHANGE colonne_ancien_nom colonne_nouveau_nom type_donnees(n) ;  
Où  
ALTER TABLE nom_table  
RENAME COLUMN colonne_ancien_nom TO colonne_nouveau_nom ;
```

II- MANIPULATION DES DONNEES

SELECT, INSERT, UPDATE et DELETE sont les 4 commandes de manipulation des données en SQL

A. LA COMMANDE SELECT(Recherche)

L'utilisation la plus courante de SQL consiste à lire des données issues de la base de données. Cela s'effectue grâce à la commande **SELECT**, qui retourne des enregistrements dans un tableau de résultat. Cette commande peut sélectionner une ou plusieurs colonnes d'une table.

L'utilisation basique de cette commande s'effectue de la manière suivante :

```
SELECT nom_du_champ  
FROM nom_du_tableau ;
```

Cette requête va sélectionner (**SELECT**) le champ « nom_du_champ » provenant (**FROM**) du tableau appelé « nom_du_tableau ».

- **Obtenir plusieurs colonnes**

Il suffit tout simplement de séparer les noms des champs souhaités par une virgule. Pour obtenir les prénoms et les noms des clients il faut alors faire la requête suivante :

```
SELECT prenom, nom  
FROM client;
```

- **Obtenir toutes les colonnes d'un tableau**

Il est possible de retourner automatiquement toutes les colonnes d'un tableau sans avoir à connaître le nom de toutes les colonnes. Au lieu de lister toutes les colonnes, il faut simplement utiliser le caractère « * » (étoile).

C'est un joker qui permet de sélectionner toutes les colonnes. Il s'utilise de la manière suivante :

SELECT * FROM client ;

Cette requête retourne exactement les mêmes colonnes qu'il y a dans la base de données.

▪ **SQL DISTINCT**

L'utilisation de la commande SELECT en SQL permet de lire toutes les données d'une ou plusieurs colonnes.

Cette commande peut potentiellement afficher des lignes en doubles. Pour éviter des redondances dans les résultats il faut simplement ajouter **DISTINCT** après le mot SELECT.

L'utilisation basique de cette commande consiste alors à effectuer la requête suivante :

SELECT DISTINCT ma_colonne

FROM nom_du_tableau ;

Cette requête sélectionne le champ « ma_colonne » de la table « nom_du_tableau » en évitant de retourner des doublons.

▪ **SQL WHERE**

La commande **WHERE** dans une requête SQL permet d'extraire les lignes d'une base de données qui respectent une condition. Cela permet d'obtenir uniquement les informations désirées.

Syntaxe

La commande WHERE s'utilise en complément à une requête utilisant SELECT. La façon la plus simple de l'utiliser est la suivante :

SELECT nom_colonnes

FROM nom_table

WHERE condition;

▪ **Opérateurs de comparaisons**

Une requête SQL peut être restreinte à l'aide de la condition WHERE. Les opérateurs logiques AND et OR peuvent être utilisés au sein de la commande WHERE pour combiner des conditions.

Syntaxe d'utilisation des opérateurs AND et OR

Les opérateurs sont à ajoutés dans la condition WHERE. Ils peuvent être combinés à l'infini pour filtrer les données comme souhaités. L'opérateur AND permet de s'assurer que la condition1 ET la condition2 sont vrai :

SELECT nom_colonnes

FROM nom_table

WHERE condition1 AND condition2;

L'opérateur OR vérifie quant à lui que la condition1 OU la condition2 est vrai :

SELECT nom_colonnes FROM nom_table

WHERE condition1 OR condition2;

Ces opérateurs peuvent être combinés à l'infini et mélangés. L'exemple ci-dessous filtre les résultats de la table « nom_table » si condition1 ET condition2 OU condition3 est vrai :

SELECT nom_colonnes FROM nom_table

WHERE condition1 AND (condition2 OR condition3);

Attention : il faut penser à utiliser des parenthèses lorsque c'est nécessaire. Cela permet d'éviter les erreurs car et ça améliore la lecture d'une requête par un humain.

Exemple de données

Pour illustrer les commandes, nous allons considérer la table « produit » suivante :

id	Nom	Catégorie	stock	prix
1	ordinateur	Informatique	5	950
2	Clavier	Informatique	32	35
3	souris	Informatique	16	30
4	crayon	Fourniture	147	2

Pour filtrer uniquement les produits informatiques qui sont presque en rupture de stock (moins de 20 produits disponible) il faut exécuter la requête suivante :

SELECT * FROM produit

WHERE (categorie = 'informatique') AND (stock < 20);

Pour filtrer les données pour avoir uniquement les données sur les produits « ordinateur » ou « clavier » il faut effectuer la recherche suivante :

SELECT * FROM produit

WHERE (nom = 'ordinateur') OR (nom = 'clavier');

▪ SQL ORDER BY

La commande ORDER BY permet de trier les lignes dans un résultat d'une requête SQL. Il est possible de trier les données sur une ou plusieurs colonnes, par ordre ascendant ou descendant.

Syntaxe

Une requête où l'on souhaite filtrer l'ordre des résultats utilise la commande ORDER BY de la sorte :

SELECT colonne1, colonne2 ...

FROM table

ORDER BY liste de colonnes [ASC | DESC] ;

L'ordre de tri est précisé par **ASC** (croissant) ou **DESC** (décroissant) ; par défaut **ASC**

B. ~~PDF Compressor Free Version~~ LA COMMANDE INSERT (Ajout)

La commande INSERT permet d'ajouter de nouvelles lignes à une table

INSERT INTO Nomtable (liste de colonnes) **VALUES** (liste de valeurs) ;

Dans le cas où la liste de colonnes n'est pas spécifiée tous les attributs de la table cible doivent être fournis dans l'ordre de déclaration

➤ Insertion d'une seule ligne

Ajouter le client ('c100', 'Duduche', 'Nice') dans la table client

INSERT INTO client (IdCli, nom, ville) **VALUES** ('c100', 'Duduche', 'Nice') ;

➤ Insertion de plusieurs lignes

Il est possible d'ajouter plusieurs lignes à un tableau avec une seule requête. Pour ce faire, il convient d'utiliser la syntaxe suivante :

INSERT INTO client (prenom, nom, ville, age)

VALUES

('Rébecca', 'Armand', 'Saint-Didier-des-Bois', 24),

('Aimée', 'Hebert', 'Marigny-le-Châtel', 36),

('Marielle', 'Ribeiro', 'Maillères', 27) ;

A noter : lorsque le champ à remplir est de type VARCHAR ou TEXT il faut indiquer le texte entre guillemet simple. En revanche, lorsque la colonne est un numérique tel que INT ou BIGINT il n'y a pas besoin d'utiliser de guillemet, il suffit juste d'indiquer le nombre

C. LA COMMANDE UPDATE (Mise à jour)

La commande UPDATE permet d'effectuer des modifications sur des lignes existantes. Très souvent cette commande est utilisée avec WHERE pour spécifier sur quelles lignes doivent porter la ou les modifications.

Syntaxe

La syntaxe basique d'une requête utilisant UPDATE est la suivante :

UPDATE table

SET nom_colonne_1 = 'nouvelle valeur'

WHERE condition ;

Cette syntaxe permet d'attribuer une nouvelle valeur à la colonne nom_colonne_1 pour les lignes qui respectent la condition stipulée avec WHERE. Il est aussi possible d'attribuer la même valeur à la colonne nom_colonne_1 pour toutes les lignes d'une table si la condition WHERE n'était pas utilisée.

D. LA COMMANDE DELETE(Suppression)

La commande DELETE en SQL permet de supprimer des lignes dans une table. En utilisant cette commande associée à WHERE il est possible de sélectionner les lignes concernées qui seront supprimées.

Syntaxe

La syntaxe pour supprimer des lignes est la suivante :

DELETE FROM table

WHERE condition ;

Attention : s'il n'y a pas de condition WHERE alors toutes les lignes seront supprimées et la table sera alors vide.

Jeu Bilingue :

Mots en français	Traduction en anglais
Base de données	DATA BASE
SQL	SQL
SGBD	DBMS
Requête	Request
Privilèges	Privileges
Contrainte d'intégrité	Integrityconstraint

Exercice 1 : soit la table étudiant suivante :

Nom	Prénom	Matricule	Sexe	Age	note
Mékong	David	15iut01	M	17	19.5
Tcheuffa	Auguste	15iut02	M	20	05
Tchuba	Lyse	15iut03	F	12	14
Njiké	Yann	15iut04	M	8	09

Ecrire les requêtes SQL qui permettent de :

- 1) Créer la table étudiant.
- 2) Afficher tous les garçons ayant plus de 16.
- 3) Modifier l'âge de Njiké car maintenant il a 14 ans
- 4) Supprimer tous les garçons ayant plus de 15 ans
- 5) Afficher les élèves par note ayant plus de 10
- 6) Insérer dans la table étudiant l'élève suivant : fotso, Séverin, 15iut20, M, 24, 15.5.

- 7) Ajouter la colonne date-naissance
- 8) Supprimer la colonne note

PDF Compressor Free Version

Exercice 2 :

Le questionnaire suivant a été proposé aux candidats à un emploi d'agent de maîtrise dans une entreprise qui souhaite mettre en place une division des systèmes d'informations.

Traiter les questions ainsi formulées fort des compétences acquises en classe et de vos connaissances sur les systèmes de gestion des bases de données.

1. Donner une définition du terme : requête.
2. "SQL est un langage de définition des données". Donner deux opérations de définition de données.
3. Citer deux caractéristiques d'un système de gestion des bases de données.
4. Considérons la table suivante nommée : élève

CodeE	Nom	Moyenne
E001	NONO	13
E002	KALLA	15
E003	NFOR Gu	16
E004	ONANA	12
E005	ADAMA	17

- a. Donner la requête SQL permettant d'afficher les élèves en ayant une moyenne supérieure ou égale à 10.
- b. Donner la requête SQL permettant de modifier le nom de KALLA. Ce nom s'écrit avec un seul L.

Correction des exercices

Exercice 1

- 1) Créer la table étudiant.
CREATE TABLE étudiant ;
- 2) Afficher tous les garçons ayant plus de 16.
SELECT*
FROM étudiant
WHERE (Sexe="M") AND (Age>16);
- 3) Modifier l'âge de Njiké car maintenant il a 14 ans
UPDATE étudiant
SET Age=14
WHERE Nom="Njiké;
- 4) Supprimer tous les garçons ayant plus de 15 ans
DELETE FROM étudiant
WHERE(Sexe= "M")AND(Age>15);

- 5) Afficher les élèves par note ayant plus de 10

SELECT*

FROM étudiant

WHERE note>=10

GROUP BY note;

- 6) Insérer dans la table étudiant l'élève suivant : fotso, Séverin, 15iut20, M, 24, 15.5.

INSERT INTO étudiant (Nom, Prenom, Matricule, Sexe, Age, Note) VALUES ("fotso ", "Séverin", "15iut20", "M", 24, 15.5) ;

- 7) Ajouter la colonne date-naissance

ALTER TABLE étudiant

ADD CULUMN date-naissance ;

- 8) Supprimer la colonne note

ALTER TABLE étudiant

DROP COLUMN note ;

Exercice2

1. Donner une définition du terme : requête.

Une requête est la sollicitation d'un service du serveur.

2. "SQL est un langage de définition des données". Donner deux opérations de définition de données.

Création d'une base de données

Création d'une table

Ajout d'une nouvelle colonne

Suppression d'une base de données

Suppression d'une table

Suppression d'une colonne

3. Citer deux caractéristiques d'un système de gestion des bases de données.

Rapidité des accès, non redondance, indépendance physique et logique, sécurité etc.

4. Donnez

- Donnons la requête SQL permettant d'afficher les élèves ayant une moyenne supérieure ou égale à 10 :

SELECT*

FROM eleve

WHERE (moyenne>=10);

- Donnons la requête SQL permettant de modifier le nom de KALLA. Ce nom s'écrit avec un seul L. .

UPDATE eleve

SET nom="KALA"

WHERE codeE="E002" ;

PDF Compressor Free Version**Module II : SYSTÈME D'INFORMATION ET BASES DE DONNEES****UNITE D'APPRENTISSAGE 6 : IMPLEMENTATION D'UNE BASE DE DONNEES****UE19 : CREATION D'UNE BASE DE DONNEES EN MODE GRAPHIQUE****Compétences visées :**

- ❖ démarrer l'application MS Access
- ❖ Créer une base de données et l'enregistrer
- ❖ Créer une table
- ❖ Insérer une clé primaire

Situation problème :

Votre papa est un propriétaire d'un collège d'enseignement secondaire et a organisé un recrutement d'enseignants pour l'année scolaire 2020/2021 et voudrais avoir un fichier dans lequel il pourra sauvegarder la liste de tous les enseignants candidats pour pouvoir les utiliser plus tard et demande une aide à son informaticien qui malheureusement n'est pas dans la ville mais demande à celui-ci de créer une base de données sur Access avec les champs **Nom ; Prénom ; Date de naissance ; Sexe ; Numéro CNI ; Matière enseignée**. Dans laquelle il devra enregistrer tous les candidats et de mettre son nom en premier. Mais Papa n'ayant aucune compétence requise pour mener à bien ce travail sollicite votre aide. Utilise tes connaissances acquises à l'école pour aider Papa.

Noms de champs	Types de données	Clé primaire	Taille du champ
Nom	Texte	Non	25
Prénom	Texte	Non	25
Sexe	Texte	Non	10
Date de naissance	Date	Non	
Numéro CNI	Numérique	oui	Entier long
Matière En	Texte	Non	15

Noms	Prénom	Sexe	Date de naissance	Numéro NIC	Matière En
Bivoula	Andrey	M	25/12/1980	1149987	Informatique

Consigne :

- Ouvre l'application Ms Access
- Créez une base de données nommée « **Candidats** »
- Créez une table nommée « **Enseignant** »

PDF Compressor Free Version

- Sachant que chaque candidat un numéro de carte d'identité unique, insérer **Numéro CNI** comme clé primaire
- Insérez le nom du professeur d'informatique
- Enregistrer votre travail et fermer

Introduction

ACCESS c'est une application qui fait partie du package Microsoft Office qui nous permet de créer et de manipuler des bases de données. Il en existe plusieurs versions (2000, 2003, 2017, 2010, 2013, 2016 et 2019). Une base de données Access est un fichier d'extension **.mdb** (mais porte l'extension **.accdb** depuis la version 2007). Dans cette leçon nous étudierons la version 2007 et son icône se présente sous la forme :



1. Démarrer Access sous Windows

Comme toutes les autres Application du package Microsoft Office, pour démarrer Access on peut procéder de l'une des manières suivantes :

- Double-cliquer sur son icône sur le bureau
- Cliquer sur le bouton Démarrer -> Tous les programmes -> Microsoft Office -> Microsoft Office Access 2007.
- Cliquer sur le bouton Démarrer -> Exécuter et saisir la commande « msaccess » et OK

2. Créer une base de données

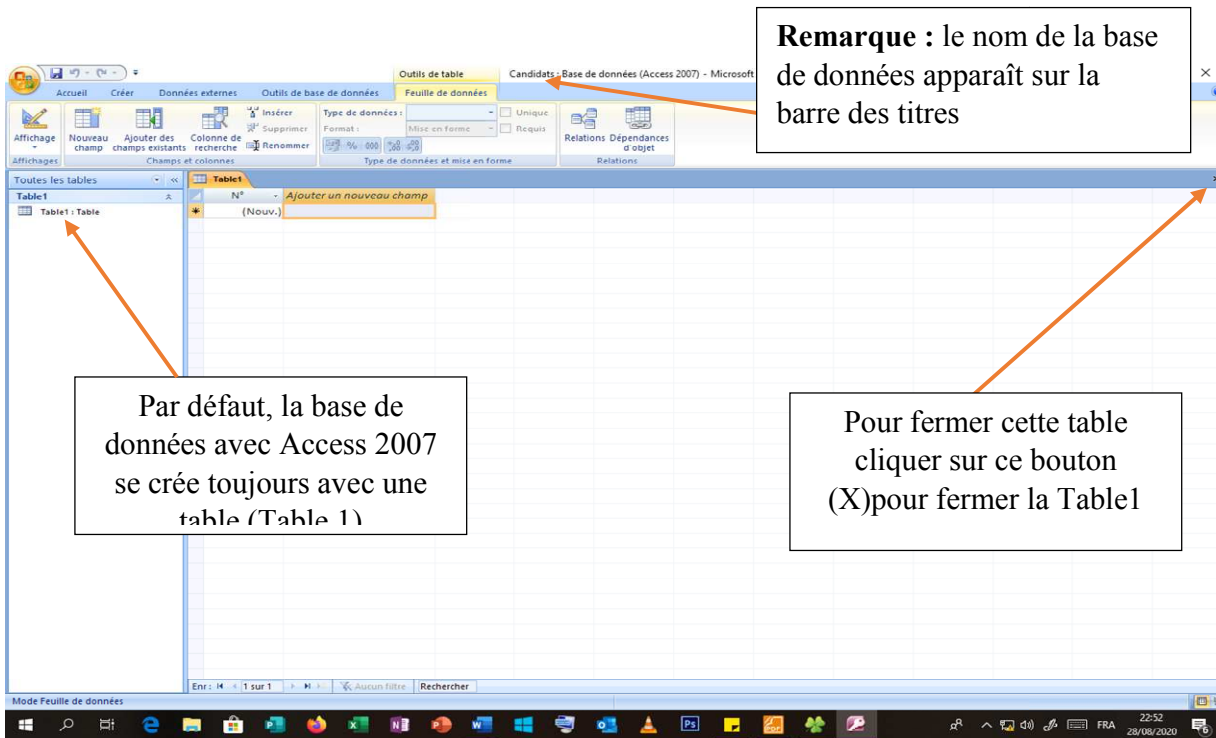
Une fois l'application Access démarrée, pour créer une base de données il faut :

1. Faire un clic sur « Base de données »

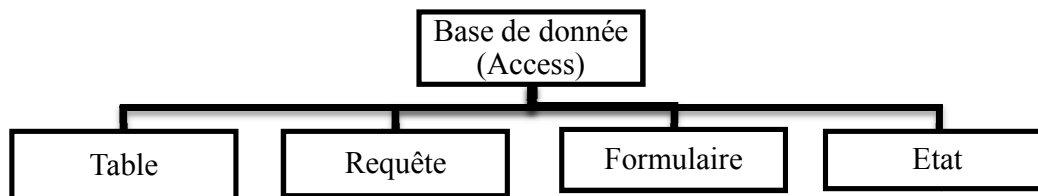
2. Saisir le nom de la base de données (Exemple : **Candidats**)

3. Cliquez sur « créer »

Une fois avoir cliqué sur « **créer** » voilà-ci-dessous votre base de données créer

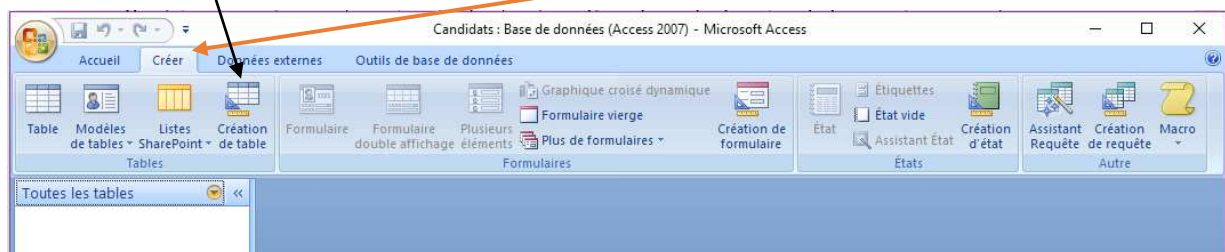


Il est à noter qu'on peut représenter la structure d'une base de données de la manière suivante



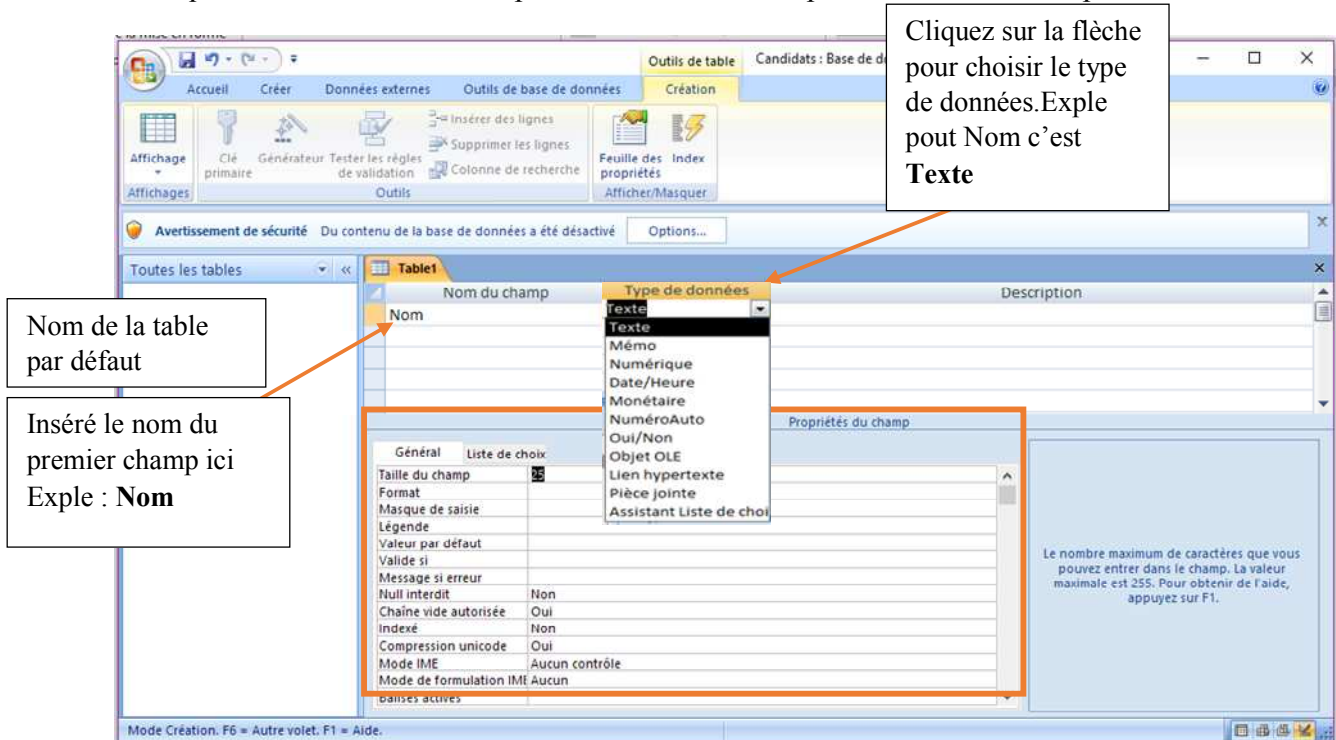
3. Création d'une table

Pour créer une table dans la base de données, on clique sur le menu « **Créer** » puis sur « **Création de table** »



PDF Compressor Free Version

Une fois que la table est créée vous pouvez maintenant remplir les différents champs de votre table



Remplir les propriétés correspondantes au champ créé : Exemple pour le champ **Nom** la taille du champ c'est 255

NB : Les propriétés de chaque champ vous sont données dans le tableau représentant le dictionnaire de données

Exercice : Créez les autres champs de la table « **Enseignant** » pour la base de données de votre papa

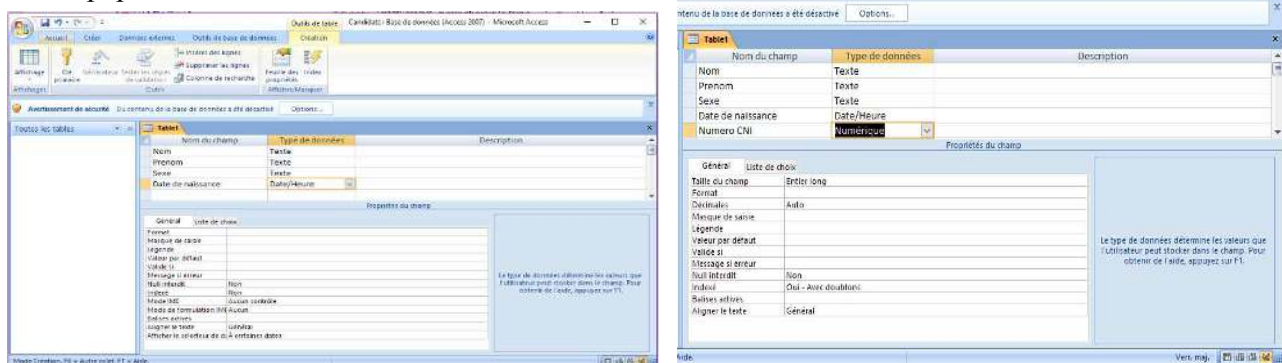
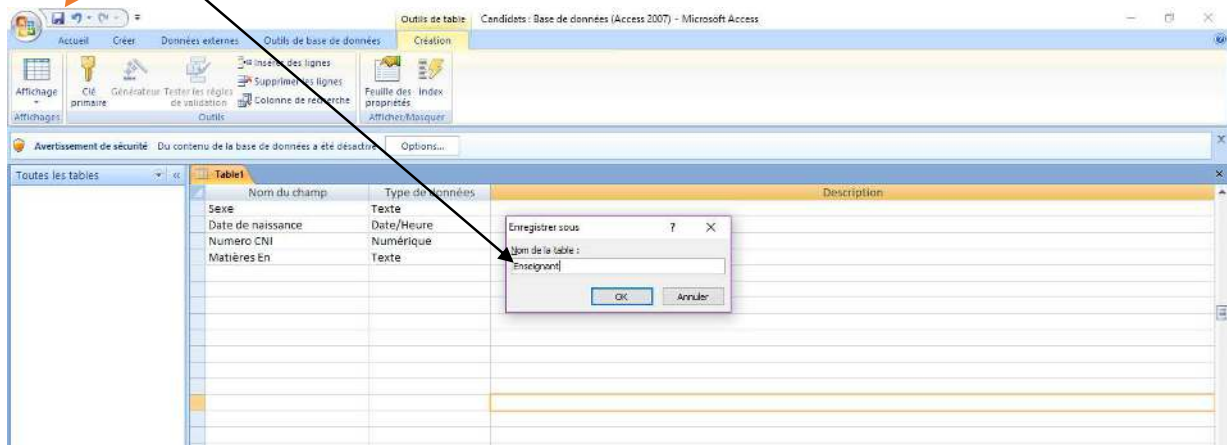


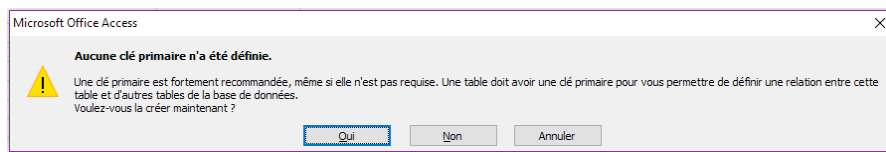
Table1	
Nom du champ	Type de données
Sexe	Texte
Date de naissance	Date/Heure
Numero CNI	Numérique
Matières En	Texte

PDF Compressor Free Version

Une fois que vous avez finis de remplir les champs des différentes tables, il faut cliquer sur le bouton **enregistré**. Dans la boîte de dialogue qui s'ouvre saisissez le nom de votre table (Exple Enseignant) et cliquez sur **OK**.



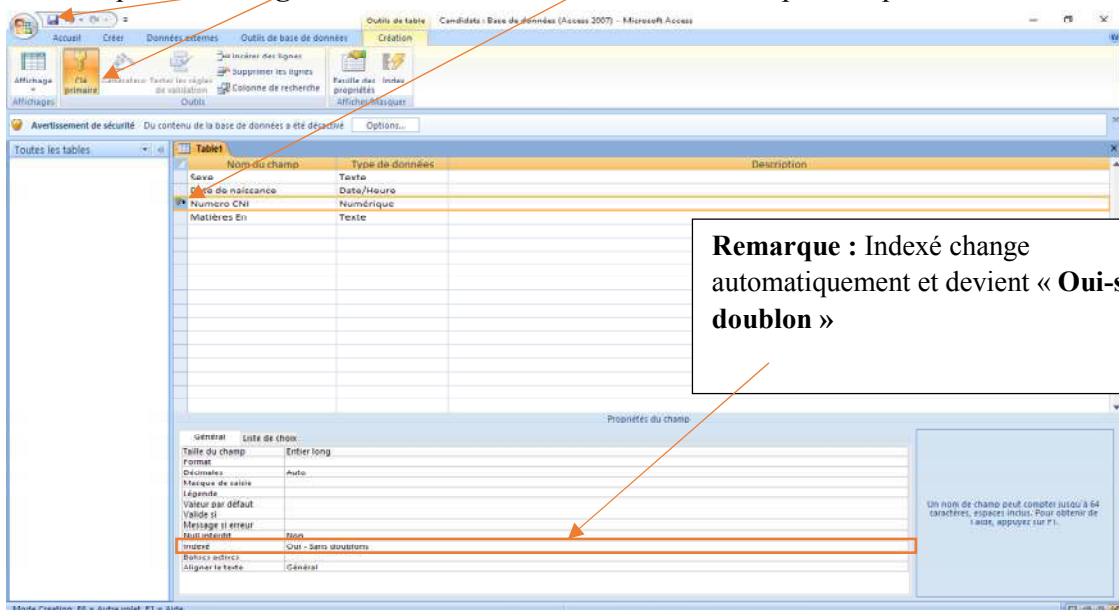
Remarque :



Lorsque vous cliquez sur le bouton « **OK** » une boîte de dialogue ci-dessus apparaît, c'est tout à fait normal puisque vous n'avez pas défini la clé primaire de votre table.

4. Création d'une Clé Primaire

Pour insérer une **clé primaire** il faut simplement cliquer sur le champ de la table à définir comme clé et ensuite cliquer sur « **clé primaire** », un symbole de **clé** apparaîtra devant la ligne de champ sélectionné. Cliquer sur **enregistré** et ressaisissez le nom de la table puis cliquez sur **OK**.

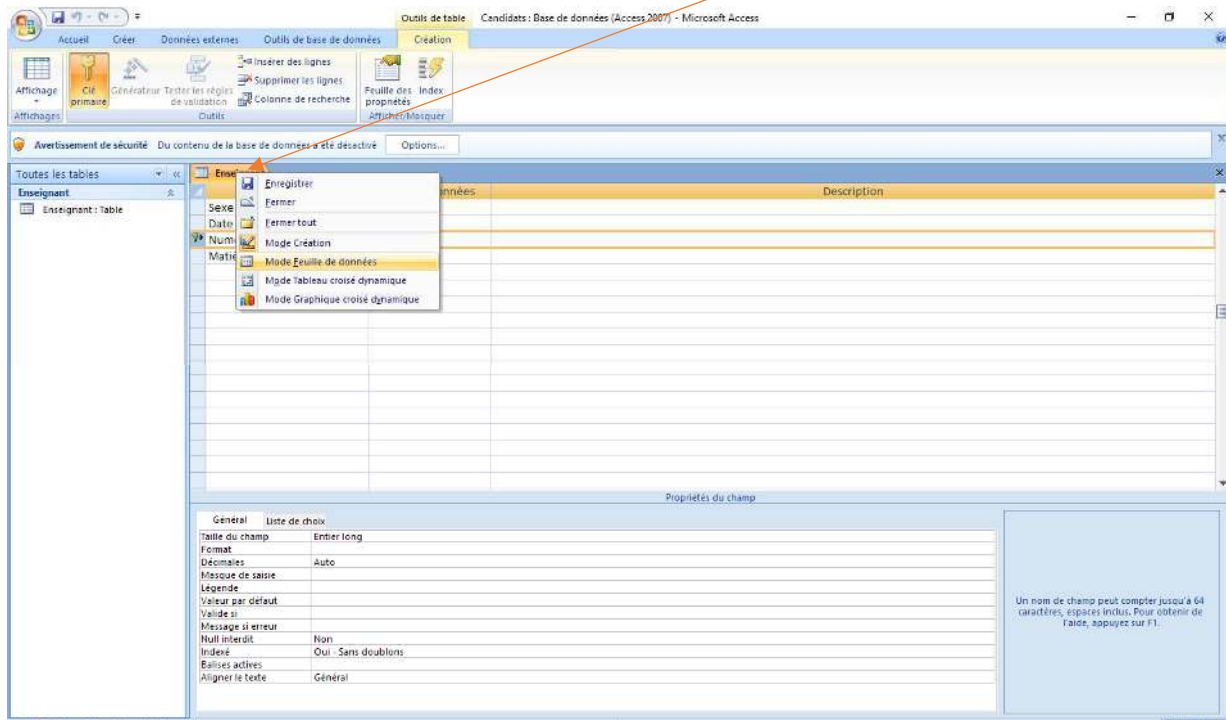


PDF Compressor Free Version

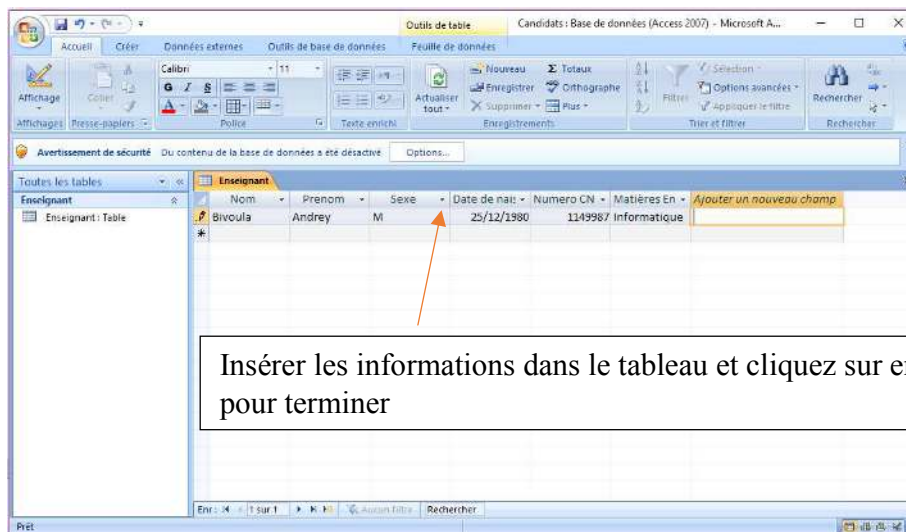
5. Effectuer un enregistrement dans une table

a) Changer le mode d'affichage

Pour effectuer un enregistrement dans une table, il faut d'abord changer le **mode d'affichage** de la table et choisir le **mode « Mode Feuille de données »**. Pour cela faire un **clic droit** sur le nom de la table et choisir le mode correspondant dans la liste qui apparaît.



b) Effectuer les enregistrements



Voilà donc votre base de données créée avec un tableau

PDF Compressor Free Version

Exercices

1. C'est quoi Ms Access ?
2. Peut-on créer une table sans avoir créé une table sur Access sans définir la clé primaire ?
3. Créez une base de données nommée «**Ecole** » et deux tables nommées «**Elève** » et «**Personnel** » en fonction du dictionnaire de données ci-dessous et faire quatre (04) enregistrements de votre choix dans chaque table.

Dictionnaire de données

champs	Clé primaire	Clé secondaire	Type de données	Taille	format	Décimale	Indexé
Matricule	Oui	Non	Texte	30			Oui sans double
Nom_Eleve	Non	Non	Texte	30			Non
Num_Eleve	Oui	Non	NuméroAuto				Non
Sexe_Eleve	Non	Non	Texte	7			Non
Classe_Eles	Non	Non	Numérique	Entier-long	Standard	0	Non
Nom_Ens	Non	Non	Texte	30			Non
Num_Ens	Oui	Non	NuméroAuto				Non
Mat_Ens	Oui	Non	Texte	30			Oui sans double
Matière_Ens	Non	Non	Texte	30			Non
Sexe_Ens	Non	Non	Texte	30			Non

MODULE 3 : ALGORITHME ET PROGRAMMATION**PDF Compressor Free Version****UA7 : ECRITURE ET EXECUTION D'UN ALGORITHME INTÉGRANT UNE STRUCTURE DE DONNÉES****UE 20 : LES STRUCTURES DE CONTRÔLE ALGORITHMIQUE****Compétence visées :**✓ *Décrire les structures de contrôle***Situation problème :**

Soit l'algorithme suivant qui affiche une décision sur la base de la moyenne obtenue au BEPC.

```

Algorithme : Resultat
Variable Moyenne:Reel;
Debut :
Ecrire("Veuillez saisir la moyenne");
  Lire(Moyenne);
Si(Moyenne>=10)alors
Ecrire("Admis au BEPC");
Finsi
  Si(Moyenne<10)alors
    Ecrire("Recalé au BEPC");
  Finsi
Fin

```

Consignes:

1. Identifier, puis relever dans l'algorithme le critère qui permet d'indiquer qu'un élève est admis au BEPC. Bonne réponse: (Moyenne >= 10)
2. Identifier, puis relever l'instruction qui est exécutée dans ce cas de figure
Bonne réponse : Ecrire ("Admis au BEPC");
3. Relever la portion de l'algorithme qui contient les éléments de la question 1 et de la question 2 :
Si (Moyenne >= 10) alors
Ecrire("Admis au BEPC");
Finsi

I. Définitions :

Algorithme : suite finie et ordonnée d'instructions permettant de résoudre un problème donné.

Instruction : c'est un ordre donné à l'ordinateur et qu'il exécute à la lettre.

Structure de contrôle : C'est une instruction particulière pouvant modifier l'ordre d'exécution des instructions d'un programme lorsqu'elle est présente.

Structure itérative : La structure itérative ou boucle répète l'exécution d'une instruction ou d'un ensemble d'instructions.

II- les structures algorithmiques

Il existe deux types de structure algorithmique : **Les structures de contrôle et les structures de données.**

II-1- Les structures de contrôles

On distingue trois grands types de structure de contrôle à savoir : **les structures itératives, les structures conditionnelles et les structures séquentielles.**

Nous allons nous étaler sur les structures itératives et conditionnelles.

II-1-1- Les structures itératives ou répétitive

Encore appelé **boucle**, elles sont au nombre de 3 :

- La boucle « Pour ...Faire... » :

Syntaxe (*Comment écrire cette boucle*) :

Pour (Valeur initiale jusqu'à valeur finale) faire

Instructions

Finpour

Remarque :

Celle boucle est utilisée lorsqu'on connaît le nombre d'itérations (de tour) à effectuer.

- La boucle « tant que ...faire... » :

Syntaxe :

Tantque (condition) faire

Instructions

FinTantque

Remarques :

→ La condition est analysée à l'entrée de la boucle, si elle n'est pas réalisée, la boucle ne sera jamais parcourue.

→ La boucle est parcourue tant que la condition est vraie.

→ On ne connaît pas dès le départ le nombre de fois où la boucle va être réalisée.

- La boucle « répéter... jusqu'à » :

Syntaxe :

Répéter **PDF Compressor Free Version**

Instructions

Jusqu'à (condition)

Remarques :

→ La condition est analysée à la fin de la boucle, la boucle sera parcourue au moins une fois même si la condition n'est pas réalisée.

→ La boucle est parcourue tant que la condition est fausse.

→ On ne connaît pas le nombre de fois où cette boucle sera réalisée.

Bon à savoir : Différence entre ces 3 boucles itératives

Boucle « pour »	Boucle « tantque »	Boucle « répéter »
Nombre d'itération connue	Nombre d'itérations inconnus	Nombre d'itérations inconnus
La boucle peut ne pas être parcourue	La boucle peut ne pas être parcourue	La boucle est parcourue au moins une fois
La condition est analysée à l'entrée	La condition est analysée à l'entrée	La condition est analysée à la fin de la boucle
La boucle sera parcourue tant que la condition est vrai	La boucle sera parcourue tant que la condition est vrai	La boucle sera parcourue tant que la condition est fausse

II-1-2- Les structure conditionnelles ou alternatives : Rappel

Elles permettent d'exécuter une portion de code lorsqu'une condition est respectée.

On a :

- La Boucle Si ... alors (structure conditionnelle simple)

Syntaxe :

Si (condition) alors

Instructions

Finsi

La structure **si....alors**

• La boucle **Si ... sinon ...alors** (structure conditionnelle complète)

Syntaxe :

Si (condition1) alors

Instructions

Sinon si (condition2) alors

Instructions

Sinon

Instructions

Finsi

Jeu bilingue :

Français	Anglais
Algorithme	Algorithm
Instruction	Instruction
pour	for
Tant que	While
Repeter jusqu'a	Repeat until

EXERCICES ET CORRIGES

Exercices :

- 1.Écrire un algorithme permettant de faire la somme des n premiers entier naturel, « n » étant le nombre entré par l'utilisateur.
- 2.Écrireun algorithme qui vérifie si le nombre entré par l'utilisateur est égale de 2020, si tel est le cas, il affiche « Bienvenue dans 2020 »
- 3.Pour aller de la ville Pour la campagne, un taxi porte plusieurs personnes au prix total de 6000frs par tour c'est-à-dire aller et retour. Mozo est un taximan faisant cette ligne, il souhaite gagner au 30 000frs. Ecrire un algorithme qui détermine le nombre de tour qu'il doit faire.

CORRIGES

1.

Algorithme : Somme_des_n_entiers

Var n, i, somme: entier

Début :

Somme \leftarrow 0

Ecrire ("Entrer un nombre entier n :")

Lire n

Pour i allant de 0 à n faire

Somme \leftarrow somme +1 ;

i \leftarrow i+1 ;

Finpour

Ecrire somme ;

Fin

2.

Algorithme : Annee_Actuelle

Var nombre : entier

Début :

nombre \leftarrow 0

Tantque (nombre # 2020) faire

Ecrire ("Entrer un nombre :")

Lire nombre ;

FintantQue

Ecrire "Bienvenue dans 2020" ;

Fin

3.

Algorithme : Nombre_de_tour

Var tour, gain :entier

Début :

tour \leftarrow 0 ;

gain \leftarrow 0 ;

Répéter

gain \leftarrow gain+6000 ;

tour \leftarrow tour+1 ;

jusqu'à ce que (gain=30000)

ecrire tour ;

Fin

MODULE 2 : ALGORITHMIQUE ET PROGRAMMATION

UA7: ECRITURE ET EXECUTION DES ALGO INTEGRANT STRUCTURES DE DONNEES

UE 21 : LES STRUCTURES DE DONNEES ET DECLARATION DE TABLEAU

Compétences visées :

- Caractériser une structure de données
- Créer un tableau en langage algorithmique

Situation problème :

Votre professeur désire concevoir un programme qui lui permette de saisir et calculer la moyenne générale de sa classe. Il voudrait en même temps que ce programme puisse stocker la moyenne de chaque élève pour une utilisation ultérieure. Quelle structure de données est la mieux indiquée pour implémenter ces tâches ?

I- DEFINITION

Une **structure de données** est une manière particulière de stocker et d'organiser des données pour les traiter plus facilement. Elle permet ainsi d' utiliser ces données de façon efficace. La plupart des bons algorithmes fonctionnent grâce à une méthode astucieuse pour organiser les données.

II- LES TYPES DE STRUCTURE DE DONNEES

Différents types de structures de données existent pour répondre à des problèmes très précis. Parmi celles-ci nous pouvons citer : les tableaux, les enregistrements, les listes chaînées, les piles et les files.

1. Les tableaux

Un tableau est une structure de donnée T qui permet de stocker un certain nombre d'éléments $T[i]$ repérés par un index i . Les tableaux sont des structures de données séquentielles. Les tableaux vérifient généralement les propriétés suivantes :

- tous les éléments ont le même type de base ;
- le nombre d'éléments stockés est fixé ;

12	05	8	16	14
----	----	---	----	----

Exemple de tableau à 5 éléments de type Entier

2. les enregistrements

Les enregistrements sont des structures de données dont les éléments peuvent être de type différent. Pour créer des enregistrements, il faut déclarer un nouveau type (type structuré), basé sur d'autres types existants.

Après avoir défini un type structuré, on peut l'utiliser comme un type normal en déclarant une ou plusieurs variables de ce type.

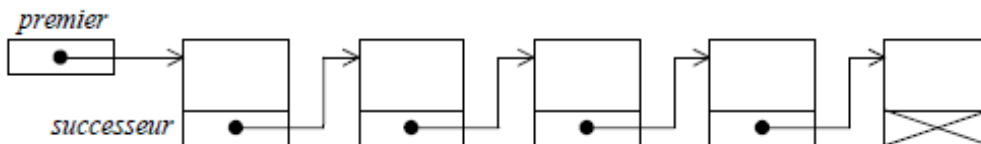
Les éléments qui composent un enregistrement sont appelés champs

Code	Nom	Prénom	Classe
0FG23	FANDIO	Samuel	Tle C
5GH34	DEFFO	Rita	TleD

Cette table quatre champs (Code, Nom, Prénom, Classe) et deux enregistrements

3. Listes chaînées

Une liste chaînée est une structure de données représentant une collection ordonnée et de taille arbitraire d'éléments de même type. Les listes chaînées sont représentées dans la mémoire de l'ordinateur par une succession de cellules faites d'un contenu et d'un pointeur vers une autre cellule



NB : les listes chaînées sont des structures linéaires

Contrairement aux tableaux et aux listes chaînées, les piles et les files se définissent par leur comportement (en répondant à la question « qu'est-ce que cela fait ») ; on dit que ce sont des structures de données abstraites.

4. Les piles

Une pile (stack en anglais) est une structure de données dynamique fondée sur le principe « dernier arrivé, premier sorti » (en anglais LIFO pour *Last In First Out*). Ceci veut dire en général que le dernier élément ajouté dans la pile est le premier à en sortir.

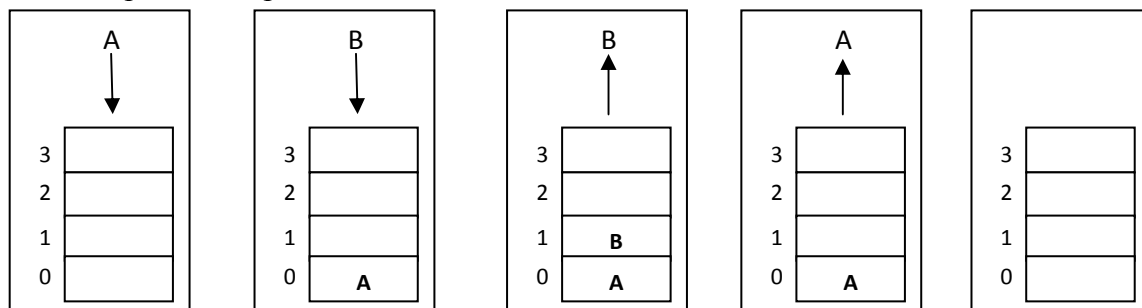


Schéma d'une pile gérée en LIFO

C'est une propriété abstraite, qui n'impose pas une constitution particulière. En fait, on réalise aussi bien des piles à partir de tableaux qu'à partir de listes chaînées.

5. Les files

Une *file* ou *file d'attente*, *queue*, est une structure de données dynamique basée sur le principe « premier entré, premier sorti » désigné en anglais par l'acronyme FIFO pour *First In First Out*. Ceci veut dire que les premiers éléments ajoutés à la file seront les premiers à en être retirés.

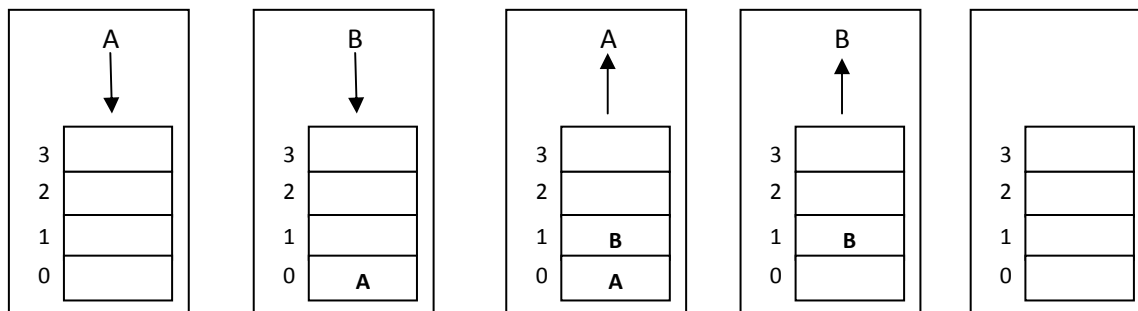


Schéma d' une file gérée en FIFO

III- DECLARATION D' UN TABLEAU

La déclaration d' un tableau peut se faire de plusieurs façons. Les plus utilisées sont les suivantes :

Variable nom_tableau : **Tableau** [taille] de type ;

Ou bien **Variable Tableau** nom_tableau [taille] : type ;

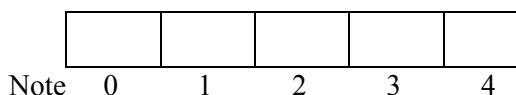
Ou bien **Variable** nom_tableau : **Tableau** [indice_min .. indice_max] de type ;

Rappelons que dans un tableau, chaque élément est repéré par son indice ; le premier élément peut avoir pour indice 0 ou 1 selon le langage utilisé.

Exemple : On désire déclarer un tableau nommé *Note* de 4 éléments de type réel

Variable Note : Tableau [5] de Réel ;

Résultat :



Dans cet exemple, le premier élément a pour indice 0 et le dernier 4

NB : ce tableau peut aussi être déclaré comme suit : *Variable Note : Tableau [1 .. 5] de Réel ;* dans ce cas, le premier élément aura pour indice 1 et le dernier 5

EXERCICE DE CONSOLIDATION

Exercice 1.

On considère les fonctions suivantes :

- *creerPile (P)*
- *empiler (P, "X")* permettant d'insérer l'élément X dans la pile P
- *depiler (P)* permettant d'enlever un élément dans la pile P

Donner le contenu de la pile après exécution des instructions suivantes :

Debut

creerPile (P);

empiler (P, "A");

empiler (P, "B");

depiler (P);

empiler (P, "C");

empiler (P, "D");

depiler (P);

PDF Compressor Free Version

Fin

Exercice 2 : reprendre l'exercice 1 en considérant la file F. on utilisera les fonctions `creerFile()`, `enfiler()` et `defiler()`.

MODULE 3 : ALGORITHMIQUE ET PROGRAMMATION**PDF Compressor Free Version****UNITE D'APPRENTISSAGE 7 : ECRITURE ET EXECUTION D'UN ALGORITHME
INTEGRANT UNE STRUCTURE DE DONNEES (6H)****Situation problème :**

Voici un tableau contenant les notes en informatique des élèves de la classe de Tle C.

11	9	8,5	12	12,5	10	16	7	6,5	5,5	13	9	11	11,5	8,5	14	14	7
----	---	-----	----	------	----	----	---	-----	-----	----	---	----	------	-----	----	----	---

Etant donné le nombre élevé d'élève, on désire écrire un algorithme permettant de rechercher la plus grande note de cette classe.

Consigne :

- 1- Identifier le nombre d'élément que comporte ce tableau
- 2- Expliquer la dont vous allez procéder pour retrouver la plus grande note de ce tableau.

Solution

- 1- Le tableau contient 18 éléments
- 2- Méthode pour trouver le plus grand élément : il faut comparer successivement les éléments du tableau en prenant le premier élément comme la plus grande note au départ

UE 22 : PARCOURS ET RECHERCHE D'UN ÉLÉMENT DANS UN TABLEAU**Compétences visées :**

- Parcourir un tableau pour effectuer la lecture et l'affichage
- Ecrire l'algorithme de recherche du maximum ou du minimum dans un tableau

I- PARCOURIR UN TABLEAU POUR EFFECTUER LA LECTURE ET L'AFFICHAGE**1- Principe du parcours**

Le principe est de « *boucler sur les indices (index)* » puisque'on connaît la longueur du tableau, on utilise souvent la boucle POUR.

2- Algorithme de Lecture et d'affichage dans un tableau

Exemple d'application : écrire un algorithme permettant de saisir 5 nombres entiers, les stocker dans un tableau puis les afficher à l'écran.

Solution :

Algorithme Lecture_affichage_tableau
Var

V : tableau [1..5] de entiers
i : entier;

Debut

//lecture des éléments du tableau
Pour i de 1 à 5 faire
 Ecrire('entrer l'élément N°', i);
 Lire(V[i]) ;

Finpour
//afficher des éléments du tableau
Pour i de 1 à 5 faire
 Ecrire(' l'élément N° ', i, 'est : ', V[i]);

Fimpour

fin

PDF Compressor Free Version

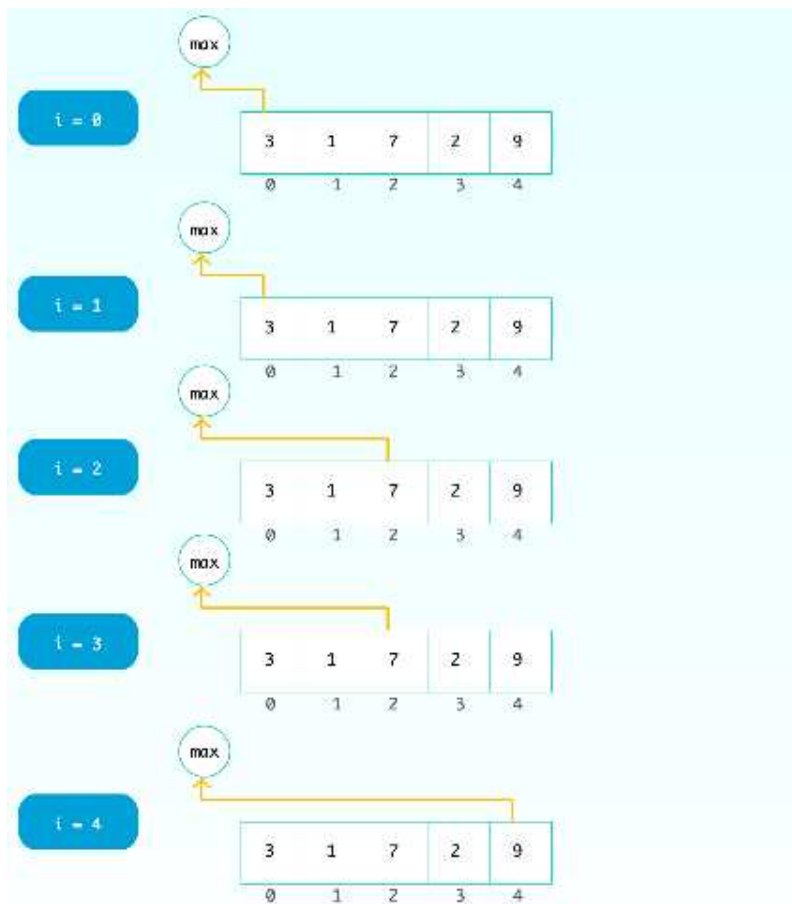
II- ALGORITHME DE RECHERCHE DU MAXIMUM DANS UN TABLEAU

1- Principe

La recherche s'effectue en considérant le premier élément du tableau étant le maximum (resp. minimum), et en parcourant le tableau itérativement, si à l'itération i la valeur se trouvant à l'indice i est supérieure (resp. inférieure) à la valeur considérée maximale (resp. minimale) au préalable, la valeur à l'indice i devient alors le maximum (resp. minimum).

L'exemple suivant détaille les différentes itérations de la recherche du maximum dans un tableau d'entiers:

La variable max prend initialement la valeur se trouvant à l'indice 0. Lorsque la valeur à l'indice i est plus grande que max , max devient alors la valeur à l'indice i .



2- Algorithme de recherche du maximum

On supposera que le tableau contient n éléments au préalable.

L'algorithme est le suivant :

Algorithme Recherche_Max

Var

$T: \text{tab}[1..n]: \text{entier};$

$\text{max}, i: \text{entier};$

Début

$\text{max} \leftarrow T[1];$

Pour i de 2 à n Faire
 Si (T[i] > max) Alors
 max ← T[i];
 FinSi
 FinPour
 Ecrire("Le max est: ", max);
 Fin

Jeu Bilingue :

Mots en français	Traduction en anglais
Algorithme	Algorithm

EXERCICES DE CONSOLIDATION

Exercice 1 : (voir cours)

Soit le tableau « Tab » suivant :

3	7	15	2	8	6
---	---	----	---	---	---

Ecrire l'algorithme qui recherche et affiche le minimum de ce tableau

Exercice 2 :

Ecrire un algorithme permettant d'ajouter un élément à la fin d'un tableau.

Exercice 3

Ecrire un algorithme qui déclare un tableau de 30 notes dont en fait ensuite saisir les valeurs par l'utilisateur

Exercice 4

Ecrire un algorithme permettant de saisir 5 réels au clavier, les stocker dans un tableau, calculer leur somme et les afficher avec leur somme à l'écran.

Exercice 5

Ecrire un algorithme permettant de saisir et d'afficher N éléments d'un tableau.

CORRECTION DES EXERCICES

<p>Exercice 2 Algorithme Ajout Var T : tableau [1..100] de réels ; N : entiers ; X : réel ; Début Ecrire("entrer la valeur de l'élément à ajouter :") ; Lire(X) ; N<-- N+1 ; <i>//il y aura un élément en plus</i> T[N] <-- X ; Finsi Fin</p>	<p>Exercice 3 Algorithme tableau_Notes Var N : tableau [1..30] de réels ; i : entier; Début <i>//saisit des valeurs par l'utilisateur</i> Pour i <-- 1 à 30 faire Ecrire('entrer la note N° ',i); Lire(N[i]) ; Fin pour fin</p>
---	---

Exercice 4

Algorithme tableau_somme

Var

V : tableau [1..5] de réels ;

S : réel ;

i : entier;

Début

//lecture des éléments du tableau

Pour i <-- 1 à 5 faire

Ecrire(“entrer l’élément N° ”,i);

Lire(V[i]) ;

Fin pour

//calcul de la somme des éléments du tableau

S <-- 0 ;

Pour i <-- 1 à 5 faire

S <-- S + V[i] ;

Fin pour

//afficher des éléments du tableau

Pour i <-- 1 à 5 faire

Ecrire(“l’élément N° ”,i, “est : “,V[i]);

Fin pour

Ecrire(“la somme des éléments du tableau est

:”,S) ;

fin

Exercice 5

Algorithme saisie_affichage

Var

T : tableau [1..100] de réels ;

N,i : réel ;

Début

Ecrire(“entrer le nombre d’éléments du tableau :”) ;

Lire(N) ;

// lecture des éléments du tableau

Pour i <-- 1 à N faire

Ecrire(“entrer l’élément N° ”,i);

Lire(T[i]) ;

Fin pour

//afficher des éléments du tableau

Pour i <-- 1 à N faire

Ecrire(“l’élément T[”,i,“] est : “,T[i]);

Fin pour

Fin

MODULE 3 : ALGORITHMIQUE ET PROGRAMMATION**PDF Compressor Free Version****U.A.7 : Écriture Et Exécution D'un Algorithme Intégrant Une Structure De Données****UE 23 : EXÉCUTION D'UN ALGORITHME DE RECHERCHE SÉQUENTIELLE DANS UN TABLEAU****Compétences visées**

- Décrire les méthodes de recherche dans un tableau;
- Exécuter un algorithme de recherche séquentielle ;

Prérequis

- Savoir déclarer un tableau ;
- Savoir effectuer les opérations simples dans un tableau (insertion, accès, recherche d'un élément);

Situation Problème

Étant donné un tableau contenant 10 notes dont la structure est la suivante :

12	12.5	10	9.75	8	17	13	11	5.5	9
----	------	----	------	---	----	----	----	-----	---

On désire écrire l'algorithme permettant de rechercher la note 17 et afficher vrai si on a retrouvé la note.

Consigne

Donnez le principe de recherche d'un élément dans un tableau.

Solution

Pour rechercher un élément dans un tableau, on saisit d'abord l'élément ensuite ; on utilise une structure itérative (boucle) et un indice pour parcourir le tableau jusqu'à ce qu'on trouve l'élément ; enfin quand l'élément est trouvé, on affiche un message.

Introduction

Il existe différentes méthodes algorithmiques plus ou moins futées pour chercher, dans un tableau, un élément dont la valeur est donnée. Parmi ces méthodes, nous avons la **recherche séquentielle** et la **recherche dichotomique**. Dans cette leçon, nous allons nous intéresser uniquement à la méthode de recherche séquentielle.

1. Recherche séquentielle

C'est la méthode de recherche la plus simple. On parcourt le tableau du début à la fin ou de la fin au début jusqu'à ce qu'on ait trouvé ce qu'on cherchait ou qu'on ait entièrement parcouru le tableau.

Écrivons l'algorithme correspondant à la situation de notre problème. On suppose que le tableau est déjà créé.

Algorithme Recherche_Note

Const n=10 ; //le nombre d'élément du tableau ou encore la taille du tableau

Var i : Entier ; // indice du parcours du tableau

ElemCherch : Réel ; //élément recherché dans le tableau

Trouve : Booléen ; // vrai si l'élément a été trouvé

Notes : Tableau [1..n] de Réel ; //déclaration du tableau

Début

// Lecture de l'élément recherché

Écrire ("entrez la note que vous recherchez") ;

Lire (ElemCherch) ;

//initiation des variables de contrôles

i ← 1 ;

Trouve ← faux ;

Tantque (i ≤ n) et (Trouve = faux) **Faire** //Parcours du tableau

Si Notes[i] = ElemCherch **Alors** //on a trouvé la note

Écrire ("vrai") ;

Trouve ← Vrai ;

Sinon

i ← i + 1 ;

Finsi

Fintantque

Fin

2. Exécution d'un algorithme de Recherche séquentielle

Pour exécuter un algorithme, il suffit de conserver une trace des valeurs en cours des différentes variables et d'exécuter une à une les opérations qui composent l'algorithme (en respectant la sémantique des structures de contrôle) en reportant leur éventuel impact sur les valeurs des différentes variables. Dans le cas présent, nous avons à faire à un algorithme de recherche séquentielle dans un tableau utilisant un structure itérative.

L'exécution de cet algorithme se présente sous la forme d'un tableau sachant que la valeur recherchée est 17.

Notes (Tableau contenant les notes)	ElemCherch (Élément recherché)	i (indice de parcours du tableau)	Notes[i] = ElemCherch (On vérifie si l'élément du tableau à l'indice i correspond à la valeur recherchée)		Trouve (Vrai si l'élément est trouvé)	Résultat
12	17	1	12=17	Faux	Faux	/
12,5	17	1+1=2	12,5=17	Faux	Faux	/
10	17	2+1=3	10=17	Faux	Faux	/
9,75	17	3+1=4	9,75=17	Faux	Faux	/
8	17	4+1=5	8=17	Faux	Faux	/
17	17	5+1=6	17=17	Vrai	Vrai	Vrai
13	/	/	/	/	/	/
11	/	/	/	/	/	/
5,5	/	/	/	/	/	/
9	/	/	/	/	/	/

3. La recherche dichotomique

Cette méthode est rapide car on ne consulte pas séquentiellement tous les éléments du tableau.

On s'appuie sur le fait que le tableau soit trié pour, au cours des itérations, évaluer avec de plus en plus de précision l'endroit où se trouve l'élément cherché.

La **dichotomie**, c'est le fait de couper en deux le tableau trié et de regarder dans quelle partie du tableau se trouve l'élément cherché.

On recoupe en deux cette partie du tableau et on regarde à nouveau dans quelle moitié l'élément peut se trouver.

Exercices de consolidation

Exercice 1

Soit le tableau suivant contenant six couleurs :

Rouge	Jaune	Noir	Bleu	Orange	Violet
-------	-------	------	------	--------	--------

Écrire un algorithme permettant de rechercher une couleur quelconque que l'utilisateur aura saisi et d'afficher le message "*cette couleur existe*" si la couleur est présente dans le tableau et donnez une trace de l'exécution de cet algorithme si la couleur recherchée est "*Orange*"

On utilisera les variables

- **Tcoul** : nom du tableau ;
- **couleur** : le nom de la couleur qu'on recherche ;
- **j** : l'indice qui parcourt le tableau

Exercice 2

Écrire un algorithme qui crée un tableau dans lequel l'utilisateur saisira 10 matières de son choix ; ensuite l'algorithme devra permettre à l'utilisateur de rechercher si une matière est présente dans le tableau après avoir saisi son nom.

Solutions

Exercice 1

Algorithme Recherche_Couleur

Const n=6 ; //le nombre d'élément du tableau ou encore la taille du tableau

Var j : Entier ; // indice du parcours du tableau

Couleur : Chaîne de caractère ; //couleur recherchée dans le tableau

Trouve : Booléen ; // vrai si on trouvé la couleur

Tcoul : Tableau [1..n] de Chaîne de caractère ; //déclaration du tableau

Début

// On suppose que le tableau a déjà été créé

// Lecture de l'élément recherché

Écrire ("entrez le nom de la couleur que vous recherchez") ;

Lire (Couleur) ;

//initiation des variables de contrôle

j ← 1 ;

Trouve ← Faux

Tantque(j ≤ n) et (Trouve = Faux) Faire //Parcours du tableau

Si Tcoul[j] = Couleur Alors //on a trouvé la couleur

Écrire ("Cette couleur existe") ;

Trouve ← Vrai ;

Sinon

$j \leftarrow i+1$;

Finsi **PDF Compressor Free Version**

Fintantque

Fin

Trace d'exécution de cet algorithme

Tcoul (Tableau contenant la couleur)	Couleur (couleur recherchée)	j (indice de parcours du tableau)	Tcoul[j] = Couleur (On vérifie si l'élément du tableau à l'indice j correspond à la valeur recherchée)		Trouve (Vrai si l'élément est trouvé)	Résultat
Rouge	Orange	1	Rouge = Orange	Faux	Faux	/
Jaune	Orange	1+1=2	Jaune = Orange	Faux	Faux	/
Noir	Orange	2+1=3	Noir = Orange	Faux	Faux	/
Bleu	Orange	3+1=4	Bleu = Orange	Faux	Faux	/
Orange	Orange	4+1=5	Orange = Orange	Vrai	Vrai	Cette couleur existe
Violet	Orange	/	/	/	/	/

Exercice 2

Algorithme Recherche_Matière

Const n=10 ; //le nombre de matière du tableau ou encore la taille du tableau

Vari : Entier ; // indice du parcours du tableau

Matiere : Chaîne de caractère ; //nom de la matière recherchée dans le tableau

Trouve : Booléen ; // vrai si on trouvé la matière

Tmat : Tableau [1..n] de Chaîne de caractère ; //déclaration du tableau

Début

// Création du tableau

Pour i allant de 1 à 10 **Faire**

Écrire ('entrez la matière numéro', i) ;

Lire (Tmat[i]) ;

FinPour

// Recherche de la matière

// Lecture de l'élément recherché

Écrire ('entrez le nom de la matière que vous recherchez') ;

Lire (Matiere) ;

//initiation des variables de contrôle

$i \leftarrow 1$;

Trouve \leftarrow Faux

Tantque($i \leq n$) et (Trouve=Faux) **Faire**//Parcours du tableau

Si Tmat[i]=Matiere **Alors**//on a trouvé la matière

Écrire ('Cette matiere existe) ;

Trouve \leftarrow Vrai ;

Sinon

$i \leftarrow i+1$;

Finsi

Fintantque

Fin

MODULE 3 : ALGORITHMIQUE ET PROGRAMMATION**PDF Compressor Free Version**
UNITE D'APPRENTISSAGE 8 : PROGRAMMATION EN LANGAGE C**UE 24 : NOTIONS DE BASES DE LA PROGRAMMATION EN C****Compétences visées**

- Citer quelques langages de programmation ;
- Citer quelques exemples d'IDE ;
- Installer un IDE ;
- Écrire la structure d'un programme C ;

Prérequis

- Connaitre la structure d'un algorithme
- Savoir exécuter un algorithme

Situation Problème

Vous avez écrit un algorithme permettant de résoudre une équation de second degré dans R. Votre enseignant d'informatique vous informe que vous pouvez traduire cet algorithme dans un langage de programmation dans votre machine de telle sorte à l'exécuter à chaque fois que vous le voudrez juste en entrant les valeurs. Il vous informe qu'il existe plusieurs langages de programmation parmi lesquels le C et que c'est ce langage que vous allez utiliser.

Questions et réponses:

1. Définir langage de programmation ? (*un langage de programmation est un langage utilisé en informatique pour écrire des programmes*)
2. Citer les outils nécessaires pour programmer en C (*pour programmer en C, il faut installer un IDE*)
3. Citer deux exemples de langages de programmation (*C++, C#, python, PERL, java..*)

INTRODUCTION

Programmer signifie réaliser des « programmes informatiques ». Les programmes demandent à l'ordinateur d'effectuer des actions. Cet ordinateur est rempli de programmes en tous genres : la calculatrice est un programme ; votre traitement de texte est un programme ; votre logiciel de « chat » est un programme ; les jeux vidéo sont des programmes. Il existe de nombreux langages de programmation plus ou moins de haut niveau en informatique dans lesquels on peut écrire des programmes. En voici quelques-uns par exemple : le C ; le C++ ; Java ; Visual Basic ; Delphi. Dans cette leçon, nous allons étudier les notions élémentaires pour la programmation en C.

I. NOTION DE LANGAGE DE PROGRAMMATION

1. Définitions

Un **langage de programmation** est ensemble de règles syntaxiques permettant l'écriture des programmes.

Un **programme** est ensemble d'instructions permettant à un ordinateur de résoudre une ou plusieurs tâches.

Une **instruction** est un ordre élémentaire interprétée et exécuté par le processeur

2. Types de langage de programmation

Il existe deux types de langage de programmation : les langages procéduraux et les langages orientés objets.

Un **langage procédural** est un langage dans lequel les programmes sont essentiellement composés de procédures. Un **langage orienté objet** est langage dans lequel les composants du programme sont des objets.

3. Langage Compilé / langage interprété

Dans un langage compilé, le compilateur traduit le programme entier une fois avant son exécution. Exemples : C, C++, Java, Pascal,...

Dans un langage interprété, l'interpréteur traduit au fur et à mesure les instructions du programme à chaque exécution. Exemples : PHP, Python, Shell, JavaScript

II. HISTORIQUE ET CARACTÉRISTIQUES DU LANGAGE C

1. Historique

Le langage C a été conçu en 1972 dans «Bell Laboratories » par *Dennis Ritchie* avec l'objectif d'écrire un système d'exploitation (UNIX). En 1978, une première définition rigoureuse du langage C (*standard K&R-C*) a été réalisée par *Kernighan et Ritchie* en publiant le livre «The C Programming Language ».

2. Caractéristiques

- *Universel* : n'est pas orienté vers un domaine d'application particulier (applications scientifiques, de gestion, ...);
- *Près de la machine* : offre des opérateurs qui sont proches de ceux du langage machine (manipulations de bits, d'adresses, ...);
- *Modulaire*: peut-être découpé en modules qui peuvent être compilés séparément ;
- *Portable*: il est possible d'utiliser le même programme sur plusieurs systèmes (hardware, système d'exploitation).

III. NOTION D'IDE

De manière générale, pour programmer, on a besoin de trois éléments :

- un éditeur de texte pour écrire le code source du programme ;
- un compilateur pour transformer (« compiler ») votre code source en binaire ;
- un débogueur pour vous aider à traquer les erreurs dans votre programme.

Ces trois éléments peuvent être regroupés dans un seul programme appelé IDE (Integrated Development Environment) ou Environnement de Développement intégré en français. Les IDE les plus connus sont : Dev-C++ ; Code ::Blocks. ; Visual C++. ; Visual Studio Express.

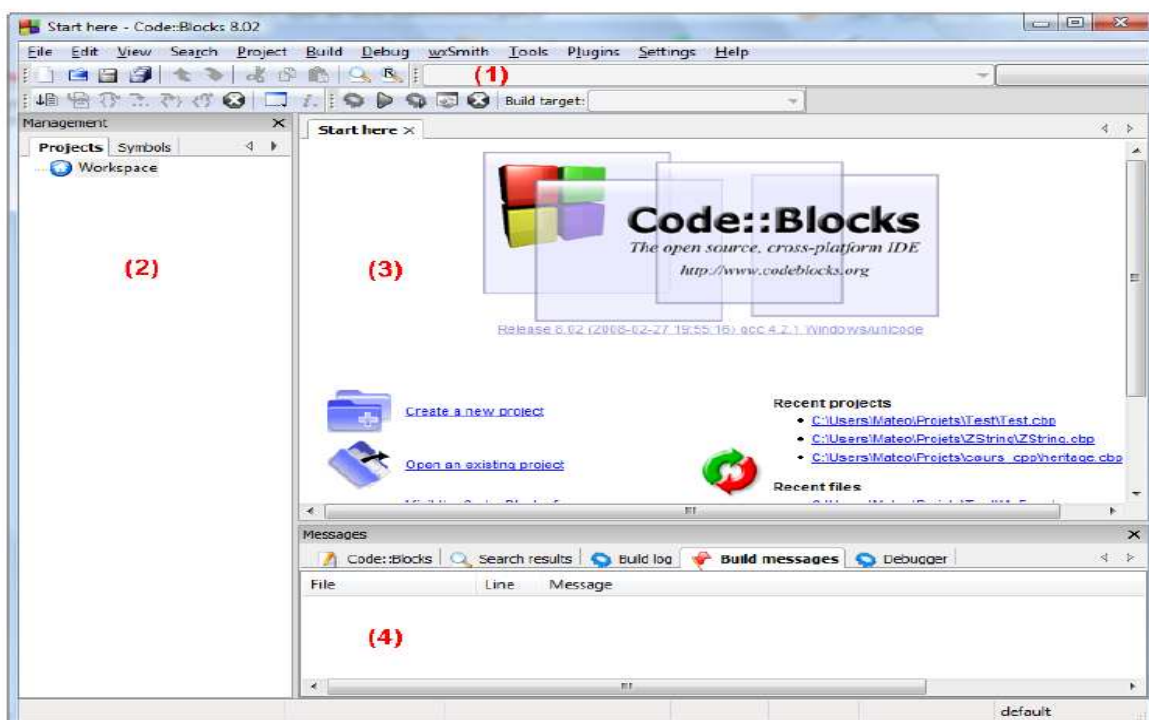
Certains IDE intègrent leur propre compilateur, il suffit de les cocher au moment de leur installation ; d'autres par contre doivent être installés indépendamment.

IV. INSTALLER UN IDE CAS DE CODE ::BLOCKS

Code::Blocks est un IDE libre et gratuit, disponible pour Windows, Mac et Linux et téléchargeable à partir du lien <http://www.codeblocks.org/downloads/binaires>

- Si vous êtes sous Windows, repérez la section « Windows » un peu plus bas sur cette page. Téléchargez le logiciel en prenant le programme qui contient mingw dans le nom (ex. : *codeblocks-10.05mingwsetup.exe*). L'autre version étant sans compilateur, vous auriez eu du mal à compiler vos programmes !
- Si vous êtes sous Linux, choisissez le package qui correspond à votre distribution.
- Enfin, sous Mac, choisissez le fichier le plus récent de la liste. Ex. : *codeblocks-10.05-p2-mac.zip*

L'installation est très simple et rapide. Laissez toutes les options par défaut et lancez le programme. Vous devriez voir une fenêtre similaire à la fig. suivante.



On distingue 4 grandes sections dans la fenêtre, numérotées sur l'image :

1. **la barre d'outils** : elle comprend de nombreux boutons,
2. **la liste des fichiers du projet** : c'est à gauche que s'affiche la liste de tous les fichiers source de votre programme;
3. **la zone principale** : c'est là que vous pourrez écrire votre code en langage C ;

4. la zone de notification : aussi appelée la « zone de la mort », c'est ici que vous verrez les erreurs de compilation s'afficher si votre code comporte des erreurs. Cela arrive très régulièrement !

PDF Compressor Free Version

V. STRUCTURE D'UN PROGRAMME EN C

Un programme C se présente de la façon suivante :

[directives au préprocesseur]

[déclarations de variables externes]

[fonctions secondaires]

Int main()

{ déclarations de variables internes,
instructions

}

La ligne : `int main()` se nomme un "en-tête". Elle précise que ce qui sera décrit à sa suite est en fait le "programme principal". Elle peut avoir des paramètres formels.

Le programme (principal) proprement dit est constitué des variables internes et des instructions et vient à la suite de cet en-tête. Il est délimité par les accolades "{" et "}".

CONCLUSION

Le langage C est un langage de programmation de haut niveau ; il permet de créer des applications exécutables et des applications fenêtre. La programmation en C nécessite l'installation d'un IDE.

Jeu bilingue

Français	Anglais
Programme	Program
Programmeur	programmer
Langage de programmation	Programming language

Exercice

1. Quelle différence faites-vous entre un algorithme et un programme ?
2. Citer trois exemples de langage de programmation que vous connaissez.
3. Que signifie IDE ? donnez en deux exemples
4. Répondre par vrai ou faux :
 - a) Le C est un langage de programmation interprété
 - b) Le C est un langage portable c'est-à-dire qu'on peut l'utiliser sur plusieurs systèmes.
 - c) Le noyau du système d'exploitation UNIX est écrit en C

Correction

1. *Un algorithme est écrit dans un langage naturel proche du langage humain alors qu'un programme est la traduction d'un algorithme dans un langage de programmation ?*
2. *Trois exemples de langage de programmation : C, java, PHP, Delphi*
3. *IDE signifie : Integrated Development Environment. Deux exemples : DevC++, Code::Blocks*
4. Répondre par vrai ou faux :
 - a) *Faux*
 - b) *Vrai*
 - c) *Vrai*

MODULE 3 : ALGORITHME ET PROGRAMMATION**UNITE D'APPRENTISSAGE 8 : PROGRAMMATION EN LANGAGE C**

PDF Compressor Free Version

Compétences à développer :

- Programmation en mode console

UE 25 : LES INSTRUCTIONS DE BASES EN C**Objectifs pédagogiques :**

- Structure d'un programme en C ;
- Connaître et savoir utiliser les directives du préprocesseur ;
- Pouvoir inclure et utiliser les bibliothèques standards ;
- Savoir déclarer des variables et produire des instructions de base en C.

Contrôle de prérequis :

- 1- Savoir écrire des algorithmes pour résoudre un problème ;
- 2- Savoir définir programmation et programme ;
- 3- Savoir définir les mots langages compilés, compilateur et compilation.

SITUATION PROBLEME :

Votre ami, après avoir entendu parler du C, décide de traduire son algorithme en langage C. Mais après la première compilation, il se trouve avec plusieurs erreurs. Ne comprenant rien sur ce qui se passe, il fait donc appel à vous dans le but de l'expliquer quelques notions sur langage C.

Consignes :

- 1) Définir langage de programmation (**Réponse** : ensemble des mots et symboles permettant d'écrire un programme).
- 2) Définir programme (**Réponse** : algorithme déjà traduit en langage de programmation).
- 3) Comment appelle-t-on l'application qui permet d'exécuter un programme sur un ordinateur ? (**Réponse** : compilateur) ;
- 4) Donner la structure du programme écrit en langage C. (**Réponse** :
[Directives au préprocesseur]
[Déclarations de variables externes]
[Fonctions secondaires]
int main ()
{ Déclarations de variables internes ;
instructions ;
}
)
- 5) Donner le rôle des bibliothèques en langage C. Puis énumérer quelques exemples (**Réponse** : Nous offrir des possibilités en termes d'objets et fonctions permettant de résoudre des problèmes complexes *exemple* `stdio.h`).
- 6) Définir instruction (c'est un ordre élémentaire interprétée et exécuté par le processeur)

I. INTRODUCTION :

Comme tous langage de programmation, le C obéit à des règles syntaxiques strictes.

II. STRUCTURE GÉNÉRALE D'UN PROGRAMME C

a) Définition :

Une expression : est une suite de composants élémentaires syntaxiquement correcte. Elle est toujours suivie d'un point-virgule (;)

En C on n'a pas une structure syntaxique englobant tout, comme la construction « **Algorithme ... Fin.** ». Un programme n'est qu'une collection de fonctions assortie d'un ensemble de variables globales. Ainsi un programme C est structuré comme suit :

[Directives au préprocesseur]

[Déclarations de variables externes]

[Fonctions secondaires]

int main ()

{

Déclarations de variables internes ;

Instructions ;

}

La ligne : **int main()** se nomme un "**en-tête**". Elle précise que ce qui sera décrit à sa suite : C'est le "programme principal". Elle peut avoir des paramètres formels (qui seront placés dans les parenthèses qui suivent le **main** !).

NB : La fonction main() est le seul bloc obligatoire d'un programme C.

Le corps du programme (principal) proprement dit est constitué des variables internes et des instructions et vient à la suite de cet en-tête. Il est délimité par les accolades "{" et "}".

Exemple de programme C affichant "Bonjour" :

```

1  #include <stdio.h>
2
3
4  int main(void)
5  {
6      printf("Bonjour tout le monde !\n" ) ;
7      return 0;
8  }
```

La première ligne de notre programme : `#include <stdio.h>` est en fait un peu particulière. Il s'agit d'une "directive" qui est prise en compte avant la traduction (compilation) du programme.

Ces directives, contrairement au reste du programme, doivent être écrites à raison d'une par ligne et elles doivent obligatoirement commencer en début de ligne. Leur emplacement au sein du programme n'est soumis à aucune contrainte (mais une directive ne s'applique qu'à la partie du programme qui lui succède). D'une manière générale, il est préférable de les placer au début.

La directive demande en fait d'introduire (avant compilation) des instructions (en langage C) situées dans le fichier `stdio.h`. Notez qu'un même fichier en-tête contient des déclarations relatives à plusieurs fonctions. En général, il est indispensable d'incorporer `stdio.h` (car toutes les fonctions nécessaires à l'affichage à l'écran et lectures au clavier y sont définies). Dans notre exemple, l'instruction d'appel `#include` signifie "ajouter le contenu du fichier `stdio.h`".

Quelques bibliothèques (ou directives de préprocesseur) utilisés en langage C sont :

Bibliothèques	Rôle
<code><stdio.h></code>	Fournit la capacités centrales d'entrées/sorties
<code><math.h></code>	Pour calculer des fonctions mathématiques
<code><stdlib.h></code>	Pour exécuter les opérations de conversion, l'allocation des mémoires, le contrôle de processus, le tri, la recherche, ...
<code><string.h></code>	Pour manipuler les chaines de caractères

NB : cette liste des bibliothèques standard C n'est pas exhaustive.

III. NOTION D'INSTRUCTIONS, VARIABLES ET OPERATEURS

a) Les instructions :

Les instructions constituent le travail réalisé par le programme. Elles permettent d'afficher les informations sur l'écran, de lire les données saisies au clavier, d'effectuer les opérations mathématiques, d'appeler les fonctions, de lire les fichiers et d'accomplir tous types d'opérations nécessaires à un programme. Chaque instruction occupe généralement une ligne et se termine par un point-virgule.

Les blocs d'instructions (ensemble d'instructions) commencent par une accolade ouvrante « { » et se terminent par une accolade fermante « } ».

b) Variable :

Une variable est un objet manipulé dans l'exécution d'un programme. En réalité, une variable est un nom donné à une zone mémoire. Elle est caractérisée par son **identificateur** (nom), son **type** et parfois sa valeur (le cas de constante qui est une variable dont le contenu ne changera jamais !).

Les identificateurs commencent par une lettre ou le caractère souligné ("_") et peuvent contenir des lettres, des chiffres et le caractère souligné (cependant, les identificateurs commençant par deux caractères soulignés, ou un caractère souligné suivi d'une majuscule sont réservés par l'implémentation, et ne doivent pas être utilisés dans un programme « ordinaire »). Tous les mots-clés ainsi que les symboles (variables, fonctions, champs, etc.) sont sensibles à la casse des lettres (un mot en majuscule est différent du même mot en minuscule)

Exemple de quelques mots **ne pouvant pas** être utiliser comme variable :

PDF Compressor Free Version

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
inline	short	int	long	register	restrict	signed	return
sizeof	static	struct	switch	typedef	union	unsigned	void

Quelques types prédéfinis en langages C sont :

- *int* (entier naturel) **NB** : parfois on utilise **short** pour les petits entiers, **long** pour les grands et **long long** pour les plus grand !
- *Char* (caractères) ;
- *float* , *double*, *long double* (nombre à virgule) ;
- *void* qui représente rien, le vide.

NB :

- Une variable doit être déclarée avant son utilisation.
- Une variable est dite locale, si elle est définie à l'intérieur d'une fonction (ne sont visibles que dans le bloc dans lequel elles sont définies) et globale si définie en-dehors.

En C, toute instruction composée d'un spécificateur de type et d'une liste d'identificateurs séparés par une virgule est une déclaration. Ainsi en C, une variable est déclarée de la manière suivante :

type identificateur ; ou bien

type identificateur1, identificateur2, identificateur4, ... ;

- Exemples : `int a ;` et `char a ;` ; `float a,b,c,d ;` sont de déclarations
- **const** `int i = 0 ;` permet de déclarer une constante ayant pour nom `i`, et pour valeur définitive 0.

Initialiser une variable c'est de lui affecter une valeur pour la première fois.

Exemple : `i=1 ; /* ceci est un commentaire !!!! Juste pour dire que j'ai initialiser i à 1 */`

Une constante littérale est une valeur qui est introduite directement dans le code source.

Voici deux exemples :

`int count = 20;`

`float tax_rate = 0.28;`

Ici, 20 et 0.28 sont des constantes littérales.

c) Opérateur

Le tableau ci-dessous résume quelques opérateurs utilisés en langage C :

Opérateurs	Symboles	Opérateurs	Symboles
Affectation	=	Égal	==
Reste de la division entière (modulo)	%	Différent	!=
Multiplication	*	ET logique	&&
Inférieur ou égal	<=	OU logique	
Addition	+	Soustraction	-
Division	/	Supérieur ou égal	>=

IV. LES ENTRÉES/SORTIES EN C

BDE Compressor Free Version

Durant l'exécution d'un programme, le processeur, qui est le cerveau de l'ordinateur, a besoin de communiquer avec le reste du matériel ou avec l'utilisateur. Il doit en effet recevoir des informations pour réaliser des actions et il doit aussi en transmettre. Ces échanges d'informations sont appelés les entrées et les sorties (ou input / output en anglais).

a) Les sorties

Le tableau ci-dessous décrit les trois fonctions d'affichage de données :

Fonctions	Rôle
Printf()	Pour écrire une chaîne de caractères formatée
puts ()	Pour écrire une chaîne de caractères toute simple
Putchar()	Pour écrire un caractère

La syntaxe de l'utilisation de ces fonctions est :

fonction ("texte_ a_afficher... ") ;

Exemples :

```
printf("Bonjour le monde");
puts("Hello world !") ;
put("c");
```

La fonction **Printf** permet non seulement d'afficher des chaînes de caractères simples, mais également la valeur d'une variable passée en paramètre. Pour ce faire, il suffit d'utiliser un indicateur de conversion : il s'agit du caractère spécial % suivi d'une lettre qui varie en fonction du type de la variable.

Le tableau ci-dessous donne quelques indicateurs de conversion :

PDF Compressor Free Version

Type	Indicateurs de conversions
char	%c
int	%d
long	%ld
short	%hd
float	%f
double	%f
long double	%Lf
unsigned int	%u
unsigned short	%hu
unsigned long	%lu

Après avoir inscrit un indicateur de conversion dans la chaîne de caractère (dans les guillemets ""), il faut indiquer de quelle variable il faut afficher la valeur. Il suffit de rajouter une virgule après les ces derniers, suivis du nom de la variable, comme ceci :

```
printf ("% [lettre]", variable_ a_afficher);
```

Exemples :

```

1  #include <stdio.h>
2
3
4  int main(void)
5  {
6      int variable = 20;
7
8      printf("%d\n", variable);
9      return 0;
10 }
```

Remarque : Plutôt qu'appeler plusieurs fois la fonction printf pour écrire du texte, on peut ne l'appeler qu'une fois et écrire plusieurs lignes. Pour cela, on utilise le signe \n à chaque fin de ligne.

b) Les entrées

Pour récupérer les valeurs saisies par l'utilisateur, nous utilisons les fonctions suivantes : scanf () et get ()

La syntaxe de l'utilisation de la fonction Scanf() est :

scanf("%[lettre]", &variable qui_contiendra_notre_valeur);

NB : La fonction scanf a besoin de connaître l'emplacement en mémoire de nos variables afin de les modifier. Afin d'effectuer cette opération, on utilise le symbole &. Donc, si vous oubliez le &, le programme plantera quand vous le lancerez, car vous tenterez d'accéder à une adresse mémoire inexistante !

Exemple :

```

1  #include <stdio.h>
2
3
4  int main(void)
5  {
6      int age;
7
8      printf("Quel âge avez-vous ? ");
9      scanf("%d", &age);
10     printf("Vous avez %d an(s)\n", age);
11     return 0;
12 }
```

Exercice 1 :

Vous souhaitez écrire un programme C qui demande à l'utilisateur de saisir un nombre puis affiche ce nombre.

1. Donner la structure d'un programme C
2. Quelles sont les bibliothèques standard que vous avez besoin dans votre programme C
3. Écrire l'algorithme qui permet de réaliser cette tâche.
4. Traduire votre algorithme en langage C

Exercice 2 :

1. Définir le terme suivant : compilateur, programme, programmation, langage de programmation.
2. Citer deux exemples de compilateur C
3. Donner deux exemples de langage de programmation.
4. Donner le rôle des bibliothèques standard suivant : <stdio.h>, <stdlib.h> et <math.h>
5. Donner 03 exemples des fonctions C utilisées pour afficher les textes à l'écran.
6. Donner deux exemples des fonctions C qui permet la récupération des données de l'utilisateur.
7. Donner la différence entre les fonctions puts() et printf()
8. Quel rôle joue le symbole & dans la récupération de la valeur d'une variable ?

Exercice 3 :

Écrire un programme C qui affiche « bonjour » suivi du nom de l'utilisateur.

Exercice 4 :

Votre petit frère qui apprend la programmation vient d'écrire le programme suivant :

```
1 #include <math.h>
2 int main() {
3 int a ;
4 put("entrer un nombre");
5 scanf("%d,a) ;
6 printf("le nombre saisi est:",a);
7 return 0 ;
8 )
```

Travail demandé :

Ce programme est-il correct ? Si non relevez toutes les lignes incorrectes puis corriger.

MODULE 3 : ALGORITHME ET PROGRAMMATION**UNITE D'APPRENTISSAGE 8 : PROGRAMMATION EN LANGAGE C****UE 26 : STRUCTURES DE CONTROLES EN C****Compétences visées :**

- Ecrire les syntaxes des structures ci-après : if(){}, for(){}, while(){}, et do {}while() ;

Situation problème 1 :

Soit le programme suivant qui affiche une décision sur la base de la moyenne obtenue au BACCALAUREAT.

```
#include<stdio.h>
intmain(void)
{
Float moyenne= 12,50;
if(moyenne<10)
{
printf("recaler au Baccalaureat\n");
}else{
printf("admis au Baccalauréat\n");
}
return0;
}
```

Consignes 1 :

1. Identifier, puis relever dans le programme le critère qui permet d'indiquer qu'un élève est admis au Baccalauréat.(réponse:(**Moyenne>=10**))
2. Identifier, puis relever l'instruction qui est exécutée dans ce cas de figure. (réponse : **printf("admis au Baccalauréat\n");**)
3. Relever la portion du programme qui contient les éléments de la question 1 et de la question 2. (Réponse

```
else{
printf("admis au Baccalauréat\n");
}
)
```
4. Identifier, puis relever dans le programme la structure de contrôle utilisée.(Réponse :
Ils'agitdelastructure alternative complète **if avec else**.

Situation problème 2 :

Soit le programme suivantquiaffiche le message Hello world ! 2 fois.

```
#include<stdio.h>
intmain(void)
{
inti = 0;

while (i<2)
{
printf(" Hello world! \n");
i++;
}
return0;
}
```

Consignes 2:

1. Identifier, puis relever dans le programme la condition qui doit être réalisée pour que l'instruction **printf** (" Hello world! \n"); s'exécute la 1ère fois. (réponse:(**i = 0**)c'est-à-dire (**0 <2**))
2. Identifier, puis relever dans le programme la condition qui doit être réalisée pour que l'instruction **printf** (" Hello world! \n"); s'exécute la 2 ème fois. (réponse : (**i = 1**)c'est-à-dire (**1 < 2**))
3. Identifier, puis relever dans le programme la structure de contrôle utilisée. (Réponse :

Ils'agit de la structure répétitive while

```
while (i<2)
{
printf(" Hello world! \n");
i++;
}
)
```

Trace écrite

Notion de structure de contrôle

Les structures de contrôle définissent la suite dans laquelle les instructions sont effectuées. On distingue comme structures de contrôle : conditionnelle ou alternative(if(){}), itérative ou boucle (for(){} , while(){} , et do {}while()) en langage C.

I- La structure alternative

1. La structure alternative réduite (if sans else qui signifie « si sans sinon »))

Permet d'exécuter une liste d'instructions sur la base d'une condition préalablement fixée.

Syntaxe :

La syntaxe de la structure if sans else est la suivante.

```
if (/* Condition */)
{
/* instruction ou Bloc d'instructions */
}
```

Exemple :

```
#include<stdio.h>
int main(void)
{
inti=-5;

if(i<0)
{
printf("ce nombre est négatif\n");
}
return 0;
}
```

Résultat : ce nombre est négatif

2. La structure alternative complète(if avec else qui signifie « si avec sinon »))

Syntaxe : La condition est testée et la liste d'instructions 1 est exécutée lorsque cette condition est vraie.

Dans le cas contraire, c'est plutôt la liste d'instructions 2 qui est exécutée.

La syntaxe de la structure if avec else est la suivante.

```
if (/* Condition */)
{
/* instruction 1 ou Bloc d'instructions 1*/
}else{
```



```
/* instruction 2 ou Bloc d'instructions 2*/
}
```

Exemple : PDF Compressor Free Version

```
#include<stdio.h>
intmain(void)
{
    int i=7;

    if(i <0)
    {
printf("ce nombre est négatif\n");
    }else{
printf("ce nombre est positif\n");
    }
    return0;
}
```

Résultat : ce nombre est positif

3. La structure alternative imbriquée(if – else if - ...- elsequi signifie « *si-sinon si-...-sinon* »))

En combinant plusieurs structures `if-else` en une expression nous obtenons une structure qui est très courante pour prendre des décisions entre plusieurs alternatives.

Syntaxe :La syntaxe de la structure `if - else if - ...- else`est la suivante.

```
if (/* Condition 1 */)
{
/* instruction 1 ou Bloc d'instructions 1*/
}elseif (/* Condition 2 */)
{
/* instruction 2 ou Bloc d'instructions 2*/
}elseif (/* Condition N */)
{
/* instruction N ou Bloc d'instructions N*/
} else {
/* instruction n+1 ou Bloc d'instructions n+1*/
}
```

Exemple :

```
#include<stdio.h>
intmain(void)
{
    int x, y;
    printf("Entrer deux nombres entiers\n");
    scanf("%d %d",&x,&y) ;

    if(x > y)
    {
printf("%d est plus grand que %d\n",x,y);
    }else if( x < y){
printf("%d est plus petit que %d\n",x,y);
    }else{
printf("%d est égal à %d\n",x,y);
    }
    return0;
}
```

Résultat : pour `x=2` et `y= 3`, ce programme affiche 2 est plus petit que 3.

II- Notion de boucle

Une boucle est un moyen de répéter des instructions suivant le résultat d’une condition. Ces structures, dites **itératives**, que nous allons voir sont les suivantes.

Structure itérative	Action
while	Répète une suite d'instructions tant qu'une condition est respectée.
do... while...	Répète une suite d'instructions tant qu'une condition est respectée. Le groupe d'instructions est exécuté au moins une fois.
for...	Répète un nombre fixé de fois une suite d'instructions.

1. La boucle **while**(qui signifie « *tant que* »)

La première des boucles que nous allons étudier est la boucle `while` (qui signifie « tant que »). Celle-ci permet de répéter un bloc d'instructions tant qu'une condition est remplie.

Syntaxe :

La syntaxe de notre boucle `while` est assez simple.

```
while (/* Condition */)
{
    /* Bloc d'instructions à répéter */
}
```

Exemple :

```
#include<stdio.h>
intmain(void)
{
    inti = 0;
    while (i<4)
    {
        printf("La variable i vaut %d\n", i);
        i++;
    }
    return0;
}
```

Résultat :

```
La variable i vaut 0
La variable i vaut 1
La variable i vaut 2
La variable i vaut 3
```

Le fonctionnement est simple à comprendre :

- Au départ, notre variable `i` vaut zéro. Étant donné que zéro est bien inférieur à quatre, la condition est vraie, le corps de la boucle est donc exécuté.
- La valeur de `i` est affichée.
- `i` est augmentée d'une unité et vaut désormais un.
- La condition de la boucle est de nouveau vérifiée.

Ces étapes vont ainsi se répéter pour les valeurs un, deux et trois. Quand la variable `i` vaudra quatre, la condition sera fausse, et l'instruction `while` sera alors passée.

2. La boucle **do-while**(qui signifie « *faire-tantque* »).

La boucle `do while` fonctionne comme la boucle `while`, à un petit détail près : elle s'exécutera toujours au moins une fois, alors qu'une boucle `while` peut ne pas s'exécuter si la condition est fausse dès le départ.

Syntaxe

À la différence de la boucle `while`, la condition est placée à la fin du bloc d'instruction à répéter, ce qui explique pourquoi celui-ci est toujours exécuté au moins une fois. Remarquez également la présence d'un point-virgule à la fin de l'instruction qui est obligatoire.

```
do
{
/* Bloc d'instructions à répéter */
} while (/* Condition */);
```

Exemple

Voici le même code que celui présenté avec l'instruction `while`.

```
#include<stdio.h>
intmain(void)
{
inti = 0;
do
{
printf("La variable i vaut %d\n", i);
++i;
} while (i <3);
return0;
}
```

Résultat :

```
La variable i vaut 0
La variable i vaut 1
La variable i vaut 2
```

3. La boucle `for` (qui signifie « *Pour* »)

Syntaxe

```
for (/* Expression/Déclaration */; /* Condition */; /* Expression */)
{
/* Instructions à répéter */
}
```

Une boucle `for` se décompose en trois parties (ou trois clauses) :

- une expression et/ou une déclaration qui sera le plus souvent l'initialisation d'une variable ;
- une condition ;
- une seconde expression, qui consistera le plus souvent en l'incrémentation d'une variable.

Techniquement, une boucle `for` revient en fait à écrire ceci.

```
/* Expression/Déclaration */
while (/* Condition */)
{
/* Bloc d'instructions à répéter */
/* Expression */
}
```

Exemple

```
#include<stdio.h>
intmain(void)
{
for (inti = 0; i<3; ++i)
printf("la variable i vaut %d\n", i);
return0;
}
```

Résultat

```
variable vaut 0
variable vaut 1
variable vaut 2
```

Jeu Bilingue :

PDF Compressor Free Version

Mots en français	Traduction en anglais
EDI (Environnement de Développement Intégré)	IDE (Integrated development environment)
Programme	Program
Structure de contrôle	Control structure

Exercice 1 :

Ecrire un programme qui détermine les solutions réelles d'une équation de second degré en utilisant les fonctions de la bibliothèque math.h(power(q, n) pour q^n et sqrt(n) pour racine carrée de n).

Résolution exercice 1 :

```
#include<stdio.h>
#include<math.h>
intmain(void)
{
    inta,b,c;
    double D ;
    printf("Entrer les valeurs de a , b et c pour ax^2 +bx +c=0 \n");
    scanf("%d %d %d ",&a,&b,&c) ;

    /*calcul du discriminant */
    D=power(b,2) - (4.0*a*c);

    /* Les diferentes cas*/
    if(a==0 && b==0 && c==0 )
    {
        printf(" Tout réel est solution de cette équation \n");
    }else if ( a==0 && b==0) /* c # 0 et c==0 */
    {
        printf("cette équation de possède pas de solution \n");
    }else if ( a==0) /* bx + c=0 */
    {
        printf("La solution de cette équation de 1er degré est x=%.2f\n",c/b);
    }else if ( D< 0) /* b^2 - 4ac < 0 */
    {
        printf("Cette équation n'a pas de solution réelle \n");
    } else if ( D = 0) /* b^2 - 4ac = 0 */
    {
        printf("Cette équation a une seule solution réelle x1=x2=%.2f\n", -b/(2*A));
    }else /* b^2 - 4ac > 0 */
    {
        printf("Les solutions réelles sont x1=%.2f:x1=x2=%.2f\n",(-b- sqrt(D)/(2*a*c), (-b+
sqrt(D)/(2*a*c)));

    }

    return0;
}
```

Exercice 2 :

Soit le programme suivant :

```
#include<stdio.h>
intmain(void)
{
    inti=0,n, som= 0;
    do
    {
        printf(" Entrer un nombre entier\n");
```

```
scanf("%d",&n) ;
som= som + n;
i=i+1;
} while (i<4);
```

PDF Compressor Free Version

```
return 0;
}
```

Ecrire un programme C réalisant exactement la même chose, en utilisant :

- La structure for ;
- La structure while

Resolution exercise 2 :

a) Structure for

```
#include<stdio.h>
int main(void)
{
    int i,n, som= 0;

    for(i = 0; i<4;i++)
    {
        printf(" Entrer un nombre entier\n");
        scanf("%d",&n) ;
        som= som + n;
    }
    return 0;
}
```

b) Structure while

```
#include<stdio.h>
int main(void)
{
    int i=0,n, som= 0;

    while(i<4)
    {
        printf(" Entrer un nombre entier\n");
        scanf("%d",&n) ;
        som= som + n;
        i=i+1;
    }
    return 0;
}
```

Exercice 3 :

Soit le programme ci-dessous

```
#include<stdio.h>
#include<stdlib.h>
int main(void){
    int n,i;
    float a,f;
    i=1; f=1;
    while(i<=n)
    {
        f=f*a;
        i=i+1;
    }
    printf("le résultat est \"%f\":, f  \n") ;

    return 0;
}
```

- Quelle est la structure de contrôle utilisée ? Préciser la condition de sortie
- Pour **a=4** et **n=5**, exécuter le programme pas à pas dans un tableau et donner les valeurs de f et i à la fin du programme.
- Dire ce que fait le programme.
- Réécrire le programme en utilisant la structure for

Solution :

- Structure **while(...)**. La condition de sortie est **i>n**
-

i	F	Condition

PDF Compressor Free Version

1	1	1<=5 vrai
1+1=2	1*4	2<=5 vrai
2+1=3	1*4*4	3<=5 vrai
3+1=4	1*4*4*4	4<=5 vrai
4+1=5	1*4*4*4*4	5<=5 vrai
5+1=6	1*4*4*4*4*4	6<=5 faux

A la fin du programme **i=6** et **f=20**

3. Le programme calcule a puissance n
- 4.

```
#include<stdio.h>
#include<stdlib.h>
int main(void){
    int n,i;
    float a,f;
    f=1;
    for(i=1;i<=n; i++)
    {
        f=f*a;
    }
    printf("le résultat est \"%f\":, f  \n") ;

    return 0;
}
```

Exercice4 :

Ecrire un programme qui invite l'utilisateur à saisir un nombre calcul et affiche la valeur absolue de ce nombre.

Solution

```
#include<stdio.h>
#include<stdlib.h>

int main(void){

    float nbre,resultat;

    printf("saisir le nombre \n") ;
    scanf(("%f",&nbre);
    if(nbre>0)
    {
        resultat=nbre;
    }
    Else
    {
        Resultat=-1*nbre;
    }

    printf("la valeur absolue de \"%f\" est: \"%f\", nbre, resultat) ;

    return 0;
}
```

Exercice3 :

Ecrire un programme qui invite l'utilisateur à saisir successivement les notes d'un élève et les coefficients en une séquence puis calcule et affiche la moyenne de cet élève.

PDF Compressor Free Version

FIN DU LIVRE

TERMINALES C-D-E

TRES GRAND RESPECTS ET REMERCIEMENTS AUX COLLEGUES DE DIVERS HORIZONS DU
PAYS QUI ONT PRIS PART A CETTE RICHE EXPERIENCE.(les admins)

VOS REMARQUES ET SUGGESTIONS POUR L'AMELIORATION SONT ATTENDUES:

- 678157860
- 675388456
- 679359583
- 651803226
- 697355373