



INFORMATIQUE TERMINALE **GD**

LES GRANDPROFS  INFO



DOCUMENT
DESTINÉ AUX

- ENSEIGNANTS
- ÉLÈVES

COURS DÉTAILLÉS



EXERCICES



JEUX BILINGUES



RÉDIGÉ PAR

 LES GRANDPROFS  INFO

SUPERVISÉ PAR

 KOUOKAM KAMGUE CYRILLE

Les auteurs

ABDOUL AZIZ NDJOBDI
ABOULAOU YAFET
ADAWAL Urbain
AMOUGOU François
BOUKAR
FIA Evarice Prosper
MOHAMMAD Aly
NKONDA Nyamsi
NJINKEU Jean Jules
OUAFO Péguy
OUTSÉ Bélibi Félicité

PELEU Anicet
PEUPIE Guy Bertrand
Servais NGO'O MBANG
SIMO Sandrine Epse OUAFO
SOPTSI Voltaire
TCHEPDA Soreil
TOUZA Isaac
TSIMI ELOUNDOU Roland
YAYA IBRAHIMA
YOSSA Alex Ludovic
ZELKIF Njamen

Les administrateurs

| | |
|------------------------|-----------|
| NTAKENDO Emmanuel | 676519464 |
| NGOUABA Borel | 699386259 |
| MBASSI Ova'a Philomène | 699322676 |
| KAMGUE Cyrille | 690709030 |
| NA'ATMEM SADOU Felix | 696875758 |
| DETIO Arnaud | 675864546 |
| ABDOULAYE | 675469530 |

Avant-propos

La collection des livres du groupe WhatsApp « Les grandprofs Info » a été conçue pour vous permettre d'entrer de manière efficiente et efficace dans les APC dans les classes de Troisième, Secondes, Premières et Terminales ESG et EST, adaptés aux nouveaux programmes informatiques, ces ouvrages vous offrent entre autres :

- ☞ **Des cours détaillés** (chaque module contient des unités d'apprentissage, et dans chaque unité d'apprentissage, des unités d'enseignement)
- ☞ **Des situations problèmes contextualisées** qui permettront à l'apprenant d'exécuter des tâches déterminées à travers des consignes explicites dont l'issue est la construction de ses propres compétences
- ☞ **Des activités d'intégration** : qui permettront à l'apprenant de réinvestir ses acquis dans un module donné
- ☞ **Des exercices d'application** : à la fin de chaque unité d'enseignement il est question de vérifier la compétence à travers des petits exercices

Ce travail a été fait par un groupe d'enseignants. Le déroulement (l'organisation) des travaux a eu lieu dans un groupe WhatsApp (" Les grandprofs Info"), et dans ce grand groupe sont nés d'autres petits groupes ou ateliers de travail, ainsi dans chaque atelier, les membres formaient des duos pour produire des résultats des objectifs fixés au préalable. À noter qu'il n'y a pas eu de rencontres physiques entre les membres.

Les contenus de ces livres répondent donc aux objectifs tant du côté enseignant que du côté apprenant, c'est une véritable référence exploitable à toutes fins utiles

Nous sommes ouverts à toutes les suggestions, remarques et même critiques constructives afin d'améliorer nos œuvres. Ne dit-on pas souvent que « la perfection n'est pas de ce monde ? »

La communicatrice du groupe Mme Mbassi Né Mendouga Philomène

Table des matières

| | |
|----------------------------------------------------------------------------------------|-----|
| Module 1 : SYSTEMES INFORMATIQUES..... | 6 |
| UA 1: EXPLOITATION D'UN ENVIRONNEMENT NUMÉRIQUE 6H..... | 6 |
| UE 1 : Description des types des systèmes informatiques IH ok..... | 6 |
| UE 2 : Description de l'architecture logicielle d'un système informatique IH ok..... | 9 |
| UE 3 : Description de l'architecture matérielle d'un système informatique 2H ok..... | 12 |
| UE 4 : Protection matérielle et logicielle d'un système informatique IH ok..... | 18 |
| UE 5 : Gestion des fluctuations de l'énergie électrique IH ok..... | 22 |
| UA 2: CRÉATION DE CONTENUS NUMÉRIQUES 4H..... | 25 |
| UE 6 : Généralités sur les tableurs 2H OK..... | 25 |
| UE 7 : Utilisation d'une feuille de calcul 2H ok..... | 29 |
| UA 3: RESEAUX INFORMATIQUES 10H..... | 36 |
| UE 8 : Types - Topologie et architectures réseaux 2H ok..... | 36 |
| UE 9 : LES EQUIPEMENTS RESEAUX IH ok..... | 41 |
| UE 10 : LES PROTOCOLES RESEAUX IH ok..... | 47 |
| UE 11 : L'ADRESSAGE RESEAU 2H ok..... | 49 |
| UE 12 : Créer et configurer un réseau informatique de 03 PC 2H..... | 54 |
| UE 13 : COMMANDES RESEAUX ET SIMULATION D'UN RESEAU 2H ok..... | 55 |
| Module 2 : SYSTEMES D'INFORMATION ET BASES DE DONNEES | 61 |
| UA 4: SYSTEMES D'INFORMATION 6H..... | 61 |
| UE 14: GENERALITE SUR LES SI IH ok..... | 61 |
| UE 15: ELABORATION D'UN MCD 3H ok..... | 64 |
| UE 16 : PASSAGE DU MCD AU MLD 2H ok..... | 73 |
| UA 5: BASES DE DONNÉES 10H..... | 77 |
| UE 17 : Les SGBD et fonctions de manipulations des données dans une BD 2H ok..... | 77 |
| UE 18 : Requêtes SQL de définition et de manipulation des données 3H ok..... | 81 |
| UE 19 : Requêtes SQL de manipulation des données et les operateurs logiques 3H ok..... | 84 |
| UE 20 : Implémentation une base de donnée dans Ms Access 2H ok..... | 90 |
| Module 3 : ALGORITHMIQUE ET PROGRAMMATION..... | 99 |
| UA 6: ALGORITHMIQUE 6H..... | 99 |
| UE 21 : Généralités Sur Les Algorithmes Et Structure Des Données 3H ok..... | 99 |
| UE 22 : Recherche séquentielle dans un tableau 2H ok..... | 106 |
| UE 23 : Recherche du maximum/minimum dans un tableau IH ok..... | 108 |
| UA 7: PROGRAMMATION EN C 8H..... | 114 |
| UE 24 : Les langages de programmation et IDE IH ok..... | 114 |
| UE 25 : Ecrire un programme en C 2H ok..... | 118 |

| | |
|---------------------------------------------------------------------------------|-----|
| UE 26 : STRUCTURES DE CONTROLE EN LANGAGE C IH ok..... | 124 |
| UE 27 : UTILISER LES TABLEAUX EN C IH ok..... | 129 |
| UE 28 : Traduire un algorithme de recherche séquentielle en C 2H ok..... | 132 |
| UE 29 : Identifier/interpréter/corriger les erreurs de programmation IH ok..... | 135 |

Module I : SYSTEMES INFORMATIQUES

UA I: EXPLOITATION D'UN ENVIRONNEMENT NUMÉRIQUE 6H

UE I : Description des types des systèmes informatiques IH

Compétence visée

- ❖ Décrire les trois types de systèmes informatiques (personnel, d'organisation, de contrôle et de commandes)

Situation-problème

Après avoir lu dans une revue scientifique un article sur les systèmes informatiques, votre petit frère qui est un fanatique de l'informatique aimerait en savoir davantage sur cette notion. Aidez-le en définissant l'expression système informatique et en donnant les types de systèmes informatiques.

INTRODUCTION

Un système informatique est un ensemble de moyens informatiques (matériels et logiciel) ayant pour but de collecter, traiter, stocker, acheminer et présenter des données. Les systèmes informatiques existent sous des formes variées en fonction du contexte dans lequel ils s'insèrent. On peut distinguer, entre autres les systèmes informatiques personnels, les systèmes informatiques d'organisation et les systèmes informatiques de contrôle et commande qui feront l'objet de ce cours.

I. LES SYSTÈMES INFORMATIQUES PERSONNELS

Un système informatique personnel est le système dont le but est de rendre des services (utilitaires ou ludiques) à son possesseur. Il peut comporter différents matériels, connectés de façon permanente ou non, comme :

- Des ordinateurs personnels (ordinateur de bureau, ordinateur portable, tablette etc.)
- Des équipements périphériques (imprimante, scanner, etc.),
- Des équipements de transmission de données (modem, Switch, box Internet, etc.),
- Des matériels plus spécialisés, comme les consoles de jeu ou les équipements de domotique (habitat intelligent),
- Des équipements terminaux de systèmes externes, comme les Smartphones en téléphonie.

Parmi les nombreux services offerts par les systèmes informatiques personnels, on peut citer :

- La navigation sur le web,
- Les messageries et les réseaux sociaux,
- Les jeux individuels ou en réseau,
- La bureautique, avec le calcul (à l'aide d'un tableur), le traitement de texte,
- Le multimédia (retouche d'images, traitement du son, montage vidéo, streaming) etc.

II. LES SYSTÈMES INFORMATIQUES D'ORGANISATION

Le système informatique d'organisation est un assemblage d'éléments matériels et logiciels qui coopèrent pour réaliser un objectif commun en utilisant un réseau comme moyen d'échange des données. Ce réseau peut être un réseau local ou un réseau étendu, le plus souvent Internet.

Les systèmes informatiques d'organisation se trouvent au sein des organisations comme les entreprises, les associations, les administrations, les laboratoires de recherche, etc. Ils comprennent une grande diversité de composants parmi lesquels :

- Les ordinateurs (postes de travail, serveurs d'applications, serveurs de données, des supercalculateurs, etc.),
- D'autres équipements de traitement et de transmission de données (capteurs et actionneurs, des concentrateurs, commutateurs et routeurs, des robots, des machines dédiées, etc.),
- Les réseaux (LAN, MAN, WAN).

III. LES SYSTÈMES INFORMATIQUES DE CONTRÔLE ET DE COMMANDE

Un système informatique de contrôle et commande est le système qui reçoit des données sur l'état d'un procédé extérieur via des capteurs, les traite et selon le résultat, agit sur ce procédé extérieur via des actionneurs, afin de le maintenir dans l'état souhaité. Les systèmes informatiques de contrôle et commande constituent la grande partie des systèmes informatiques car ils sont enfouis dans les objets de la vie quotidienne. Un système informatique de contrôle et de commande est composé de :

- Capteurs pour l'acquisition des données
- Opérateurs permettant l'analyse et le traitement des données
- Actionneurs chargés de la restitution des données

NB : Les **systèmes de contrôle et de commande** sont indispensables sur toute ligne de production automatisée moderne à travers l'interface homme-machine (IHM).

Les systèmes informatiques sont très répandus de nos jours. Leur objectif principal est le traitement automatique de l'information. Ils comptent plusieurs équipements parmi lesquels on peut citer les matériels réseau, les capteurs et actionneurs, les machines spécialisées (comme les guichets automatiques bancaires ou les boîtiers de stockage en réseau, les robots, les smartphones, les cartes à puce etc.).

Jeu bilingue

1. Définir Système informatique en anglais

Data processing system is a system of interrelated techniques and means of collecting and processing the data needed to organize control of some system.

2. Traduire en anglais les expressions suivantes :

Système informatique : Data processing system,

Système informatique personnel : personal Data processing system

Système informatique de contrôle et de commande : control and command data processing system.

Activités d'intégration

- 1- Définir : Système informatique, actionneur, capteur
- 2- Enumérer trois services offerts par le système informatique personnel.
- 3- Expliquer le principe de fonctionnement du système informatique de contrôle et de commande.
- 4- Identifier la famille de système informatique à laquelle appartiennent les systèmes embarqués.
- 5- Votre père a acheté un ordinateur, une imprimante, une flybox, un scanner qu'il a relié afin d'effectuer ses travaux.
 - a. Identifier le système informatique mis sur pieds par votre père.
 - b. Citer deux autres systèmes informatiques
 - c. Enumérer deux tâches que pourras effectuer votre père avec ce système informatique.

UE 2 : Description de l'architecture logicielle d'un système informatique IH

Compétence visée : Prise en main d'un système informatique ;

Situation problème

En observant votre Père qui travaille sur son ordinateur à la maison, votre petit frère qui est encore à l'école primaire est intrigué par quelques actions posées par votre Père, mais surtout par les réactions de l'ordinateur qui semble lui obéir à la lettre.

Il se rapproche de vous et vous pose les questions suivantes :

QI : *Qu'est ce qui fait en sorte que l'on puisse donner des ordres à un ordinateur qui est à la base formé que de ferraille et circuits électriques ?*

Réponse attendue : des programmes

I. GENERALITES

Un **système informatique** comprend deux grandes parties : une partie matérielle appelée Hardware et une partie logicielle appelée Software. Cette partie logicielle est constituée d'un ensemble de programmes qui permettent au système informatique d'accomplir toutes ses tâches.

Définitions :

Information : élément de connaissance susceptible d'être transformé afin d'être traité par un système informatique.

Donnée : information représentée sous une forme conventionnelle, afin de pouvoir être traitée automatiquement.

Langage informatique: ensemble de caractères, de symboles, de mots-clés et de règles permettant de donner des instructions à un ordinateur.

Instruction : ordre que l'on donne à un processeur.

Programme : séquence d'instructions exécutable par l'ordinateur

Logiciel : Ensemble de programmes.

Progiciel: ensemble complet de programmes informatiques, conçu pour différents utilisateurs en vue d'une même application, généralement professionnelle, et commercialisé avec une documentation.

Freeware est un logiciel entièrement gratuit, qu'on peut copier comme on veut.

Graticiel Synonyme de l'anglais **freeware**, cousin de **gratuiciel**.

Logiciel libre: ensemble d'**applications** livrées avec leurs **sources**, que l'on peut donc modifier à volonté pour l'adapter à ses besoins. Libre ne signifie pas forcément gratuit. En anglais : **Open Source**.
Contraire: **logiciel propriétaire**.

Logiciel propriétaire: logiciel dont l'acheteur ne peut utiliser qu'en suivant des règles établies par son constructeur parfois extrêmement limitatives.

Ludiciel: logiciel de divertissement

Didacticiel c'est un logiciel destiné à l'enseignement

Q2 : Est-ce que tous ces programmes viennent avec l'ordinateur lors de l'achat ?

Réponse attendue : non.

Q3 : Comment appelle-t-on le petit programme que les constructeurs mettent dans les machines lors de la fabrication

Réponse attendue : le bios

II. LE BIOS

Le BIOS (Basic Input Output System) en français **système de gestion de base des entrées / sorties** est un petit programme installé par les constructeurs lors de fabrication de l'ordinateur. Il a pour rôle de vérifier l'état du matériel et de permettre ainsi l'installation des premiers programmes. Lors de la mise sous tension de l'ordinateur, il exécute le POST (Power On Self Test). Lors de cette phase, les différents composants sont testés, le POST ne se déroule pas bien, des messages d'erreurs sont affichés à l'écran. Le BIOS est enregistré dans une mémoire de type ROM : Le CMOS (Complementary Metal-Oxide Semiconductor). Le CMOS doit être constamment alimenté par d'une pile pour sauvegarder des informations sur la configuration du système (l'heure, la taille de la mémoire vive installée sur l'ordinateur, le type de disquette utilisée, les caractéristiques du disque dur, etc...).

Q4 : comment appelle-t-on le premier programme qu'on installe en premier dans un ordinateur

Réponse attendue : le SE

III. LE SYSTEME D'EXPLOITATION

I. Définition

Un **logiciel de base** est un ensemble de programmes qui assurent le bon fonctionnement de l'ordinateur. Les logiciels de base constituent **Le système d'exploitation**.

Le système d'exploitation sert d'interface entre l'utilisateur et le matériel. Autrement dit il assure l'interaction entre le matériel et l'utilisateur de l'ordinateur.

EXEMPLES : MacOS, Unix, Linux, MS-DOS, Windows...

2. Les fonctions d'un système d'exploitation

Il a pour fonctions :

- la gestion du processeur,
- la gestion de la mémoire,
- la gestion des entrées/sorties,
- la gestion des programmes
- la gestion des utilisateurs

3. Les types de systèmes d'exploitation

- Selon la gestion des tâches on distingue : les systèmes d'exploitation monotâches des systèmes d'exploitation multitâches
- Selon la gestion des postes on distingue : les systèmes d'exploitation monoposte des systèmes d'exploitation multi-postes
- Selon la gestion des utilisateurs on distingue : les systèmes d'exploitation mono-utilisateur des systèmes d'exploitation multi-utilisateur

Q4 : quel nom donne-t-on autres logiciels que l'on utilise pour exécuter des tâches avec un ordinateur ?

Réponse attendue : les logiciels d'application

IV. LES LOGICIELS D'APPLICATION

Il existe deux types de logiciels : les logiciels de base que nous avons évoqués dans le paragraphe précédent et les logiciels d'application.

Un **logiciel d'application** est un ensemble de programmes spécialement conçus pour exécuter une tâche précise ou particulière sur l'ordinateur. On les regroupe par catégories en fonction de ce qu'ils exécutent.

EXEMPLE : Tableur, ludiciel, Didacticiel,...

Dans cette catégorie de logiciels, on retrouve **les utilitaires**.

Un **Utilitaire** c'est un logiciel qui exécute des opérations annexes aux fonctions principaux du système d'exploitation.

Entre autres exemples on a les pilotes ou drivers qui sont des petits programmes permettant de gérer des périphériques.

Jeu bilingue

| Français | Anglais |
|------------------------|------------------|
| Système informatique | Computer |
| Logiciel | Software |
| Système d'exploitation | Operating System |
| Pilote | Driver |

UE 3 : Description de l'architecture matérielle d'un système informatique 2H

Compétences visées

- Décrire les caractéristiques de quelques périphérique et composants ;
- Enoncer le rôle d'un pilote ;
- Décrire les principaux composants internes de l'unité centrale.

Situation problème En entrant dans la salle informatique de votre établissement, vous trouverez sur la table du professeur les composants ci-dessous de l'ordinateur. Tout le monde se pose des questions sur ce que c'est et à quoi tout cela peut bien servir.





Consigne : décrire chacun des éléments

INTRODUCTION

Un système informatique est généralement constitué d'une partie matérielle et d'une partie logicielle. La partie matérielle possède une **unité centrale** et des **périphériques**. Unité centrale d'un ordinateur regroupe le microprocesseur, la mémoire, le disque dur et les lecteurs... Un périphérique est un matériel pouvant être connecté à l'unité centrale par l'intermédiaire d'un **pilote**.

I- Caractéristique de quelques périphériques

a) Périphériques d'entrée

Les **périphériques** d'entrées permettent de transmettre information à unité centrale. Les plus connus sont :

- Le **clavier** est un dispositif qui permet de saisir les caractères et qui permet d'envoyer des commandes vers l'unité centrale
- La **souris** est un périphériques de pointage servant à déplacer un curseur sur l'écran et permettant de sélectionner, de déplacer ou de manipuler les objets grâce à des boutons.
- Le **scanner** est périphériques d'acquisition permettant de numériser des documents est a dire transformer un document papier en images numérique
- Le **microphone** c'est un périphérique sensible à la voix et permet d'introduire les informations dans l'ordinateur sous forme sonore
- **Camera et la webcam** : ils permettent de capturer des images réelles et de les numériser afin d'être traitées par un ordinateur
- L'**appareil photo** c'est un périphériques destinés à capturer des images
 - Le **crayon optique** : est un stylet électronique, permettant de dessiner ou sélectionner des options directement sur écran à tube

b) Les périphériques de sortie

Les périphériques de sortie permettent à l'unité centrale de transmettre les informations aux utilisateurs

- **L'imprimante** est un périphérique qui permet de faire sortir sur du papier les données contenus dans un ordinateur
- **L'écran** permet d'afficher les données contenues dans unité centrale
- **Le vidéoprojecteur** est un appareil qui permet de projeter sur un mur une vidéo, une image
- **Les haut-parleurs** permettent de faire sortir du son
- **Le graveur** permet de sauvegarder les données d'un ordinateur dans un disque optique (CD-ROM, DVD ...)
- **Ecran tactile** est un périphérique informatique qui cumule les fonctionnalités d'affichage d'un écran, d'une souris, celle d'un stylet optique

c) Les périphériques d'entrée sortie

Encore appelé **périphériques mixte**, un périphérique d'entrée/sortie est un périphérique ou un organe qui permet d'entrer et sortir des informations d'un ordinateur .il s'agit de :

- **Le modem** c'est un périphérique qui permet de communiquer avec des utilisateurs par intermédiaire d'une ligne téléphonique. il permet de se connecter à internet
- **Le lecteur graveur CD/DVD** permet la lecture et la gravure des CD ou DVD
- **La clé USB et le disque dur externe** : ce sont les supports de stockage qui permettent de conserver les informations.

d) Les périphériques de communication

Ce sont des appareils électroniques permettant l'échange d'informations entre ordinateurs. Desque les données "quittent" l'ordinateur, elles vont subir un certain nombre de manipulations avant d'arriver à leur destination. Comme périphériques de communication nous avons :

- **La carte réseau** : c'est une carte de circuit imprimés qui permet la communication réseau depuis et vers un ordinateur personnel (échanges de données). Elle constitue l'interface entre ordinateur et le câble réseau
- **Le modem** c'est un périphérique qui permet de communiquer avec des utilisateurs par intermédiaire d'une ligne téléphonique. il permet de se connecter à internet
- **Le répéteur** : c'est un équipement qui relie deux ordinateurs distant tout en régénérant le signal.
- **Le switch ou commutateur** c'est équipement qui permet de choisir le destinataire exacte vers qui envoyer information

- **Le routeur** : c'est un équipement d'interconnexion qui permet de sortir d'un réseau pour aller vers un autre
- **Le pont** : il permet de passer d'un réseau vers un autre réseau de même type
- **La passerelle** : c'est un système matériel et logiciel qui permet des réseaux qui utilisent des protocoles différents

II- Les principaux composants de l'unité centrale.

L'unité centrale comporte plusieurs éléments dont les plus importants sont:

I. la carte mère

La carte mère un circuit imprimé constituant le support principal des éléments essentiels d'un ordinateur. Elle permet les échanges d'informations entre les différents composants vitaux d'un pc.

Une carte mère est caractérisée par:

- **le facteur d'encombrement** : c'est dire sa forme
- **le chipset** : c'est un circuit qui organise l'interconnexion entre tous les composants de la carte mère
- **l'horloge et la pile du CMOS** : l'horloge est chargée de synchroniser les signaux du système
- **le slot et le socket** : qui sont les supports du processeur
- **les connecteurs d'extension** : sont des réceptacles dans lesquels on insère des cartes d'extension. On peut citer : le connecteur Isa , connecteur PCi, AGP...
- **les connecteurs d'entrée-sortie** parmi lesquels : le port série (connecte les anciens périphériques), port parallèle (connectait imprimante mais désormais obsolète), port USB, Port VGA, prises audio

2. le microprocesseur

Est un circuit intégré qui effectue les opérations arithmétiques et logiques. Il est constitué de l'unité arithmétique et logique encore appelée unité de traitement, l'unité de contrôle responsable de la lecture en mémoire et du décodage des informations.

Les caractéristiques du microprocesseur

- la largeur de ces registres internes de manipulation des données (8, 16, 32, 64 bits) ;
- la cadence de son horloge exprimée en Mhz ou Ghz ;
- le nombre de noyaux de calcul (core)
 - son jeu d'instructions qui dépend de la famille (CISC, RISC , etc)



3. les mémoires

Une mémoire est un dispositif électronique capable d'enregistrer, de conserver et de restituer des informations. On la retrouve dans les ordinateurs, les jeux, les GPS et de nombreux appareils.

La caractéristique d'une mémoire:

- **La capacité:** nombre total de bits que contient la mémoire. Elle s'exprime aussi en octet
- **Le format des données:** nombre de bits qu'on peut mémoriser par case mémoire. On parle de largeur du mot mémoire
- **le temps d'accès :** temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture en mémoire et l'instant où la première information est disponible sur le bus de données ;
- **le temps de cycle :** il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture
- **le débit :** nombre maximum d'informations lues ou écrites par seconde.

On distingue plusieurs types de mémoire

- **la mémoire vive**, généralement appelée RAM, c'est la mémoire principale du système, elle permet de stocker de manière temporaire des données lors de l'exécution d'un programme
- **la mémoire morte** appelée ROM, est un type de mémoire permettant de conserver les informations qui y sont contenues même lorsque la mémoire n'est plus alimentée
- **la mémoire BIOS** est le programme qui sert d'interface entre le système d'exploitation et la carte mère
- **la mémoire Flash**

Les mémoires de l'ordinateur



4. le disque dur

C'est une mémoire de masse magnétique utilisée principalement dans des ordinateurs, mais également dans caméscopes, des lecteurs/enregistreur de DVD

5. les bus

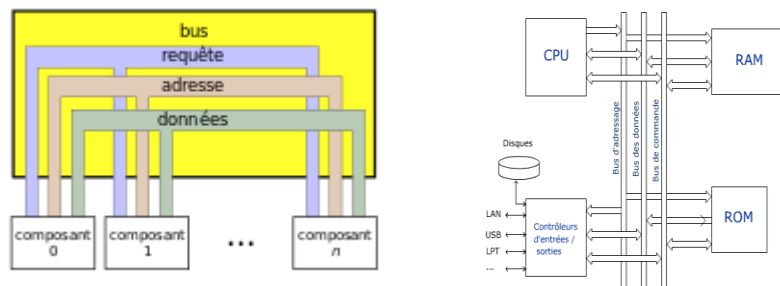
On appelle bus un ensemble de liaison physique pouvant être exploitées en commun par plusieurs éléments matériels afin de communiquer.

Un bus est caractérisé par:

- sa largeur, c'est le volume d'information transmise simultanément
- sa fréquence c'est le nombre de paquet de données envoyés ou reçus par seconde elle est exprimée en hertz

Quelque type de bus

- le **bus d'adresse** transporte les adresses mémoires auxquelles le processeur souhaite accéder pour lire ou écrire une donnée
- le **bus de données** véhicule les instructions en provenance ou à destination du processeur.
- Le **bus de contrôle** transporte les ordres et les signaux de synchronisation en provenance de l'unité de commande et à destination de l'ensemble des composants matériels



6. Le boîtier d'alimentation

L'alimentation est chargée de convertir la tension électrique du secteur en différentes tensions continues compatibles avec les circuits de l'ordinateur.



UE 4 : Protection matérielle et logicielle d'un système informatique IH

Compétences visées

- Lister quelques causes de dysfonctionnement du matériel et des logiciels ;
- Décrire la fonction du bloc d'alimentation ;
- Décrire le rôle des équipements suivants : onduleur, régulateur de tension ;
- Lister quelques risques liés à la non protection de l'environnement de travail ;
- Protéger un ordinateur/imprimante à l'aide des équipements suivants : onduleur, régulateur de tension ;

Situation problème

Paul Vient d'acquérir un ordinateur il y'a trois mois. Après installation il s'est lancé dans les téléchargements des logiciels sur le net sans controle. Malheureusement il n'a mis en place aucun dispositif de sécurité et son ordinateur a des difficultés à démarrer. Il se rapproche de vous et vous pose quelques questions

- a) Quelles sont les causes de ce dysfonctionnement ?
- b) Ce dysfonctionnement est-il matériel ou logiciel ?
- c) Comment pourrait-il éviter de tels désagréments dans le futur ?

Contenu de la leçon

I- Définition des concepts

- **Panne:** Arrêt accidentel par un dysfonctionnement du fonctionnement d'un système ou d'un élément.
- **Panne:** C'est un état d'un équipement informatique inapte à accomplir une fonction requise.
- La **maintenance informatique** regroupe toutes les tâches et actions nécessaires **au bon fonctionnement d'un système informatique.**
- La **maintenance préventive** consiste à empêcher tout incident technique au moyen de la prévention, autrement dit, elle est une maintenance effectuée avant la détection d'une défaillance d'un outil informatique.
- La **maintenance préventive, ou maintenance technique planifiée**, consiste à intervenir sur un équipement à intervalles réguliers ou en fonction de critères prédéfinis. Son objectif premier est de réduire les risques de panne sur vos équipements

II- Les causes de dysfonctionnement du matériel et des logiciels

a- Les causes de dysfonctionnement des logiciels

Plusieurs causes peuvent être à l'origine d'un dysfonctionnement d'un logiciel:

- le défaut de conception du logiciel ou « bug »,

- les virus informatiques ou logiciels malveillants,
- les incompatibilités diverses entre applications,
- des paramétrages erronés,
- des événements propres à l'utilisateur (mauvaises manipulations),
- des événements extérieurs (température, champs magnétiques).

b- Les causes de dysfonctionnement du matériel

Plusieurs causes peuvent être à l'origine d'un dysfonctionnement d'un matériel:

- la défectuosité des matériels informatiques,
- Mauvais branchement d'un périphérique
- Choc physique du matériel
- Les variations de tension, surtension
- Foudre et coupure incessantes du courant électrique
- **Les pannes Post**
- **Les pannes CMOS/BIOS**
- **Les pannes CARTES MERES**

La carte mère coordonne le bon fonctionnement des composants du système. Elle permet aux périphériques de communiquer et de travailler ensemble

- **Les pannes CPU**

Les symptômes d'une panne processeur sont une baisse de performance, des bips au POST, ou un système qui ne fonctionne pas correctement. Ces erreurs indiquent généralement qu'une erreur interne est survenue.

- **Les pannes RAM**
- **Les pannes d'alimentation**

L'alimentation joue un rôle vital dans le fonctionnement de tout système informatique. Si l'alimentation ne fonctionne pas correctement, les composants informatiques seront mal alimentés et ne fonctionneront pas normalement.

- **Les pannes des disques durs**
- **Pannes de périphériques**

La plupart des périphériques d'entrée sont détectés au démarrage. Quand on diagnostique des périphériques d'entrée, vérifiez que le périphérique est correctement connecté.

III- fonction du bloc d'alimentation ;

Le **bloc d'alimentation** (*power supply unit* en anglais, souvent abrégé **PSU**) ou simplement l'**alimentation** d'un PC est le matériel informatique l'alimentant.

Le **rôle du bloc d'alimentation** est de transformer le courant alternatif de tension efficace 220V. 50Hz en un courant continu et constant compatibles avec les circuits électroniques de l'ordinateur.

En d'autres termes, **Le rôle du bloc d'alimentation** consiste fournir l'énergie basse tension nécessaire pour alimenter la carte mère et les divers éléments inclus dans l'unité centrale.

Le bloc d'alimentation va donc faire 3 choses :

- Passer d'un courant alternatif à un courant continu
- Baisser la tension
- proposer plusieurs tensions en sortie

IV- Rôle des équipements

a- Onduleur

Un **onduleur** est un dispositif d'électronique de puissance permettant de générer des tensions et des courants alternatifs à partir d'une source d'énergie électrique de tension ou de fréquence différente.

Un onduleur est un boîtier à brancher sur secteur. Il permet de protéger les appareils électroniques reliés et de corriger les perturbations électriques du réseau par un régulateur de tension. C'est un appareil indispensable pour protéger les équipements en cas de coupure de courant.

b- régulateur de tension

Un **régulateur de tension**, est un organe électrotechnique ou un composant électronique dont le rôle est de maintenir à sa sortie, dans certaines limites, une tension constante, indépendamment de la charge et de la tension d'entrée.

V- Quelques risques liés à la non protection de l'environnement de travail

Les risques liés à la non protection d l'environnement de travail sont:

- La surtension et sous tension
- La destruction des appareils
- Invasion des Virus informatique
- La perte d'informations
- Invasion de la poussière

VI- Protéger un ordinateur/imprimante à l'aide des équipements suivants : onduleur, régulateur de tension

Les circuits électroniques sensibles de l'ordinateur et de l'imprimante peuvent être endommagés par des tensions trop élevées.

Il est facile de relier les appareils électroniques que vous souhaitez protéger en les branchant sur les prises dédiées de l'onduleur ou du régulateur de tension. Pour protéger son ordinateur à l'aide d'un onduleur ou d'un régulateur de tension, il faut :

- Installer le pilote

- Brancher les câbles

Exercices

- 1- Citer quatre causes de dysfonctionnement du matériel
- 2- Citer quatre causes de dysfonctionnement des logiciels
- 3- Quelle est la fonction du bloc d'alimentation ?
- 4- Donner le Rôle des équipements de l'onduleur,
- 5- Donner le Rôle des équipements du régulateur de tension
- 6- Citer 3 risques liés à la non protection de l'environnement de travail ;

Jeu bilingue des mots contenu dans la colonne savoirs essentiels du programme

a- **Failure** : panne

b- **Maintenance** : Maintenance

c- **Preventive Maintenance** : Maintenance préventive

UE 5 : Gestion des fluctuations de l'énergie électrique IH

Compétence visée:

A la fin de cette leçon l'élève devra être capable de présenter et décrire les outils de protection d'un système informatique contre les fluctuations d'énergie.

Objectifs spécifiques:

- lister les causes du dysfonctionnement matériel;
- Identifier et décrire les outils de protection matériel contre les fluctuations d'énergie ;

Contrôle de pré requis:

- Maîtriser les composants de l'ordinateur et leurs rôle;
- Mettre un ordinateur en marche.

Situation Problème:

M. TCHIO vient d'aménager dans un nouveau quartier. En soirée il constate que les ampoules à la maison n'éclairent pas bien, et que son moniteur s'allume mais l'UC n'arrive pas démarrer entre 18h et 21h30. Sachant que vous avez des connaissances en maintenance informatique, aidez M. TCHIO à comprendre son problème.

1. définir maintenance informatique;
2. Quel est le problème rencontré par M. TCHIO ?
3. Quel conseil allez-vous prodiguer à M. TCHIO ?
4. Citer deux équipements de protection que vous allez recommander à M. TCHIO.

INTRODUCTION

L'ordinateur subit des menaces tout au long de son cycle de vie. Pour freiner l'effet des menaces afin de prolonger la durée de vie de notre ordinateur il est nécessaire d'appliquer certaines mesures qui visent à le maintenir. Cette mesure est généralement qualifiée de maintenance. L'ordinateur étant composé de la partie matérielle et la partie logicielle, on distingue la maintenance matérielle et la maintenance logicielle. Parmi les causes de disfonctionnement matériel nous avons les fluctuations d'énergie qui sont des causes à ne pas négliger. Dans cette leçon nous présenterons les causes liées aux fluctuations d'énergie ainsi que quelques mesures de prévention.

I. Quelques causes instabilité matériel

Les pannes matérielles ne sont pas essentiellement physiques. Tout composant de l'ordinateur doit impérativement être connu du système d'exploitation pour être utilisé. Ainsi, les problèmes peuvent être d'ordre logiciel :

- pilote défaillant ce qui empêche que le matériel soit fonctionnel ;

- Virus Informatique qui fait tourner la machine à un rythme anormal afin de créer le surchauffe pour détruire le matériel.

Ils peuvent aussi être d'ordre matériel :

- Accumulation de la poussière ;
- Passage d'une foudre ;
- Fluctuation du courant.

2. Outil de protection contre la fluctuation de courant

La **fluctuation** de tension électrique est causée par des perturbations électromagnétiques ou par des variations de puissance sur le réseau porteur de cette tension. Elle est généralement caractérisée par une instabilité de la tension (baisse ou hausse de tension intempestive).

Parmi les outils de protection matérielle contre les fluctuations on a :

a) le bloc d'alimentation

Faisant partie intégrante de l'Unité centrale, il contient en son sein l'alimentation et un ou plusieurs ventilateurs pour refroidir l'unité centrale. Il a pour principal rôle de prendre un courant instable (alternatif) et de le transformer en courant continu pour alimenter les composants de l'ordinateur (110V-220V en 5V-12V).



b) para sur-tenseur

C'est une rallonge multiprise spéciale constitué d'un para surtension et parafoudre qui protège efficacement les appareils informatiques contre l'orage ou les coupures de courant.



c) régulateur de tension

C'est un appareil qui a pour fonction de convertir le courant alternatif d'amplitudes variables en courant continu d'amplitudes constantes et limitées pour assurer le bon fonctionnement des équipements qu'il

alimenté. Un régulateur de tension fonctionne en trois temps : redresser le courant l'écrêter et limiter ses variations.

Il existe deux types de régulateurs de tension : **analogique** et **numérique**.



Figure 1: régulateur numérique



Figure 2: régulateur analogique

d) Onduleur

C'est un appareil électronique ayant la capacité d'accumuler de l'énergie et de stabiliser l'énergie électrique aux tensions variables à la sortie. Il dispose d'une batterie ayant la capacité d'alimenter les équipements en cas absence d'alimentation du courant secteur. La durée d'alimentation dépend de la puissance de la batterie et des appareils connectés.



CONCLUSION

En somme, nous pouvons dire que les causes du dysfonctionnement matériel peuvent être logicielles ou matérielles. Parmi les causes matérielles nous avons présenté dans cette leçon les fluctuations de courant (baisse et hausse de la tension) qui peuvent endommager les composants de l'ordinateur. Nous avons présenté l'onduleur, le régulateur de tension, le bloc d'alimentation et le parasurtenseur comme outils de protection des composants de l'ordinateur contre les fluctuations de courant ce qui pourra rallonger la durée de vie de nos équipements.

Jeux Bilingue

| Français | Anglais |
|-----------------------|-------------------|
| onduleur | Inverter |
| Régulateur de tension | Voltage regulator |

¹ L'écrêtage consiste à supprimer une partie de l'amplitude d'un signal. Le redressement supprimant les alternances négatives peut être considéré comme un cas particulier de l'écrêtage.

UA 2: CRÉATION DE CONTENUS NUMÉRIQUES 4H

UE 6 : Généralités sur les tableurs 2H

Compétences visées : Au terme de cette leçon l'apprenant doit être capable de :

- Définir tableur et donner l'utilité des tableurs,
- Décrire les parties qu'on retrouve dans l'interface graphique d'un tableur.
- Décrire les opérations qu'on peut réaliser dans un tableur notion de référence dans les tableurs

Situation problème : après avoir brigué avec succès les activités du club scolaire dans lequel vous êtes secrétaire, il est demandé à votre bureau de fournir un rapport financier de vos activités. Sachant que vous avez des connaissances avancées en informatique, notamment en bureautique, il vous demande de le renseigner sur la façon dont sera présenté ce document.

Consigne:

- D'après vous quel est le meilleur moyen de présenter un rapport financier?
- Quels outils seraient mieux adaptés à la réalisation de ce document?
- Pouvez-vous citer d'autres documents que l'on peut réaliser avec l'un des outils cités?

INTRODUCTION

Aujourd'hui, l'informatique est appliqué a de nombreux domaines parmi lesquelles le domaine de gestion, la santé, l'armée etc... les outils manipulés dans ces domaines sont très variés. On rencontre entre autres des tableurs. La principale utilisation d'un tableur est le calcul automatique de formules.il est destiné a des multiples réalisations à savoir : tableaux, calculs, graphismes, etc...

I. LES TABLEURS : DEFINITIONS ET EXEMPLES

I. Définitions

Un tableur est un programme informatique capable de manipuler des feuilles de calcul. A l'origine, les tableurs étaient destinés aux traitements automatisés des données financières. Aujourd'hui ils sont utilisés pour effectuer des tâches variées à savoir, la gestion de bases de données simples, la production graphique, les analyses statistiques, etc... Dans un tableur, on ne parle pas de fichiers mais plutôt de classeur. Ainsi, nous diront qu'un classeur est un document tableur comportant plusieurs feuilles de calcul.

2. Exemples de tableurs

La principale utilisation d'un tableur est le calcul automatique de formules. Les utilisés sont :

- Excel, qui est extrait de la suite bureautique Microsoft Office ;
- OpenCalc extrait de la suite bureautique Open Office
- Sun StarOffice calc, de la suite starOffice ;
- Lotus 1-2-3 de la suite SmartSuite
- KSpread, de la suite KOffice sous Linux
- Corel Quattro Pro de la suite WordPerfect.

II. MISES EN FORMES DANS UN TABLEUR (CAS DE MS EXCEL)

Une mise en forme est un processus qui consiste à améliorer un document pour l'embellir. En ce qui concerne les tableurs, il s'agit de préparer le résultat d'une opération, pour qu'il soit appréciable par le lecteur. Un tableur propose plusieurs styles de tableau prédéfinis que vous pouvez utiliser pour mettre en forme un tableau en un clin d'œil. Si les styles de tableau prédéfinis ne vous conviennent pas, vous pouvez créer et appliquer un style de tableau personnalisé. Bien que vous ne puissiez supprimer que les styles de tableau personnalisés, vous pouvez enlever un style de tableau prédéfini de sorte qu'il ne soit plus appliqué à un tableau. Vous pouvez également ajuster la mise en forme du tableau en choisissant les options de styles rapides correspondant à des éléments du tableau, par exemple Ligne d'en-tête, Ligne Total, Première colonne, Dernière colonne, Lignes à bandes et Colonnes à bandes ou Bouton de filtre. Sous EXCEL par exemple, pour Choisir un style de tableau Lorsqu'une plage de données n'est pas mise en forme en tant que tableau, Excel la convertit automatiquement en tableau lorsque vous sélectionnez un style de tableau. Vous pouvez également modifier la mise en forme d'un tableau existant en sélectionnant une autre mise en forme.

1. Sélectionnez la cellule ou la plage de cellules du tableau à laquelle vous voulez appliquer une mise en forme de tableau.

2. Dans l'onglet Accueil, cliquez sur Mettre sous forme de tableau.

Cliquez sur le style de tableau à utiliser.

III. QUELQUES FONCTIONS DE BASE DANS LES TABLEURS

Dans un tableur, une fonction est un outil qui permet de produire un résultat d'un type bien précis ; elles sont classées en différentes familles à savoir : statistiques, math et trigonometrie, scientifiques, finances, etc. ... Parmi les fonctions existantes, nous pouvons citer :

➤ La fonction SOMME () : a pour rôle additionner le contenu d'un ensemble de cellules, il convient d'indiquer, entre parenthèses, les coordonnées de la première cellule, suivi du séparateur « : » et les coordonnées de la dernière cellule.

Exemples : =SOMME (A1 :A10) Additionne le contenu des cellules comprises entre A1 à A10.

=SOMME (A1 ;A3;A5:A9) Additionne le contenu de la cellule de A1 avec celui de A3, puis avec celui des cellules comprises entre A5 et A9. Additionne le contenu des cellules comprises entre A1 et A3 avec celui des cellules comprises entre A5 et A9.

➤ La fonction SI() : elle permet de tester des hypothèses en faisant en sorte qu'un calcul dépende de la réalisation d'une condition.

➤ La fonction SOMME.SI() : vous permet d'effectuer une somme de valeurs en fonction d'une condition. Par exemple, vous pouvez obtenir, pour chaque marque de véhicule, le montant des ventes.

➤ La fonction MOYENNE() calcule la moyenne d'une plage de cellules numériques.

- La fonction RANG () : a pour rôle de ranger des valeurs dans un ordre précis. En mettant comme indicateur « 0 » les nombres seront rangés dans un ordre décroissant et « 1 » les nombres seront rangés dans un ordre croissant.

IV. QUELSQUES OPERATIONS SUR LES TABLEURS

Protéger une feuille de calcul

Il s'agit ici de configurer un mot de passe afin de limiter les accès à la feuille de calcul uniquement aux personnes autorisées. Pour cela il faut procéder comme suit :

Sélectionner les cellules qui peuvent être modifiées → sous l'onglet Accueil/cellule, cliquez sur la commande format → désactivez la commande « verrouiller la cellule » → cliquez sur « protéger » la feuille → tapez un mot de passe → confirmer le mot de passe.

Les zones d'impression

Il s'agit des données qui seront imprimés sur chaque page. Pour définir une zone d'impression on sélectionne les cellules que nous voulons imprimer, on va dans l'onglet mise en page, on choisit zone d'impression puis on définit la zone d'impression.

Notion de références dans les tableurs

La référence d'une cellule est le code qui nous permet de l'identifier (A1, B3, C15, T375...). On distingue deux types de références sur les tableurs à savoir :

- ❖ La référence Absolue, qui donne la ligne et la colonne précise. Exemple : LIC5 désignera la cellule de croisement de la ligne I et de la colonne 5
- ❖ La référence relative, qui sont des références dont les cellules vont varier au moment de recopier les données; ce qui signifie par exemple que si on fait un copier/coller d'une formule dans une cellule A1, alors la référence de la cellule s'ajustera en fonction du nouvel emplacement. **Exemple** : A1, C15, T375...

CONCLUSION

Un tableur permet de réaliser des feuilles de calculs. Ce sont des tableaux dans lesquels des calculs sont effectués en utilisant des formules et/ou des fonctions. Certains tableurs comme MS EXCEL peuvent aussi être utilisés comme des bases de données qui est par exemple la fonction d'un autre logiciel tel que MS ACCES.

Jeu bilingue : traduire en Français ou Anglais les mots ou expressions soulignés ci-dessous :

Un spreadsheet nous permet de réaliser des documents tels que des bulletins ou encore des **bills** et nous pouvons les réaliser à l'aide des cellules contenus dans ces tableurs. Un **système d'exploitation** donne la possibilité d'installer plusieurs **spreadsheet**.

Exercice : Soit une feuille du tableur suivant

- Comment appel-t-on cette interface?

- Comment appel-t-on la zone dans laquelle on saisit une donnée dans un tableur
- Donnez le nom de la zone dans laquelle est écrit le mot «BEAU»
- Donnez le nom de la zone dans laquelle est écrit le mot « mon Ie tableau »

UE 7 : Utilisation d'une feuille de calcul 2H

Compétences visées

- Réaliser un tableau croisé dynamique;
- Reproduire une facture, un devis, un bulletin de paie ou de notes ;
- Imprimer une feuille de calcul.

Situation problème

Madame WANDA est la patronne d'un établissement privé d'enseignement secondaire. Pour mieux gérer son établissement, elle souhaite automatiser certaines tâches notamment la production des bulletins de notes et des devis. Pour le stockage des données elle compte utiliser des tableaux. N'étant pas très avisé personnellement à l'utilisation des tableaux, elle se rapproche de vous en vous posant quelques questions.

Comment produire un bulletin de note?

Comment va t'elle créer ce tableau?

CONTENU DE LA LEÇON

I- Les tableaux croisés dynamiques

I-I C'est quoi un tableau croisé dynamique ?

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

Le **tableau croisé dynamique** (ou TCD) permet de synthétiser les données d'une base de données de façon à obtenir les "croisements" de données souhaités.

Comme son nom l'indique, un tableau croisé dynamique, est un nouveau tableau que vous allez créer sur Excel et qui va reprendre des données que vous avez déjà entré dans votre feuille de calcul et qui va les croiser de manière dynamique.

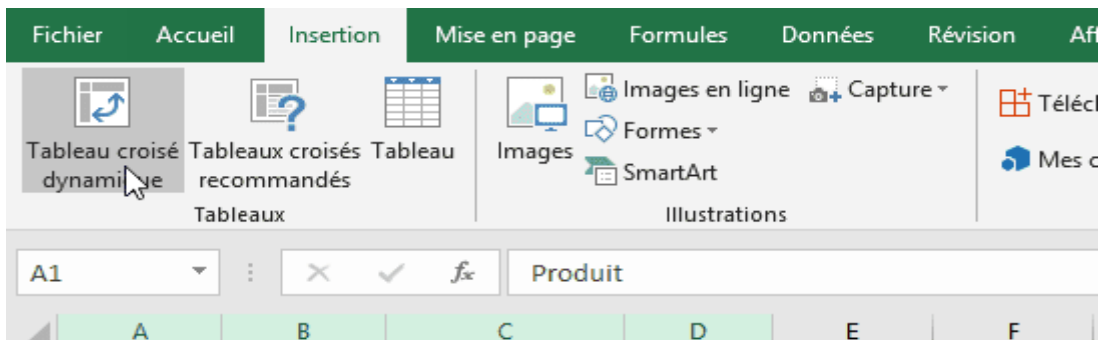
En d'autres termes, un **tableau croisé dynamique** extrait les données et les synthétise pour leur donner du sens, le tout sans entrer la moindre formule.

I-2 Création d'un Tableau Croisé Dynamique

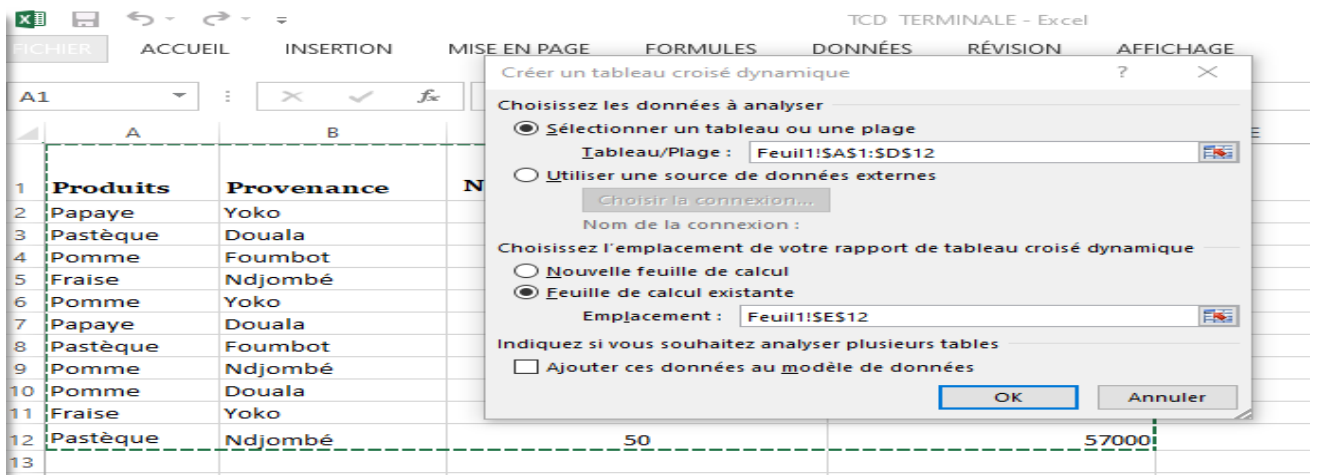
Nous allons Créer un tableau croisé dynamique à partir de la liste ci-dessous. Sélectionnez toute la liste de données dans le fichier proposé. N'oubliez pas de sélectionner les étiquettes de colonnes, c'est très important.

| | A | B | C | D | E |
|----|-----------------|-------------------|---------------------------------|-------------------------------|---|
| 1 | Produits | Provenance | Nombre d'unités achetées | Prix par unité en FCFA | |
| 2 | Papaye | Yoko | 49 | 25000 | |
| 3 | Pastèque | Douala | 20 | 35000 | |
| 4 | Pomme | Foumbot | 36 | 9000 | |
| 5 | Fraise | Ndjombé | 18 | 6000 | |
| 6 | Pomme | Yoko | 28 | 12000 | |
| 7 | Papaye | Douala | 94 | 67000 | |
| 8 | Pastèque | Foumbot | 89 | 87000 | |
| 9 | Pomme | Ndjombé | 47 | 24000 | |
| 10 | Pomme | Douala | 67 | 38000 | |
| 11 | Fraise | Yoko | 19 | 7000 | |
| 12 | Pastèque | Ndjombé | 50 | 57000 | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |
| 17 | | | | | |
| 18 | | | | | |
| 19 | | | | | |
| 20 | | | | | |
| 21 | | | | | |
| 22 | | | | | |

Dans le menu « Insertion », cliquez sur « Tableau croisé dynamique »



Dans la boîte de dialogue qui apparaît, définir la plage (Feuil1!\$A\$1:\$D\$12) et l'emplacement puis cliquer sur OK



Sélectionner les champs.

Deux champs sont sélectionnés **provenance** et **prix par unité en FCFA**

The screenshot shows an Excel spreadsheet with a PivotTable. The PivotTable is based on the following data:

| Étiquettes de lignes | Somme de Prix par unité en FCFA |
|----------------------|---------------------------------|
| Douala | 140000 |
| Foumbot | 96000 |
| Ndjombé | 87000 |
| Yoko | 44000 |
| Total général | 367000 |

The 'Champs de tableau croisé dynamique' task pane shows the following configuration:

- Champs à inclure dans le rapport:
 - Produits
 - Provenance
 - Nombre d'unités achetés
 - Prix par unité en FCFA
- PLUS DE TABLES...
- Faites glisser les champs dans les zones voulues ci-dessous:
 - FILTRES: (vide)
 - COLONNES: (vide)
 - LIGNES: Provenance
 - VALEURS: Somme de Prix par unité en FCFA
- Différer la mise à jour de la disposition
- METTRE À JOUR

Trois champs sont sélectionnés provenance, produits et prix par unité en FCFA

The screenshot shows an Excel spreadsheet with a PivotTable. The PivotTable is based on the following data:

| Étiquettes de lignes | Somme de Prix par unité en FCFA |
|----------------------|---------------------------------|
| Douala | 140000 |
| Papaye | 67000 |
| Pastèque | 35000 |
| Pomme | 38000 |
| Foumbot | 96000 |
| Pastèque | 87000 |
| Pomme | 9000 |
| Ndjombé | 87000 |
| Fraise | 6000 |
| Pastèque | 57000 |
| Pomme | 24000 |
| Yoko | 44000 |
| Fraise | 7000 |
| Papaye | 25000 |
| Pomme | 12000 |
| Total général | 367000 |

The 'Champs de tableau croisé dynamique' task pane shows the following configuration:

- Champs à inclure dans le rapport:
 - Produits
 - Provenance
 - Nombre d'unités achetés
 - Prix par unité en FCFA
- PLUS DE TABLES...
- Faites glisser les champs dans les zones voulues ci-dessous:
 - FILTRES: (vide)
 - COLONNES: (vide)
 - LIGNES: Provenance, Produits
 - VALEURS: Somme de Prix par unité en FCFA
- Différer la mise à jour de la disposition
- METTRE À JOUR

Quatre champs sont sélectionnés provenance, produits, nombre d'unités achetés et prix par unité en FCFA

| Étiquettes de lignes | Somme de Prix par unité en FCFA | Somme de Nombre d'unités achetés |
|----------------------|---------------------------------|----------------------------------|
| Douala | 140000 | 181 |
| Papaye | 67000 | 94 |
| Pastèque | 35000 | 20 |
| Pomme | 38000 | 67 |
| Foumbot | 96000 | 125 |
| Pastèque | 87000 | 89 |
| Pomme | 9000 | 36 |
| Ndjombé | 87000 | 115 |
| Fraise | 6000 | 18 |
| Pastèque | 57000 | 50 |
| Pomme | 24000 | 47 |
| Yoko | 44000 | 96 |
| Fraise | 7000 | 19 |
| Papaye | 25000 | 49 |
| Pomme | 12000 | 28 |
| Total général | 367000 | 517 |

I-3 MODIFIER UN TABLEAU CROISE DYNAMIQUE

Le TCD généré par Excel n'est pas forcément celui attendu, vous pouvez donc le modifier autant que nécessaire.

Pour permuter les lignes et les colonnes d'un tableau croisé dynamique il faut:

- sélectionnez l'un des champs à permuter,
- Sélection du premier champ à permuter
- Le faire glisser à droite du champ situé en colonne, puis relâchez le bouton de la souris.
- Le champ apparaît alors en colonne

Pour supprimer un champ il faut:

- Sélectionner le champ à supprimer
- Le faire glisser en dehors du tableau, puis relâchez la souris.

II- Reproduire un bulletin de notes

Pour reproduire un bulletin de notes sous excel il faut : lancer excel, insérer les notes par matière et les coefficients, calculer la note coefficientée, calculer la somme des coefficients, des notes coefficientées et la moyenne.

Pour calculer la note coefficientée la formule est : =B6*C6 ... =BI6*CI6

Pour calculer la somme des coefficients =SOMME(C6:CI6)

Pour calculer Calcul de la somme des notes coefficientées la formule est : =SOMME(D6:DI6)

Pour calculer la moyenne de l'élève, nous allons utiliser **la fonction MOYENNE** qui permet de calculer la moyenne de plusieurs cellules. Nous voulons calculer la moyenne de l'élève Melibi la formule est : **=MOYENNE(D17/C17)**

L'image suivante vous montre le bulletin de note de l'élève Melibi.

| BULLETIN DE NOTES | | | | | |
|--------------------------------|-----------------|--------------|--------------------|----------|-------------|
| Nom de l'élève: | Melibi Celestin | N° | 14 | Classe : | Terminale D |
| Matières | Note / 20 | Coefficient | Note X Coefficient | | |
| mathématiques | 17 | 4 | 68 | | |
| Physique | 15 | 2 | 30 | | |
| Chimie | 12 | 2 | 24 | | |
| Informatique | 16 | 2 | 32 | | |
| Science naturelle | 17 | 3 | 51 | | |
| Français | 12 | 2 | 24 | | |
| Philosophie | 13 | 2 | 26 | | |
| Anglais | 15 | 3 | 45 | | |
| Histoire & Géographie | 18 | 2 | 36 | | |
| Education civique et morale | 16 | 2 | 32 | | |
| Education physique et sportive | 15 | 2 | 30 | | |
| Somme | | 26 | 398 | | |
| Moyenne générale | | 15,31 | | | |

Pour calculer le rang de l'élève nous allons utiliser **la fonction Excel RANG** qui détermine le rang d'une valeur en fonction d'un ensemble de valeurs (pour classer des valeurs par exemple). La formule est :

= RANG(Valeur ; ensemble_de_valeurs ; ordre)

Pour calculer le rang de l'élève Melibi il faut appliquer la formule :

= RANG(B9; B9:BI2; 0).

0 : Pour l'ordre croissant et **I** : Pour l'ordre décroissant

Avant de recopier, ajouter des \$ à la plage de cellules pour éviter qu'elle ne soit décalée : **= RANG(B9; B\$9:B\$I2; 0)** puis étirez la formule vers le bas.

LISTE DES NOTES TERMINALE D - Excel

FICHIER ACCUEIL INSERTION MISE EN PAGE FORMULES DONNÉES RÉVISION AFFICHAGE Connexion

C9 : =RANG(B9;B\$5:B\$12;0)

| | A | B | C | D | E |
|----|------------------------|---------------------|-------------|---|---|
| 1 | LYCEE DE NKOLKOUGDA | | 2020 - 2021 | | |
| 2 | | RANG DES ELEVES | | | |
| 3 | Classe : Terminale D | | | | |
| 4 | Nom de l'élève: | Moyenne / 20 | Rang | | |
| 5 | Ayimngeh Hermine | 15,2 | 3 | | |
| 6 | Bella sylvie | 15 | 4 | | |
| 7 | Cokoumbo lucie | 5 | 8 | | |
| 8 | kenlefak Tatiana | 14 | 6 | | |
| 9 | Melibi Celestin | 15,31 | 2 | | |
| 10 | Nana François | 7 | 7 | | |
| 11 | Panta Leonard | 16,19 | 1 | | |
| 12 | Wadibo Moustapha | 15 | 4 | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |
| 17 | | | | | |
| 18 | | | | | |
| 19 | | | | | |
| 20 | | | | | |
| 21 | | | | | |
| 22 | | | | | |
| 23 | | | | | |

Feuil1

PRÊT 100 %

BULLETIN DE NOTE - Excel

FICHIER ACCUEIL INSERTION MISE EN PAGE FORMULES DONNÉES RÉVISION AFFICHAGE Connexion

C18 : =MOYENNE(D17/C17)

| | A | B | C | D | E | F |
|----|--------------------------------|------------------------|--------------------|-----------------------------|-------------|---|
| 1 | LYCEE DE NKOLKOUGDA | | 2020 - 2021 | | | |
| 2 | | BULLETIN DE NOTES | | | | |
| 3 | Nom de l'élève: | Melibi Celestin | N° 9 | Classe : Terminale D | | |
| 4 | | | | | | |
| 5 | Matières | Note / 20 | Coefficient | Note X Coefficient | | |
| 6 | mathematiques | 17 | 4 | 68 | | |
| 7 | Physique | 15 | 2 | 30 | | |
| 8 | Chimie | 12 | 2 | 24 | | |
| 9 | Informatique | 16 | 2 | 32 | | |
| 10 | Science naturelle | 17 | 3 | 51 | | |
| 11 | Français | 12 | 2 | 24 | | |
| 12 | Philosophie | 13 | 2 | 26 | | |
| 13 | Anglais | 15 | 3 | 45 | | |
| 14 | Histoire & Géographie | 18 | 2 | 36 | | |
| 15 | Education civique et morale | 16 | 2 | 32 | | |
| 16 | Education physique et sportive | 15 | 2 | 30 | | |
| 17 | Somme | | 26 | 398 | | |
| 18 | Moyenne générale | | 15,31 | | | |
| 19 | | | | | | |
| 20 | RANG | | | | 2ème | |
| 21 | | | | | | |
| 22 | | | | | | |

Feuil1

PRÊT 100 %

III- Impression d'une feuille de calculs.

Pour imprimer une feuille de calcul il faut:

- I. Cliquez sur l'onglet **Fichier** du ruban en haut à gauche.

2. Sur la barre latérale verte, cliquez sur le bouton **Imprimer**, puis sur le bouton **Mise en page** en bleu au bas de la fenêtre d'Excel.
3. Sur la boîte de dialogue de **Mise en page** dans la section **Échelle**, cochez l'option **Ajuster** en veillant à ce que celle-ci soit bien paramétrée sur **1 page(s) en largeur sur 1 en hauteur**.
Note : nous vous conseillons de cocher l'option **Paysage** dans section **Orientation** afin d'optimiser l'utilisation de l'espace de votre feuille lors d'une impression.
4. Toujours sur la boîte de dialogue de **Mise en page**, cliquez sur l'onglet **Marges**, et dans la section **Centrer sur la page**, cochez les options *Horizontalement et verticalement*
5. Cliquez sur le bouton **OK** pour valider votre mise en page. Vous pouvez à présent observer un aperçu de ce à quoi ressemblera votre impression.

Si vous souhaitez pousser encore plus loin les options de mise en page, vous pouvez vous rendre sur l'onglet **Mise en page** du ruban. Celui-ci regorge d'options afin d'imprimer correctement votre feuille de calcul. Il est possible d'imprimer les en-têtes, le quadrillage de modifier les marges ou encore de changer l'orientation d'impression (portrait/paysage).

EXERCICE

- 1- Définir tableau croisé dynamique
- 2- Donner la procédure de création d'un TCD
- 3- Reproduire dans un fichier excel le bulletin de note ci dessus
- 4- Donner la formule de calcul de la somme, de la moyenne et du rang d'un élève
- 5- Donner la procédure d'impression d'un fichier excel

Jeu bilingue des mots contenu dans la colonne savoirs essentiels du programme

spreadsheet : tableur

cells : cellule

Workbook : classeur

range of cells: Plage de cellule

worksheet : worksheet

Compétences visées : A la fin de cette leçon, l'élève doit être capable de :

- ✓ Décrire les différents types de réseaux informatiques en fonction de l'étendue géographique.
- ✓ Décrire les topologies physiques (bus, étoile, anneau) et logiques (Ethernet, Token ring).
- ✓ Décrire les principales architectures réseaux (Peer to peer et client/serveur)

Situation-problème :

Votre établissement vient d'être doté des équipements informatiques pour votre nouveau laboratoire informatique. Pour mieux faciliter les travaux de l'administration d'une part et des enseignants d'autre part, votre proviseur vous sollicite pour mettre lesdits équipements en réseau.

I- DEFINITIONS

Un **réseau** se définit de façon générale comme un ensemble d'entités (objet, personne, chose) interconnectées les unes aux autres pour faire circuler les éléments matériels ou immatériels. Selon le type d'objet en liaison, on distinguera : le réseau routier, le réseau téléphonique, le réseau des malfaiteurs, le réseau de transport, etc.

Un **réseau informatique** est l'ensemble des équipements (ordinateurs, périphériques) reliés entre eux et échangeant des informations sous forme de données numériques (valeurs binaires: 0 et 1)

II- DIFFERENTS TYPES DE RESEAUX

On distingue quatre (04) principaux types de réseaux classés selon leur taille, leur vitesse de transfert de données ainsi que leur étendue :

1. PAN (Personal Area Network ou Réseau personnel en français)

Il permet de relier des équipements électroniques personnels. Exemple: Transfert de fichier entre deux ou plusieurs téléphones portables.

2. LAN (Local Area Network ou Réseau local en français)

C'est un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite surface géographique (1km) par un réseau. La vitesse de transfert des données dans ce réseau peut varier entre 10 Mb et 1Gb. Sa taille peut atteindre jusqu'à 100 voire 1000 utilisateurs (ordinateurs).

3. MAN (Metropolitan Area Network ou Réseau métropolitain)

Il interconnecte plusieurs LAN géographiquement proches à des débits importants. Ainsi un MAN permet à deux nœuds distants de communiquer comme s'ils faisaient partie d'un même réseau local.

4. WAN (Wide Area Network ou Réseau étendu en français)

Il interconnecte plusieurs réseaux locaux et métropolitains à travers de grandes surfaces géographiques. Les WAN fonctionnent grâce aux routeurs qui permettent de choisir le trajet le plus approprié pour atteindre un réseau du réseau.

NB : Il existe par ailleurs, deux autres types de réseaux à savoir : TAN (Tiny Area Network) semblables au LAN mais moins étendu (2 à 3 machines) et le CAN (Campus Area Network) identique au MAN avec une bande passante maximale entre tous les LAN du réseau.

III- TOPOLOGIES ET ARCHITECTURES RESEAUX

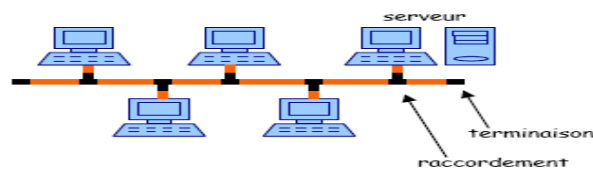
I. Topologies réseaux

On appelle **topologie réseau** la manière dont les machines sont interconnectées et la méthode utilisée pour permettre aux équipements de communiquer entre eux. Il existe deux (02) types de topologies :

a. Topologie physique

La **topologie physique** désigne la disposition spatiale des équipements dans un réseau. On distingue principalement trois (03) types de topologies physiques :

✓ La topologie en bus



✓ La topologie en étoile



✓ La topologie en anneau



✓ En plus de ces trois (03) topologies, on peut avoir : la topologie en maille, la topologie en arbre, la topologie hybride...

b. Topologie logique

La **topologie logique** décrit la façon avec laquelle les informations transitent dans le réseau.

2. Architectures réseaux

L'**architecture** est l'ensemble des mécanismes qu'utilise le réseau pour faire circuler les informations sur le réseau. Cette architecture détermine comment les ordinateurs doivent procéder pour transférer des informations sur les supports de transmission. Une architecture décrit également comment les données circulent dans une topologie de réseau. Il existe plusieurs architectures de réseau. On peut citer: L'architecture Ethernet; l'architecture FDDI; l'architecture Token Ring; l'architecture AppleTalk.

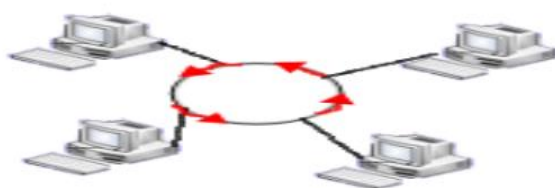
a. L'architecture Ethernet

Dans cette architecture, lorsqu'un ordinateur veut envoyer une information, il vérifie d'abord que la liaison est libérée c'est-à-dire qu'aucun autre ordinateur n'est en train d'émettre. Dans ce cas, il peut commencer à émettre ces informations. Si plusieurs ordinateurs cherchent à émettre simultanément, il se produit une collision. Ce type d'architecture est utilisé dans les topologies de réseau en bus et en étoile.



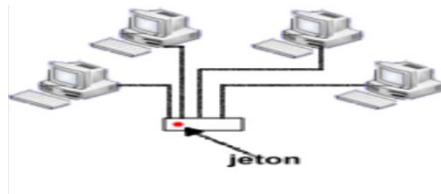
b. L'architecture FDDI

FDDI signifie Fiber Distributed Data Interface ou interface de données à distribution par fibre. Cette architecture utilise un câble en fibre optique et repose sur une topologie en anneau. Dans cette architecture, un jeton gère l'accès au support de réseau. Le jeton circule d'un ordinateur à un autre. Lorsqu'un ordinateur veut envoyer un message, il attend que le jeton arrive à lui, l'attrape et commence à envoyer ses informations. Lorsqu'un ordinateur fini d'envoyer les informations (ou ne veut pas envoyer d'information), il laisse passer le jeton qui circule à nouveau sur l'anneau.



c. L'architecture Token Ring

Comme l'architecture FDDI, l'architecture Token Ring utilise également un système à jeton pour gérer l'accès au support de réseau. Cette architecture est câblée suivant une topologie en étoile. Le jeton est appelé MAU (Multistation Access Unit) est utilisé comme point de connexion central pour tous les ordinateurs du réseau.



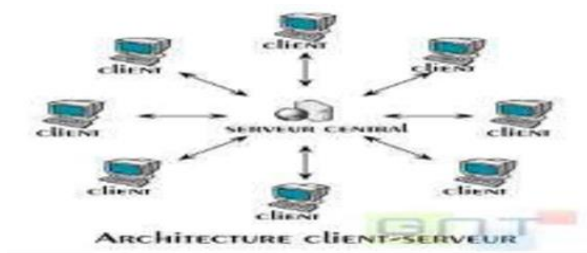
d. L'architecture AppleTalk

L'architecture AppleTalk est l'architecture de réseau utilisé par les ordinateurs Macintosh, ou Mac d'Apple. Elle emploie un système d'adressage dynamique pour déterminer l'adresse des ordinateurs du réseau. Cette architecture est similaire à l'architecture Ethernet, car lorsqu'un ordinateur veut envoyer des informations, celui se rassure d'abord que la voie est libre. Après quoi il envoie un message pour aviser aux autres machines qu'il est prêt à envoyer ces informations.

L'architecture fait également référence aux relations de dominance ou d'égalité qui existent entre les machines d'un réseau. Nous avons l'architecture client-serveur et l'architecture peer to peer (d'égal à égal)

✓ L'architecture client-serveur

Il s'agit d'une architecture où on a un serveur chargé de fournir des services et répondre aux requêtes des machines appelées clients.

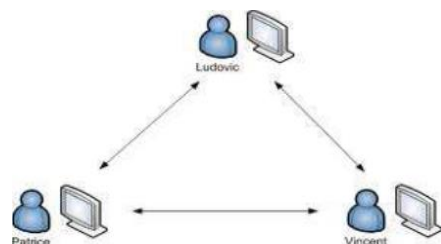


Avantages : Les ressources sont centralisées au niveau du serveur ; plus sécurisée ; évolutif : l'ajout de nouveaux clients est facile

Inconvénients : Le serveur est le maillon faible ; s'il tombe en panne c'est toute l'architecture qui est paralysée ; le coût de la mise en place de cette architecture est élevé à cause du prix des serveurs et de coût de maintenance.

✓ L'architecture client-client (peer to peer)

Il s'agit d'une architecture où chaque ordinateur joue à la fois le rôle de client et celui du serveur. Ici toutes les ressources sont partagées et aucune machine n'est supérieure à l'autre.



Avantage : facile à mettre en place, peu coûteuse

Inconvénients : peu sécurisée, peu robuste

Jeu bilingue

Réseau = Network, anneau=ring, étoile=star

Peer to peer = égal à égal

EXERCICES DE REINVESTISSEMENT

Exercice 1 :

Comparer la topologie en maille et celle en bus. Donnez des équivalents dans la vie de ces topologies.

Exercice 2 :

Pour chacun des cas suivants dire quel est l'architecture en place :

Envoie de fichier par Bluetooth; ordinateur connecté au serveur de Facebook; ordinateur sur le serveur de la CRTV ; deux joueurs de jeux en ligne

Compétences : L'élève doit être capable de :

- Identifier les supports physiques d'interconnexion ;
- Donner le rôle des équipements réseaux suivants : modem, hub, Switch, routeur ;
- Modifier les paramètres IPV4 d'une carte réseau ;
- Identifier la carte réseau (NIC) et donner son rôle
- Décrire les principaux types de câble (paires torsadées, câble coaxial, fibre optique);

Situation problème

Le proviseur de votre lycée sollicite votre aide pour créer une salle informatique et mettre tous les équipements de cette salle informatique en réseau. Quels sont les équipements qu'il aura absolument besoin d'acheter et donner le rôle de chaque équipement ? Plutard il désire connecte son réseau à internet et mettre cela à la portée du personnel du campus à l'aide du WIFI, quels équipements auraient-il encore besoin d'acheter ?

I. LES EQUIPEMENTS DE BASE D'UN RESEAU

I. Définition

Les équipements de base d'un réseau sont les composantes matériels et logiciel qui permettent de connecter deux ordinateurs en réseau.

On met les équipements informatiques en réseau dans le but de les faire communiquer ou de les faire partager une même ressource (imprimante, scanner, fichiers, ...). Les équipements requis pour créer un petit réseau sont : **Ordinateurs et périphériques (tels que les imprimantes), Système d'exploitation, Câbles réseaux, le serveur, carte réseau**

a. L'ordinateur

L'ordinateur est le principal élément d'un réseau. Pour être utiliser, il doit être équipé d'un équipement appelé carte réseau et d'un système d'exploitation. Le système d'exploitation a pour rôle de gérer le fonctionnement de l'ordinateur mais également la communication entre les autres composants du réseau.

b. La carte réseau

La carte réseau ou Network Interface Card (NIC) constitue l'interface entre l'ordinateur et le câble du réseau. Elle a pour fonction de préparer, d'envoyer et de contrôler les informations sur le réseau. La carte réseau offre une interface appelée port où viendra se connecter le câble réseau. Chaque carte réseau possède un adresse appelé adresse MAC. Exemple d'adresse MAC : **5E:FF:56:A2:AF:15**



c. Le serveur

On détermine deux types d'ordinateur connectés sur un réseau : les serveurs et les clients. Le serveur est un ordinateur très puissant disposant d'une grande capacité mémoire et d'une vitesse de calcul élevée. Il partage ses ressources (fichiers, périphériques de stockage, périphériques d'impression, ...) avec d'autres machines appelées clients.



d. Le câble réseau

Le câblage du réseau est l'ensemble des moyens destinés à créer une liaison de communication entre les différents équipements d'un réseau. Il est généralement constitué de câbles et de connectiques ou connecteurs.

Le **câble** permet de transporter les informations d'un ordinateur à un autre. Il existe plusieurs types et dont le choix dépend de la structure de réseau à mettre sur pied. Les principaux sont : **le câble à paire torsadée, le câble coaxial et la fibre optique**



fig 4

Le **connecteur** permet l'interconnexion des câbles, mais aussi le raccordement aux équipements (ordinateurs, imprimantes, ...). A chaque type de câble réseau correspond un connecteur spécifique. Les plus utilisées sont les **connecteurs RJ45** (fig. 5) pour le câble à paire torsadée non blindé, les connecteurs multi-modes pour un fibre optique.

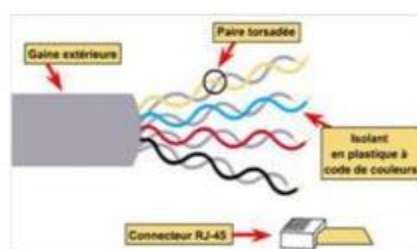
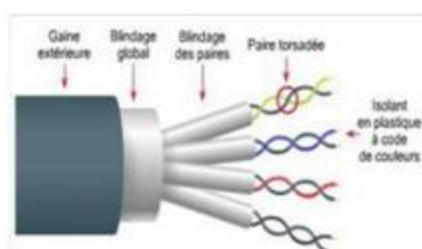


fig 5

Fig 5

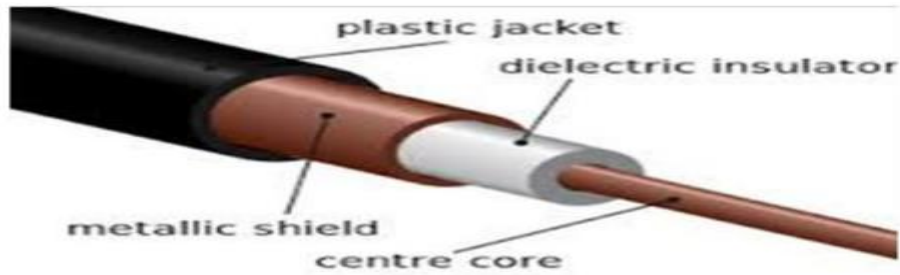
➤ *Câbles à Paires torsadées*

On distingue deux types de câble à paires torsadées : les câbles à paires torsadées blindés et les câbles à paire torsadées non blindés. Les blindés assurent une plus grande protection contre tous les types d'interférence externe.



➤ **Câble coaxial**

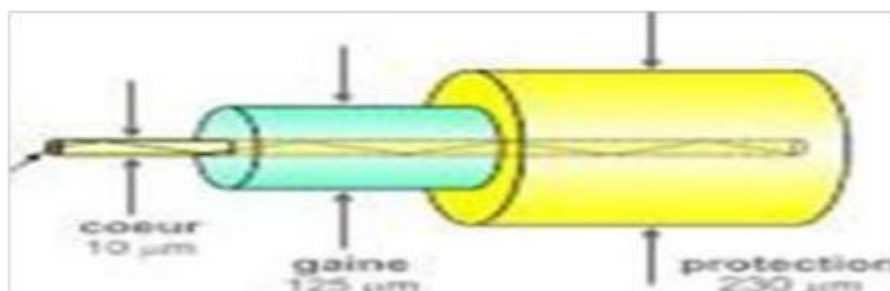
Le **câble coaxial**, parfois appelé **câble BNC** (**Bayonet-Neill Concelman** ou **British Naval Connector**), est un fil de cuivre relativement rigide noyé dans une housse en plastique isolante, elle est similaire au câble utilisé pour avoir des images sur les télévisions. Tout comme le câble à paires torsadées, le câble coaxial utilise le signal électrique pour la transmission des données.



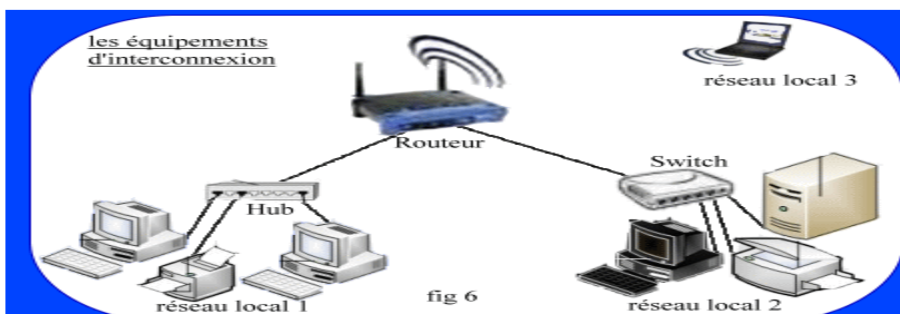
➤ fibre optique

La fibre optique conduit un signal fait de photons, c'est-à-dire de la lumière. La lumière est conduite par une fibre en verre entourée d'une gaine également en verre dont le rôle est de réfléchir la lumière et d'isoler le signal des perturbations extérieures. La fibre optique a beaucoup d'avantages par rapport au câble coaxial ou aux paires torsadées.

- La fibre optique permet des vitesses de transmission extrêmement rapides (jusqu'à 155 Mbps) très utiles pour le transfert d'images vidéo ou audio, et pour le multimédia en général.
- Puisque la fibre utilise la lumière et non l'électricité, elle est complètement imperméable aux perturbations électromagnétiques et peut ainsi transporter un signal sur plusieurs kilomètres sans aucune dégradation.



II. LES EQUIPEMENTS D'INTERCONNEXION DES RESEAUX



I. généralités

Un **réseau local** sert à interconnecter les ordinateurs d'une organisation, toutefois une organisation comporte généralement plusieurs réseaux locaux, il est donc parfois indispensable de les relier entre eux. Dans ce cas, des équipements spécifiques sont nécessaires.

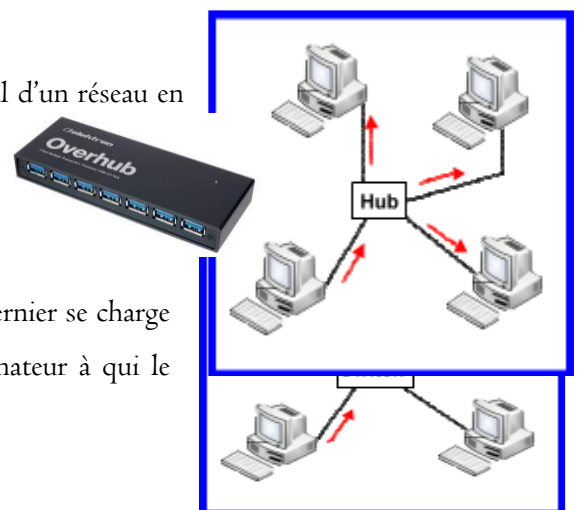
2. Définition

Un **équipement d'interconnexion** est un matériel qui permet de relier les ordinateurs d'un réseau ou plusieurs réseaux entre eux. Il existe plusieurs équipements d'interconnexion on peut citer :

- ✓ Hubs;
- ✓ modem
- ✓ Switchs ;
- ✓ Routeurs ;
- ✓ Passerelles ;

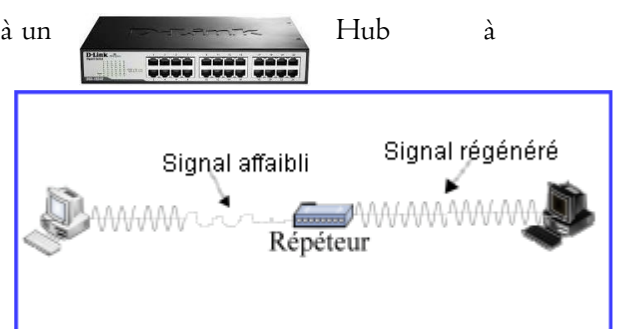
a. Le hub

Encore appelé **concentrateur**, le Hub est l'élément central d'un réseau en étoile. Il permet de faire communiquer tous les ordinateurs d'un réseau local en reliant tous les câbles à lui. Lorsqu'un ordinateur voudrait envoyer une information (texte, image, sons, etc...) à un autre ordinateur, celui l'envoie d'abord au commutateur, ce dernier se charge de le redistribuer à tous les autres ordinateurs, mais seul l'ordinateur à qui le message est destiné va ouvrir le message et l'utiliser.



b. Le Switch

Encore appelé **commutateur**, un Switch est similaire à un Hub à quelques différences près. Au sein d'un réseau local lorsque le Switch est l'élément central et lorsqu'il reçoit un message d'un ordinateur, celui-ci le renvoie directement à l'ordinateur concerné sans que les autres soient au courant.



c. Le répéteur

Un **répéteur** est un matériel ayant pour rôle de prendre le signal qu'il reçoit des équipements de réseau et de régénérer afin de préserver son intégrité le long d'un support de réseau plus long que la longueur maximale normalement.

d. La passerelle

Un réseau local sert à interconnecter les ordinateurs d'une entreprise, toutefois une entreprise peut comporter plusieurs réseaux locaux utilisant les moyens de communication (protocoles) différents. Dans

ce cas, il est indispensable de procéder à une conversion de protocoles pour relier ces réseaux locaux entre eux. Le périphérique indispensable dans ce cas s'appelle la **passerelle**.

e. **Le routeur**

Un **routeur** est un périphérique d'inter-réseau qui permet de relier plusieurs réseaux locaux situés à des distances plus ou moins éloignées. Il offre plusieurs interfaces de communication et est équipé d'un logiciel spécifique dont le rôle est de permettre la circulation de données d'un réseau à un autre de façon optimale.



f. **Le modem**

Le **modem** (modulateur-démodulateur), est un périphérique servant à communiquer avec des utilisateurs distants par l'intermédiaire d'un réseau analogique (comme une ligne téléphonique). Il permet par exemple de se connecter à Internet.



BILINGUAL GAME

| FRENCH | ENGLISH | FRENCH | ENGLISH |
|---------------|---------|------------|------------|
| Concentrateur | Hub | Pont | Bridge |
| Commutateur | Switch | Passerelle | Footbridge |

EXERCICE : EVALUATION DU SAVOIR

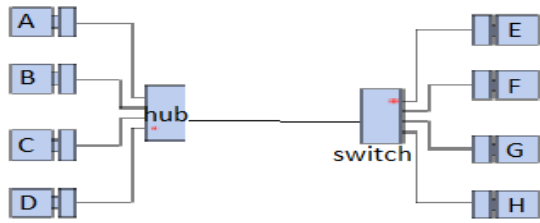
On vous donne le tableau suivant :

| COLONNE A | COLONNE B |
|------------------|----------------------------------------------------------------------|
| A) Routeur | a) Interconnecte les ordinateurs |
| B) Passerelle | b) Relie les réseaux de même technologie |
| C) Répéteur | c) Interconnecte les réseaux différents |
| D) Carte réseau | d) Relie les réseaux de protocole différent |
| E) Pont | e) Interface entre l'ordinateur et le câble réseau |
| F) Hub ou Switch | f) Amplifie les signaux afin de prolonger le support de transmission |

Associer a chaque élément de la colonne A sa fonction contenue dans la liste de la colonne B.

EXERCICE DE COMPETENCE

On considère le réseau suivant :



a) Donner la différence entre un Switch et un Hub ?

- b) Si A désire envoyer un message à C, quels sont les ordinateurs qui le recevront?
- c) Si A désire envoyer un message à G, quels sont les ordinateurs qui le recevront?
- d) donner le nom de l'équipement de ce réseau qu'il faut modifier pour que seul G reçoive le message de A

Compétences visées :

- Définir protocole réseau
- Donner le rôle des protocoles suivants (HTTP, IP, FTP, SMTP, POP3, DHCP)

Situation problème : Votre petit frère en classe de 4^e sollicite votre aide pour un exposé. Son enseignant d'informatique lui a demandé de préparer un exposé dont le thème porte sur les protocoles Internet.

Consigne :

1. Définir protocole réseau
2. Donner quelques exemples de protocoles ainsi que leurs rôles

INTRODUCTION

Un protocole est un langage commun utilisé par l'ensemble des acteurs de la communication pour échanger des données. La plupart des réseaux utilisent la pile de protocole TCP/IP (Transmission Control Protocol / Internet Protocol). Sur Internet il existe plusieurs protocoles (langages entre les ordinateurs) permettant d'accéder à différents services.

I. Définition

Un protocole réseau est un ensemble de règle qui régissent les échanges de données entre les différents acteurs du réseau.

2. Quelques exemples de protocole

a. Protocole http

Le protocole **HTTP** (HyperText Transfer Protocol) littéralement protocole de transfert hypertexte est un protocole de communication client-serveur développé pour le World Wide Web (WWW). Ce protocole permet de consulter les pages web

b. Protocole IP

Pour pouvoir envoyer des données via le réseau Internet, vous aurez besoin d'une adresse correcte (adresse IP). Les adresses IP sont attribuées de façon automatique ou manuelle au périphérique réseau lors de la configuration. Dans ce processus le protocole IP (Internet Protocol) joue un rôle important et il détermine également le chemin à prendre par les paquets dans le réseau.

c. Protocole FTP

Le protocole FTP (File Transfer Protocol) ou protocole de transfert de fichier est un protocole de communication destiné au partage de fichiers sur un réseau TCP/IP. Il permet à l'ordinateur de copier les fichiers vers un autre ordinateur du réseau.

d. Protocole SMTP

Le protocole SMTP (Simple Mail Transfer Protocol) est un protocole de communication utilisé pour le transfert du courrier électronique vers les serveurs de messagerie électronique.

e. Protocole POP3

Le protocole POP3 (Post Office Protocol) littéralement protocole de bureau de poste est un protocole qui permet de récupérer les courriers électroniques situés sur un serveur de messagerie électronique.

f. Protocole DHCP

Le protocole DHCP (Dynamic Host Configuration Protocol) est un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres IP d'une station ou d'une machine, notamment en lui attribuant automatiquement une adresse IP et un masque de sous réseau.

CONCLUSION

Pour qu'il y ait communication il faudrait respecter un certain nombre de règles, il en est de même pour des objets connectés en réseau. C'est le rôle du protocole de définir ces règles.

Jeu bilingue : Traduire en français ou en anglais les mots ou expression du tableau ci-dessous :

| Français | Anglais |
|----------|---------|
| Fichier | File |
| Réseau | Network |
| Adresse | Address |

EXERCICE

Votre professeur d'informatique vous donne des exposés à réaliser comme devoir. Il vous communique une adresse mail ou vous devez envoyer votre travail une fois terminé.

1. Afin d'effectuer un travail sérieux vous décidez d'utiliser le réseau Internet pour vos recherches.
 - a- Quel est le protocole mis en évidence dans ce cas de figure ?
 - b- Donner son rôle.
2. Votre travail achevé vous décidez d'envoyer le document à votre enseignant par mail.
 - a. Quel est le protocole mis en évidence ?
 - b. Votre enseignant reçoit une notification de sa messagerie électronique, quel protocole lui permettra d'accéder à votre mail ?

UE II : L'ADRESSAGE RESEAU 2H

Compétence visée : configurer un ordinateur dans un réseau

Situation problème

En visite dans le bureau de votre père, vous remarquez que pour faciliter leur travail, ses deux collaborateurs et lui ont connecté leurs ordinateurs en réseau. Pourtant, ce jour-là, votre père se plaint qu'il n'arrive pas à voir les ordinateurs de ses collaborateurs.

Q1 : Quel peut être le problème ?

Réponse attendue : vérifier l'adressage IP

INTRODUCTION

Un réseau doit pouvoir permettre à n'importe quel hôte de se mettre en relation avec n'importe quel autre. Afin qu'il n'y ait pas d'ambiguïté pour la reconnaissance des hôtes possibles, il est absolument nécessaire d'admettre un principe général d'identification :

1. Le nom de la machine distante,
2. son adresse,
3. la route à suivre pour y parvenir.

Le nom dit « qui » est l'hôte distant, l'adresse nous dit « où » il se trouve et la route « comment » on y parvient.

L'adresse IP (version 4) identifie l'emplacement d'un hôte sur le réseau, elle doit être unique et présenter un format normalisé sous forme d'un nombre de 32 bits qui permet à la fois l'identification de chaque hôte et du réseau auquel il appartient.

Q2 : Quel est le type de leur réseau informatique ?

Réponse attendue : un LAN, un réseau privé

Q3 : S'ils décident de connecter leur réseau à Internet, vont-ils utiliser les mêmes adresses IP ?

Réponse attendue : non

II- LES TYPES D'ADRESSES

On distingue deux types d'adresses IP.,

1. **Les adresses privées :** ce sont des adresses que tout administrateur de réseau peut s'attribuer librement pourvu qu'elles ne soient pas routées sur l'Internet. Elles s'utilisent en local.

Exemple : 10.0.0.0 ; 172.16.0.0 à 172.31.0.0 ; 192.168.0.0 à 192.168.255.0

2. **Les adresses publiques :** elles sont délivrées par une structure mondiale qui en assure l'unicité. Les FAI (Fournisseurs d'Adresses Internet) en assurent la distribution. C'est L'ICANN (Internet Corporation for Assigned Names and Numbers) qui est chargé au niveau mondial de la gestion de l'espace d'adressage IP. Il définit les procédures d'attribution et de résolution de conflits dans l'attribution des adresses, mais délègue le détail de la gestion de ces ressources à des instances régionales puis locales, dans chaque pays,

appelées « Regional Internet Registries » ou RIR. Il y a actuellement cinq « Regional Internet Registries » opérationnels :

- l'APNIC pour la région Asie-Pacifique
- l'ARIN pour l'Amérique
- le RIPE NCC pour l'Europe
- l'AfriNIC pour l'Afrique
- LACNIC pour l'Amérique Latine

En dehors de ces deux principaux types d'adresses, on distingue également :

- Les adresses « **Unicast** » : elles permettent d'identifier un équipement IP de façon unique. Exemple PC2 ordinateur Patron, ...
- Les adresses « **Multicast** » : elles permettent d'envoyer des messages vers un groupe d'équipements IP n'appartenant pas à un même réseau. Exemple : les parents d'élèves d'un Lycée
- Les adresses « **Broadcast** » : ce sont des adresses de diffusion vers toutes les adresses IP d'un même sous-réseau.

Q4 : Après avoir contacté un FAI pour se connecter à Internet, ce dernier les inclut dans son réseau. Ils deviennent donc un sous-réseau de son réseau. Comment pourront-ils reconnaître l'adresse de leur réseau dans le grand réseau ?

Réponse attendue : guider les apprenants vers la forme de l'adresse IP.

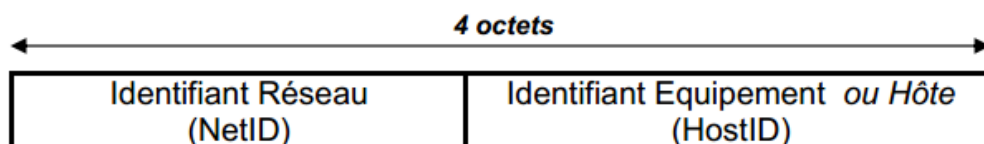
II. ANATOMIE D'UNE ADRESSE IP

I. La structure d'une adresse IP

Une adresse IP est un nombre de 32 bits que l'on a coutume de représenter sous forme de quatre nombre entiers compris entre 0 et 255, et séparés par des points. Exemple : 192.255.21.32

La partie réseau de l'adresse IP vient toujours en tête, la partie hôte est donc toujours en queue. L'intérêt de cette représentation est immédiat quand on sait que la partie réseau et donc la partie hôte sont presque toujours codées sur un nombre entier d'octets.

L'adresse IP est composée de deux parties. Une partie réseau et une partie qui identifie l'équipement connecté au réseau.



N.B : La partie réseau peut prendre 1, 2 ou 3 octets.

2. les classes d'adresses

On a principalement les trois classes d'adresses suivantes :

- Classe A : Un octet réseau, trois octets d'hôtes.
- Classe B : Deux octets réseau, deux octets d'hôtes.
- Classe C : Trois octets réseau, un octet d'hôte.

Cependant, il existe deux autres classes : la classe E et la classe D comme l'illustre le tableau ci-dessous.

Décomposition en classes

| Classe | Masque de sous réseau par défaut | Adresse réseau | Nombre de réseaux | Nombre d'hôtes |
|--------|----------------------------------|---------------------------|-------------------------|----------------|
| A | 255.0.0.0 | 1.0.0.0 à 126.0.0.0 | 126 (= $2^7 - 2$) | 16 777 214 |
| B | 255.255.0.0 | 128.0.0.0 à 191.255.0.0 | 16 384 (= 2^{14}) | 65 534 |
| C | 255.255.255.0 | 192.0.0.0 à 223.255.255.0 | 2 097 152 (= 2^{21}) | 254 |
| D | Non défini | 224.0.0.0 à 239.255.255.0 | | |
| E | Non défini | 240.0.0.0 à 255.255.255.0 | | |

Pour distinguer les classes A, B, C, D et E il faut examiner les bits de poids fort de l'octet de poids fort. Ce premier octet désigne le numéro de réseau (NetID) et les 3 autres correspondent à l'adresse de l'hôte (HostID).

- Si le premier bit est 0, l'adresse est de classe A. On dispose de 7 bits pour identifier le réseau et de 24 bits pour identifier l'hôte. On a donc les réseaux de 1 à 127 et 224 hôtes possibles, c'est à dire 16 777 216 machines différentes (de 0 à 16 777 215).

Remarque : l'adresse réseau 0.0.0.0 n'existe pas et le NetID « 127 » est réservée pour les communications en boucle locale (loopback), ce qui réduit de deux unités le nombre de machines nommables. Il reste donc seulement 16 777 214 machines adressables dans une classe A !

- Si les deux premiers bits sont 10, l'adresse est de classe B. Il reste 14 bits pour identifier le réseau et 16 bits pour identifier la machine. Ce qui fait $2^{14} = 16 384$ réseaux (128.0 à 191.255) et 65 534 (65 536 - 2) machines.

- Si les trois premiers bits sont 110, l'adresse est de classe C. Il reste 21 bits pour identifier le réseau et 8 bits pour identifier la machine. Ce qui fait $2^{21} = 2 097 152$ réseaux (de 192.0.0 à 223.255.255) et 254 (256 - 2) machines.

3. les adresses réservées

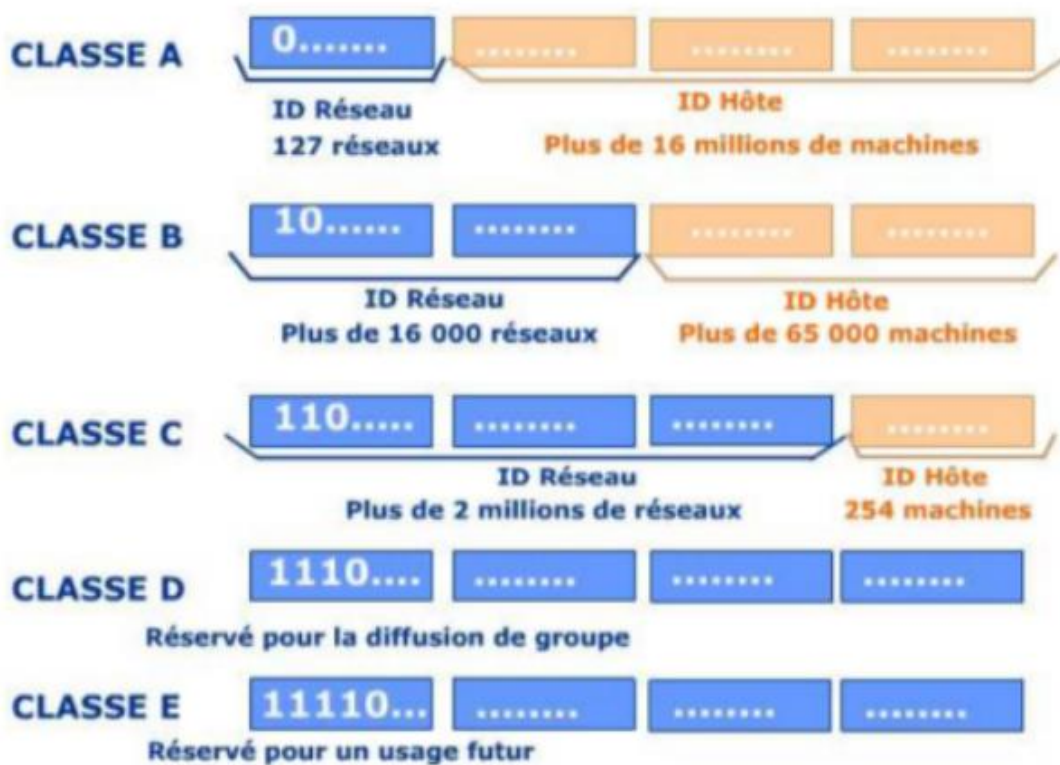
Il existe un certain nombre d'adresses IP réservées :

- hostid = 0 désigne le réseau lui-même : l'hostid égal à 0 ne sera jamais affecté à un hôte mais il désigne le réseau lui-même. Exemple : 192.145.56.0 est un réseau de classe C dont l'hostid est à 0 donc cette adresse désigne le réseau lui même. • 0.0.0.0 désigne l'hôte lui même

Lorsque tous les bits d'une adresse IP sont à 0, cela signifie "cet hôte-ci sur ce réseau". Cette adresse spéciale est utilisée par un hôte afin d'obtenir une adresse IP de manière dynamique dans le cas du protocole BOOTP.

➤ Tous les bits de l'hostid = 1 indique une diffusion dirigée. Lorsque tous les bits de l'hostid sont égaux à 1, on est en présence non pas d'une adresse d'hôte mais d'une adresse de diffusion dirigée (direct broadcast) c'est à dire un message destiné à tous les hôtes d'un réseau sans exception. Exemple : 192.145.56.255 est une adresse de classe C dont la partie réservée à l'hostid est égale à 255 donc pour laquelle tous les bits sont à 1, on est donc en présence d'un message destiné à l'ensemble des hôtes du réseau 192.145.56.0.

➤ 255.255.255.255 = diffusion limitée. Une diffusion limitée (limited broadcast) est un message qui est envoyé à tous les hôtes du réseau dont fait partie l'expéditeur. La diffusion limitée est représentée par l'adresse spéciale 255.255.255.255.



Q5 : Quand on regarde la structure d'une adresse IP, à votre avis combien d'adresses IP potentielles peut-on avoir en tout ? Est-ce suffisant pour le monde entier ?

Réponses attendues : $2^{32} = 4\,294\,967\,296$. Ce nombre apparaît très petit au vu de la population mondiale sachant qu'une personne peut avoir plusieurs équipements tous connectés à Internet

CONCLUSION

L'adresse de version 4 (IPv4) mis sur pied en Septembre 1981 est encore actuellement la plus utilisée. Mais, vu le développement de l'Internet depuis ce temps, cet adressage c'est avéré incapable de

prendre en charge tous les équipements du réseau. C'est pourquoi il a été mis sur pied en Décembre 1998 la spécification d'Internet Protocol Version 6 (Ipv6). Dans cette version 6: les adresses sont codées sur 128 bits. Une adresse Ipv6 est notée par un groupe de 4 chiffres hexadécimaux séparés par ':' exemple : FE80:0000:0000:0000:020C:76FF:FE21:1C3B. Ceci augmente considérablement le nombre d'adresses publiques.

Exercices :

1. Convertissez les adresses IP suivantes en binaire : • 145.32.59.24 • 200.42.129.16
2. Trouvez la classe des adresses IP suivantes : • 10000000.00001010.11011000.00100111 • 11101101. 1000011.00001110.01011111 • 01001010.00011011.10001111.00010010
3. Pour chaque adresse, souligner la partie demandée : • PARTIE RESEAU : 1.102.45.177 • PARTIE HOTE : 196.22.177.13 • PARTIE RESEAU : 133.156.55.102
4. Une machine est configurée avec l'adresse IP 192.168.1.1 et un masque de réseau 255.255.255.0.
 - Donnez l'adresse du réseau et l'adresse de diffusion sur ce réseau.
 - Même question avec l'adresse IP 172.26.17.100 et le masque de réseau 255.255.240.0.
 - Même question avec l'adresse IP 193.48.57.163 et le masque de réseau 255.255.255.224.

Jeu bilingue

| Français | Anglais |
|---------------------|--------------------------|
| Réseau informatique | Computer network |
| Adresse IP | Internet Protocol adress |
| Adressage | Addressing |

UE I2 : Créer et configurer un réseau informatique de 03 PC 2H

Compétences visées : A la fin de cette leçon, l'élève de T^{le} C/D doit être capable :

- D'utiliser les commandes réseaux ipconfig et ping ;
- De simuler un réseau informatique.

SITUATION PROBLEME :

Dans le cadre des travaux pratiques au laboratoire informatique, il vous est demandé de concevoir un réseau local constitué de 4 postes de travail dans le but de prédire son comportement avant son fonctionnement. Votre conception doit être représentée sur support papier et support numérique.

Consigne :

1. Enumérer les équipements matériels ainsi que les supports d'interconnexion et de transmission que vous représenterez sur le papier ;
2. Proposer un exemple de logiciel qui vous permettra de produire le support numérique correspondant au résultat obtenu au (1) ;
3. Préciser la commande qui permettra de tester les connexions de tous les postes de travail du réseau ;
4. Donner la commande qui affiche les paramètres réseaux d'un ordinateur.

INTRODUCTION

De nos jours, la mise en place d'un réseau informatique nécessite de faire des simulations avant le lancement en production. Cette bonne pratique est indispensable car elle permet d'avoir une idée sur le comportement futur du réseau et d'éviter de nombreuses erreurs. Ainsi, les tests de connexion et la vérification des paramètres réseaux des équipements peuvent être effectués à l'aide des commandes réseaux. Dans la suite notre leçon, nous décrirons quelques commandes réseaux et comment simuler un réseau informatique.

I. LES COMMANDES RESEAUX

Une commande réseau représente une instruction donnée au système d'exploitation pour exécuter une action dans le domaine des réseaux informatiques. Chaque commande répond à une syntaxe et doit être entrée dans l'invite de commande (pour les systèmes Windows) ou dans un Terminal (pour les systèmes UNIX) pour s'exécuter. Il existe plusieurs commandes réseaux parmi lesquels nous avons les commandes Ipconfig et Ping.

I. La Commande Ipconfig

Cette commande affiche ou rafraîchit la configuration réseau d'un ordinateur.

Sa syntaxe est la suivante : **Ipconfig " option "**. Pour l'utiliser il suffit d'ouvrir l'invite de commande, de saisir la commande et de la valider en tapant la touche du clavier **" Entrer "**.

Exemple :

Ipconfig /all (avec l'option /all) : affiche les informations détaillées sur la configuration réseau d'un ordinateur à savoir : le nom de l'hôte, l'adresse IP en cours, le masque de réseau, la passerelle par défaut, les adresses MAC des cartes réseaux, l'état des serveurs DHCP, etc.

Cette commande possède d'autres options qui seront abordées dans le cadre des travaux pratiques.

L'exécution de la commande **Ipconfig** sans option affiche l'adresse IP, le masque réseau et la passerelle par défaut d'une machine.

2. La Commande Ping

La commande Ping permet de tester la connectivité d'une machine à travers un réseau. Pour vérifier la disponibilité d'un ordinateur, la commande Ping envoie des requêtes (4 par défaut) vers l'adresse IP de cet ordinateur grâce au protocole ICMP (Internet Control Message Protocol).

Sa syntaxe est : **Ping " Adresse_machine "** ; où **" Adresse_machine "** représente l'adresse IP ou le nom de la machine ou bien un nom de domaine.

Exemples :

Ping www.google.fr : teste l'accessibilité de la machine locale par rapport au site web Google ;

Ping www.facebook.com : teste l'accessibilité de la machine locale par rapport au site web Facebook ;

Ping /? (suivi du symbole /?) : Affiche toutes les informations sur l'utilisation de la commande Ping.

Le résultat de la commande Ping permet ainsi de connaître :

- L'adresse IP correspondant au nom de la machine distante ;
- La durée de vie des paquets ou TTL (Time To Live) ;
- Le temps de propagation des paquets (en millisecondes) ;
- Le nombre de paquets perdus ;

La commande Ping possède également des options qui seront vues au laboratoire informatique.

II. La Simulation réseau

La simulation réseau représente une technique permettant de créer et de configurer un réseau informatique virtuel à l'aide d'un logiciel (encore appelé simulateur). L'utilisation d'un simulateur réseau ressort plusieurs avantages à savoir :

- La connaissance de l'ensemble des équipements réseaux dans le simulateur,
- La possibilité de visualiser le fonctionnement de tous les équipements à chaque échange d'informations,
- La possibilité d'animer le déplacement d'un message d'une machine à une autre.

Il existe plusieurs logiciels de simulation réseau parmi lesquels nous avons : Cisco Packet Tracer, GNS3, Netsim, NS2...

CONCLUSION

Nous retenons qu'il est important de simuler un réseau informatique avant de le réaliser car cela permet d'avoir une idée sur son comportement et permet aussi d'éviter de nombreuses erreurs. De plus, cette simulation fait appel aux commandes réseaux pour effectuer les tests.

Jeu Bilingue : Traduire en français ou en anglais les mots ou expressions contenus dans le tableau suivant.

| Français | Anglais |
|-------------------|-----------------|
| Réseau | |
| | Network command |
| Simulateur réseau | |

Exercice I : Questions à réponses ouvertes

1. Définir les expressions suivantes : Commande réseau, simulation réseau
2. Donner la fonction des commandes Ping et Ipconfig
3. Comment utilise-t-on la commande Ping? Donnez un exemple d'utilisation de cette commande
4. Comment utilise-t-on la commande Ipconfig? Donnez un exemple d'utilisation de cette commande
5. Donnez en un exemple de simulateur réseau
6. Pourquoi est-il important de simuler un réseau informatique avant sa réalisation?

Exercice 2 (TP) : Utilisation des commandes Ping et Ipconfig à l'invite de commandes Windows

Lors d'une séance de TP au laboratoire informatique, votre professeur vous demande d'utiliser les commandes réseaux dans l'invite de commandes pour afficher les paramètres réseaux des postes et effectuer les tests de connexion au serveur.

1. Accéder à l'invite de commande de votre ordinateur :

(Cliquer sur le « **Menu Démarrer** » puis saisir la commande « **cmd** » dans la zone de recherche et valider) ;

2. Pour avoir plus de détails sur la commande Ping, entrer la commande ping suivi du symbole " /? " et valider : Décrire les différentes options qui s'affichent ;
3. Pour avoir plus de détails sur la commande Ipconfig, entrer la commande Ipconfig suivie du symbole " /? " et valider : Décrire les différentes options qui s'affichent ;
4. Activer votre connexion Internet et entrer les commandes suivantes :
 - a) Ipconfig
 - b) Ipconfig /all

Question : Quelles différences pouvez-vous faire entre les résultats obtenus au « a) » et au « b) » ?

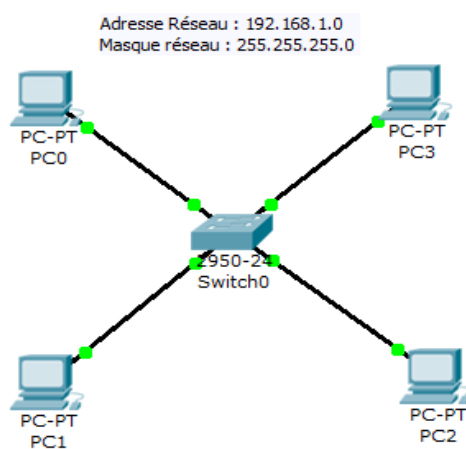
- c) Ping www.yahoo.fr
- d) Ping www.google.fr
- e) Ping <http://grandprof.org>

Questions:

- Décrire l'état de la connexion dans chacun des résultats obtenu par la commande Ping ;
- Les temps de réponses des paquets sont-ils identiques? Pourquoi?

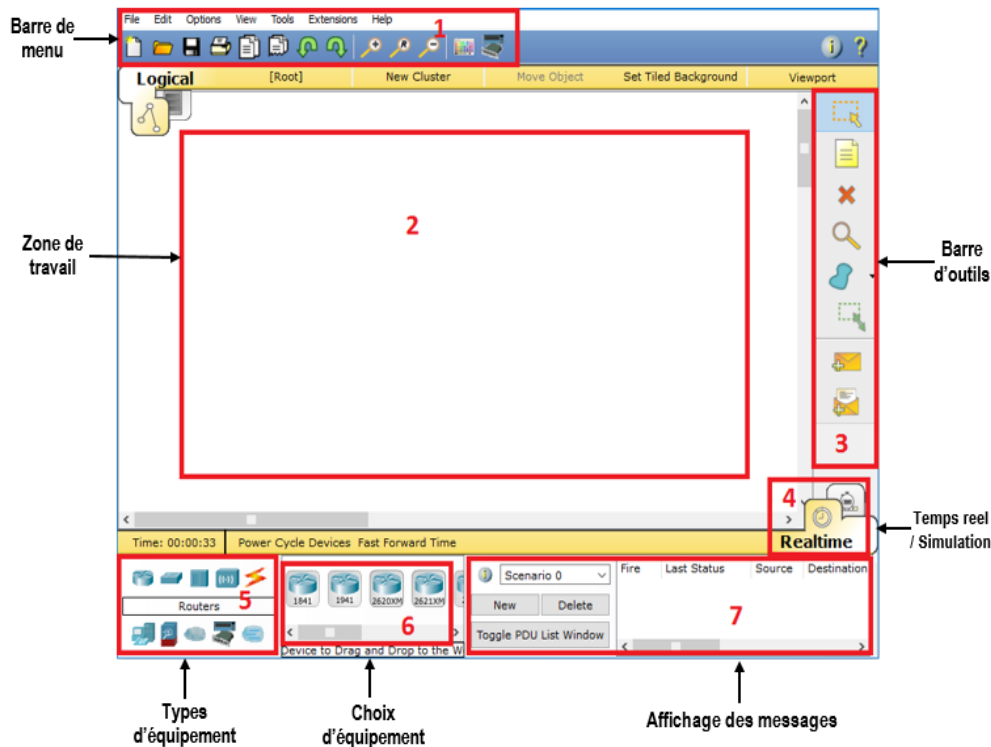
Exercice 3 (TP) : Simulation d'un réseau local : cas du logiciel Cisco Packet Tracer

Dans le but d'évaluer les compétences des élèves en réseau informatique, il vous est demandé de simuler un réseau local dont la topologie est présentée à travers l'image suivante :



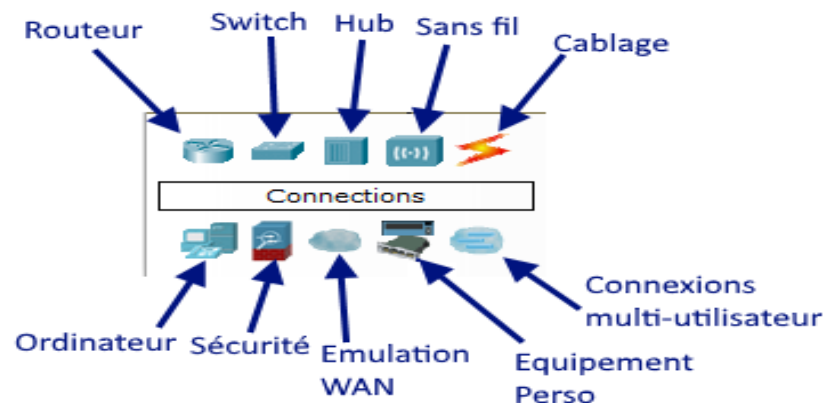
Consigne :

1. Démarrer l'application Cisco Packet Tracer
2. Après le lancement de l'application, vérifier que vous avez une interface ayant la forme suivante :



Identifier et décrire les différentes parties de cette interface

3. Aller dans la zone « **Types d'équipement** » de l'interface, cliquer sur l'icône de l'ordinateur et faites le choix de l'ordinateur dans la zone « **Choix d'équipement** » en effectuant un glisser-déposer dans l'espace de travail (partie 2). L'image suivante vous permettra d'identifier les différents types de matériel.



4. Répéter l'opération effectuée à la question précédente jusqu'à obtenir 4 ordinateurs (PC0, PC1, PC2 et PC3)
5. Ajouter le commutateur (Switch) dans l'espace de travail
6. Pour connecter les postes au switch, vous pouvez procéder de la manière suivante :
 - a. Cliquer sur l'icône des câbles réseaux et sur le câble droit (en noir foncé)
 - b. Cliquer ensuite sur « PC0 » en choisissant le port « **FastEthernet** »
 - c. Cliquer après sur le switch et choisir le port « **FastEthernet 0/I** »

- d. Utiliser le même principe pour connecter « **PCI** », « **PC2** » et « **PC3** » respectivement aux ports « **FastEthernet 0/2** », « **FastEthernet 0/3** » et « **FastEthernet 0/4** » du switch.
7. Par la suite, vous devez attribuer les adresses IP aux ordinateurs à partir de l'adresse réseau **192.168.1.0**. Vous pouvez par exemple utiliser les adresses IP **192.168.1.1**, **192.168.1.2**, **192.168.1.3** et **192.168.1.4** respectivement pour les postes « **PC0** », « **PCI** », « **PC2** » et « **PC3** ». Pour adresser ces ordinateurs :
 - a. Cliquer sur le poste « **PC0** »
 - b. Sur la petite fenêtre qui s'affiche, cliquer sur l'onglet « **Desktop** » ensuite sur « **IP configuration** »
 - c. Saisir l'adresse IP de « **PC0** » ainsi que son masque de réseau (**255.255.255.0**) et fermer la petite fenêtre.
 - d. Utiliser le même procédé pour adresser les postes « **PCI** », « **PC2** » et « **PC3** »
8. Après l'adressage réseau, vous devez consulter en ligne de commande du simulateur, les configurations réseaux des ordinateurs et tester les différentes connexions. Pour ce faire :
 - a. Cliquer sur le poste « **PC0** » ensuite sur l'onglet « **Desktop** » et enfin sur « **Command Prompt** »
 - b. Saisir les commandes suivantes pour avoir les paramètres réseaux de « **PC0** » : `Ipconfig /all`
 - c. Enter les commandes suivantes pour tester la connectivité des postes de travail par rapport à « **PC0** »
 Ping 192.168.1.2, Ping 192.168.1.3 et Ping 192.168.1.4
 - d. Vérifier également la configuration réseau des postes « **PCI** », « **PC2** » et « **PC3** »
9. Pour visualiser les échanges d'informations, vous devez passer en mode simulation en cliquant sur l'icône correspondante (partie 4).
 - a. Aller dans la partie 3 et cliquer sur l'icône des messages simples
 - b. Cliquer ensuite sur « **PC0** » (par exemple) puis sur l'icône d'un ordinateur quelconque puis cliquer sur « **Auto Capture / Play** ».
 - c. Enregistrer votre travail dans le répertoire « **Document** » et fermer l'application.

Module 2 : SYSTEMES D'INFORMATION ET BASES DE DONNEES

UA 4: SYSTEMES D'INFORMATION 6H

UE I4: GENERALITE SUR LES SI IH

COMPETENCES VISEES : A la fin de cette unité d'apprentissage, l'élève sera capable de:

- Définir méthode de conception
- Donner le rôle d'une méthode de conception
- Citer quelques méthodes de conception

SITUATION PROBLEME :

A la rentrée scolaire, on vous a nommé chef de classe. A cet effet il vous est demandé de mettre sur place un moyen pour bien gérer la discipline dans votre salle :

- Quelle démarche allez-vous adopter?
- De quels outils auriez-vous besoin?
- Quelles sont les méthodes adaptées à la réalisation de cette tâche?

INTRODUCTION

De nos jours l'utilisation compétente des moyens, des procédures et des outils informatique permet selon les besoins et les intentions exprimées par une entreprise un accompagnement dans l'automatisation, et la dématérialisation des opérations incluses dans ses activités. Toutes fois, l'intérêt et la réussite d'un S.I reposent sur l'appui d'une méthode de conception.

I- QU'EST-CE QU'UNE MÉTHODE DE CONCEPTION

I- Définitions

En ingénierie, une **méthode d'analyse et de conception** est un procédé qui a pour objectif de formaliser les étapes préliminaires de développement d'un système afin de rendre ce développement plus fidèle aux besoins du client.

2- Les composants d'une méthode de conception

Une méthode de conception possède un certain nombre de composants :

- **Une philosophie générale** : qui constitue un guide indiquant la manière d'aborder un problème dans son environnement.
- **Une démarche** : qui est en quelque sorte le mode d'emploi de la méthode.
- **Des outils** : qui aident à la conception, et aussi à l'analyse et à l'implémentation

- **Un vocabulaire** : pour identifier et décrire les concepts de la méthode.
- **Un formalisme et des normes** : pour spécifier la des divers composant du système.

II- Rôle d'une méthode de conception

Le concept de méthode de conception de système d'information découle de la nécessité d'une concertation entre utilisateurs, décideurs, et concepteurs du système. Une méthode de conception à un double rôle :

- **Elle guide et indique comment aborder des problèmes** : en ce sens qu'elle comprend une démarche de modélisation qui s'appuie sur des formalismes.
- **Elle propose des normes et standard de présentation des résultats du travail du concepteur** : les normes sont ici d'une absolue nécessité car elles garantissent que le travail de conception respecte une démarche véritable.

III- Quelques méthodes de conception d'un S.I

Il existe plusieurs méthodes d'analyse et de conception des systèmes d'information. Les plus connues sont les suivants :

| Méthode | Approche | description |
|--------------------------------------------------------------------------------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------|
| MERISE (<i>Méthode d'Etude et de Réalisation Informatique des Système d'Entreprise</i>) | <i>systemique</i> | <i>Elle est adaptée pour la gestion des projets internes aux organisations</i> |
| SADT (<i>Structured Analysis Design Technic</i>) | <i>Cartésienne</i> | <i>Permet de décrire les tâches d'un projet et leurs interactions, mais aussi le système à</i> |
| UML (<i>Unified Method Language</i>) | <i>Orienté-Objet</i> | <i>Elle est développée dans le but de définir la notation standard des applications construite à basez d'objets</i> |
| OMT (<i>Objet Modeling Technic</i>) | <i>Orienté-Objet</i> | <i>Permet de couvrir l'ensemble des processus d'analyse et de conception en utilisant un même formalisme.</i> |

CONCLUSION

Il existe de nombreuses méthodes d'analyse et de conception des S.I. Chacune d'entre elles a ses particularités, mais toutes sont construites autour d'une philosophie générale, une démarche, des outils, un vocabulaire, des formalismes. De quoi découlent les modélisations systémiques, cartésiennes et orienté-objet.

JEU BILINGUE : Traduire les mots et expressions suivantes en la langue correspondante

| Français | Anglais |
|-----------------------|--------------------|
| Système d'information | Information system |
| Modélisation | Modelization |
| Méthode de conception | Design method |

EXERCICES DE CONSOLIDATION

Vous êtes en stage dans une entreprise de la place, et dès votre arrivée, vous êtes affecté à la cellule informatique de la structure où la mission première est celle d'informatiser la gestion de son personnel. Pour mener à bien cette mission, il faut au préalable faire une étude du système actuel, et ensuite analyser le futur système et le concevoir, et pour cela, il va falloir se baser sur une méthode de conception. Ayant suivi en cour d'année un cours sur les systèmes d'information, on vous demande de répondre aux questions ci-après :

- 1) C'est quoi une méthode de conception de système d'information ?
- 2) Une méthode de conception est basée sur un certain nombre de composants, citer deux composants d'un S.I
- 3) Il existe plusieurs méthodes d'analyse et de conception des S.I
 - a) Citer deux méthodes de votre choix
 - b) Pour chacune des méthodes, donner son approche.

UE 15: ELABORATION D'UN MCD 3H

COMPETENCES VISEES : A la fin de cette unité d'enseignement sera capable de construire un Modèle Conceptuel de Données

SITUATION PROBLEME

Une quincaillerie de la place désire informatiser son système de facturation. Elle fait appel à vous pour réaliser l'analyse. Son service vente établit les factures selon le modèle ci-dessous.

EVA-Commerce
Tel: 677217454
Email: evacom@gmail.com

| FACTURE | | | | |
|---------------------------|--------------|----------|--------------------|------------------|
| Code Client : I0279 | | | N° Facture : 00564 | |
| Nom Client : ESSOMBA Jean | | | Date 13/07/2020 | |
| Ville : Yaoundé | | | | |
| Tel : 678992543 | | | | |
| Référence Produit | Désignation | Quantité | Prix Unitaire | Montant Brut |
| P001 | Ciment 50 Kg | 5 | 4 900 | 24 500 |
| P002 | Pot peinture | 3 | 18 000 | 54 000 |
| P001 | Pointe 80 | 7 | 1 500 | 10 500 |
| Total HT | | | | 89 000 |
| TVA (19,25%) | | | | 17 132,5 |
| Total TTC | | | | 106 132,5 |

CONSIGNES

1. Recenser toutes les données figurant dans ce modèle de facture dans un tableau, définir pour chaque donnée son type (élémentaire ou calculée, et écrire les règles de calcul concernant les données calculées).
2. Après avoir supprimé les doublons et les données calculées du tableau obtenu, identifier les différentes entités ainsi que leurs propriétés.
3. A travers des phrases simples, exprimer les liens existants entre les différentes entités identifiées à la consigne précédente.

INTRODUCTION

Le modèle conceptuel de données (MCD) est une représentation stable, sous forme schématique de l'ensemble des données manipulées par l'entreprise ainsi que des relations entre ces données.

Le MCD a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur les concepts «*Entité - Association*».

La construction du MCD nécessite de recenser les règles de gestion, d'élaborer le dictionnaire de données, de structurer les données en entité et de définir les relations entre les entités.

I. LE RECENSEMENT DES REGLES DE GESTION

Une règle de gestion est un élément de description globale du fonctionnement de l'organisation. Elle peut porter sur les données manipulées par l'organisation ou sur les traitements exécutés au sein de celle-ci.

Exemple : Règles de gestion relatives à la facturation dans un commerce.

RG 1 : Une facture concerne un et un seul client.

RG 2 : Un client peut régler une ou plusieurs factures.

RG 3 : Une facture peut contenir un ou plusieurs produits différents.

II. L'ELABORATION DU DICTIONNAIRE DES DONNEES

Le dictionnaire des données est un tableau qui regroupe l'ensemble des données constituant le système d'information et qui seront utilisées pour élaborer le modèle conceptuel des données. Ces données sont décrites grâce aux règles de gestions.

Le dictionnaire de donnée se construit en plusieurs étapes: le recueil des données, la spécification et l'épuration.

I. Collecte des données

Cette phase consiste à recenser à partir de l'analyse des documents utilisés, des entrevues avec les acteurs (décideurs, utilisateurs...), de la description de l'activité, de la description des objectifs et parfois des questionnaires toutes les données utilisées dans le système.

Les données recueillies sont classées en:

→ **Données élémentaires ou brutes:** données utilisées sans modification préalable. Elles ne sont pas obtenues par calcul à partir d'autres données ;

Exemple : Prix Unitaire, Quantité, Nom du Client...

→ **Données composées :** Si on a une donnée composée de plusieurs donnée alors éclatement en plusieurs données élémentaires ;

Exemple : L'identité du client composé du nom du client et du prénom du client donne lieu à deux données : Nom du client, Prénom du client.

→ **Données calculées ou déduites :** Ce sont des données obtenues par calcul à partir de données élémentaires ;

Exemple : Total HT, Total TTC, Montant Brut...

→ **Données paramètres :** données utilisés par le système d'information sans qu'elles fassent partie des données liées à l'activité de l'entreprise. Elle ne prend qu'une unique valeur.

Exemple : le taux de la TVA, le nom et l'adresse de l'entreprise...

2. Détermination du type de données

Dans un dictionnaire de données les types standards sont :

- Alphabétique (A) : lorsque la donnée est uniquement composée de caractères alphabétiques ;
- Numérique (N) : lorsque la donnée est composée uniquement de nombres ;
- Alphanumérique(AN) : lorsque la donnée peut être composée à la fois de caractères alphabétiques et numériques ;
- Date(D) : lorsque la donnée est une date (au format AAAA-MM-JJ) ;
- Booléen (B) : Vrai ou Faux ;

L'ensemble des données recueillies constitue le dictionnaire des données brut qui se présente comme suit :

| Nom symbolique | Description | Nature | Type | Taille | Règle de calcul, Contrainte |
|----------------|-------------------------------|--------|------|--------|-------------------------------------|
| Code_Client | Code du Client | E | N | 5 | |
| Nom_Client | Nom du Client | C | A | 30 | Nom + Prénom |
| ville_Client | Adresse du Client | E | A | 20 | |
| Tel_Client | N° Téléphone du Client | E | N | 9 | |
| Num_Fact | N° Facture | E | N | 5 | |
| Date_Fact | Date de la Facture | E | D | 10 | AAAA-JJ-MM |
| Ref_Pduit | Référence du Produit | E | AN | 15 | |
| Lib_Pduit | Libellé Produit | E | AN | 20 | |
| PU | Prix Unitaire | E | N | 6 | Obligatoire, > 0 |
| Quantite | Quantité Produit | E | N | 3 | Obligatoire, > 0 |
| M_Brut | Montant Brut | C | N | 6 | $M_{Brut} \times Qte_P$ |
| Total_HT | Total Hors Taxe | C | N | 6 | <i>Somme des montants hors taxe</i> |
| Tx_TVA | Taux TVA | P | N | 5 | |
| TVA | Total TVA | C | N | 6 | $Total_{HT} \times Tx_{TVA}$ |
| Total_TTC | Total toutes taxes confondues | C | N | 6 | $Total_{HT} + TVA$ |

3. Epuration des données

Pour la phase de modélisation des données, il convient conserver uniquement les données élémentaires en supprimant :

- Les données calculées ;
- Les données paramètre ;
- Des synonymes : données ayant le même sens. Pour supprimer les synonymes, il suffit de supprimer une des données ;
- Des polysèmes : donnée ayant plusieurs sens. Pour supprimer les polysèmes, il suffit de renommer une des données ;

Le dictionnaire de données épurées est organisé sous forme d'un tableau de cinq colonnes: Nom symbolique, Description, Domaine ou type, Taille et Commentaires.

Exemple: Le dictionnaire des données relatif à la facturation dans un commerce.

| Nom symbolique | Description | Type | Taille | Commentaires |
|----------------|------------------------|------|--------|------------------|
| Code_Client | Code du Client | N | 5 | |
| Nom_Client | Nom du Client | A | 30 | |
| Prenom_Client | Prénom du Client | A | 20 | |
| ville_Client | Adresse du Client | A | 20 | |
| Tel_Client | N° Téléphone du Client | N | 9 | |
| Num_Fact | N° Facture | N | 5 | |
| Date_Fact | Date de la Facture | D | 10 | AAAA-JJ-MM |
| Ref_Pduit | Référence du Produit | AN | 15 | |
| Libelle | Libellé Produit | AN | 20 | Obligatoire, > 0 |
| PU | Prix Unitaire | N | 6 | Obligatoire, > 0 |
| Quantite | Quantité facturée | N | 3 | |

III. L'ELABORATION DU MODELE CONCEPTUEL DE DONNEES

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 ».

Les éléments clés de ce formalisme sont : **entités, propriété, identifiant, occurrence, association, cardinalité et contrainte d'intégrité.**

I. L'entité

Une entité est la représentation dans le système d'information d'un objet matériel ou immatériel ayant une existence propre défini au sein d'un système d'information.

Elle est caractérisée par un certain nombre des propriétés qui lui sont spécifiques.

Exemple : Facture, Client, Produit...

2. La propriété (ou attribut ou rubrique)

La propriété est une donnée élémentaire qui caractérise une entité ou une relation entre objets.

Formalisme: Le nom de la propriété est inscrit à l'intérieur de l'objet.

Exemple, Réf_Pduit, Libelle, PU, Nom_Client ...

3. L'identifiant

L'identifiant est une propriété particulière qui permet de distinguer de façon certaine et unique les différentes occurrences d'une l'entité.

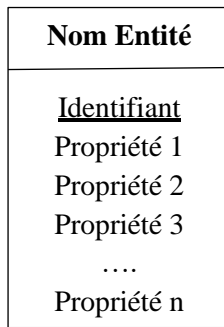
La connaissance de l'identifiant donne la connaissance des autres propriétés de l'entité. Il est toujours souligné.

Exemple : Code_Client, Num_Fact, Ref_Pduit.

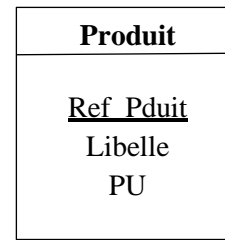
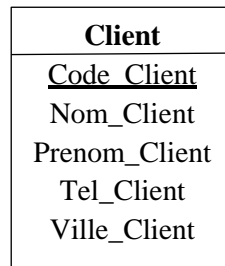
4. Occurrence

L'occurrence d'une entité correspond aux valeurs prise par les propriétés.

Représentation graphique d'une entité



Exemple:

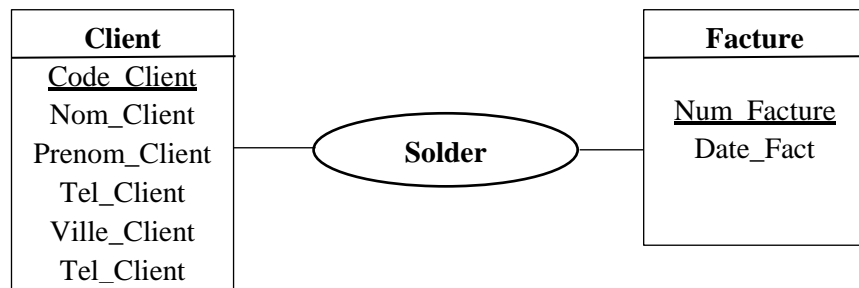


5. Les associations (Relations)

Une association permet de relier, une ou plusieurs entités. Elle est généralement caractérisée par un verbe ou un substantif. Elle peut être porteuse de propriétés. Une association est porteuse de données si elle possède une ou plusieurs propriétés.

Une association est représentée sous forme d'une ellipse à l'intérieur de laquelle est inscrit son libellé. Une patte est dessinée entre la relation et chacune des entités qu'elle relie.

Exemples : Existence d'une association **Solder** entre l'entité **Facture** et **Client** dans le cas système de facturation sera matérialisée comme suit :



6. Les cardinalités

Les cardinalités précisent les nombres minimal et maximal d'occurrences d'une entité en relation avec une autre.

Elles permettent de connaître avec exactitude le degré de participation de chaque entité à une relation.

La cardinalité minimale représente le nombre de fois « au minimum » où une occurrence de l'association participe aux occurrences de l'entité.

La cardinalité maximale représente le nombre de fois « au maximum » où une occurrence de l'association participe aux occurrences de l'entité.

Les cardinalités sont mentionnées par couple du côté de l'entité à considérer. La cardinalité minimale est représentée en premier, la maximale en second et se formalise comme suit : (**Min**, **Max**).

Les couples possibles sont : (0,1), (1,1); (0,n), (1,n).

7. Les contraintes d'intégrité fonctionnelles (CIF)

Une CIF ou Contrainte d'Intégrité Fonctionnelle également nommée « association hiérarchique » est un type d'association entre deux entités. Elle se caractérise par un **I** en cardinalité supérieure (0,1 ou 1,1) sur une des pattes de la relation.

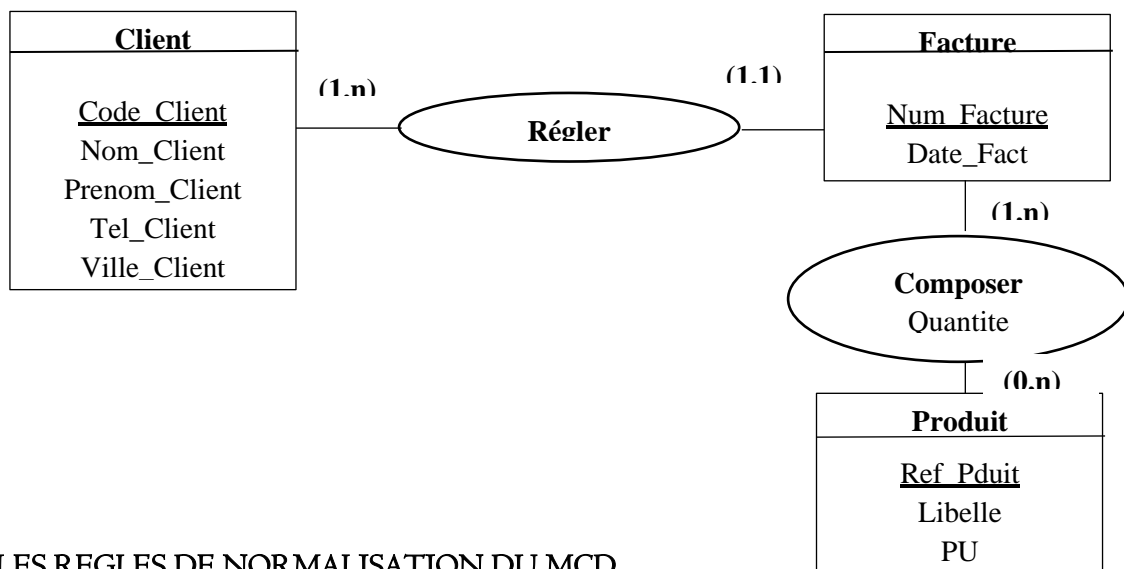
Elle traduit la présence d'une dépendance fonctionnelle entre les identifiants des entités participant à l'association

IV. Représentation du modèle conceptuel

Le MCD est élaboré à l'aide du dictionnaire des données. La démarche de construction est la suivante:

- Repérer les identifiants existants pour dégager les entités ;
- Rattacher à ces entités leurs propriétés grâce aux dépendances fonctionnelles,
- Placer les associations entre entités et leur rattacher leurs éventuelles propriétés ;
- Étudier les cardinalités de chaque couple Entité – Relation ;
- Simplifier le modèle à l'aide des contraintes d'intégrité fonctionnelle ;
- Procéder à la vérification à l'aide des règles.

Exemple : Le MCD relatif à la facturation dans un commerce.



V. LES REGLES DE NORMALISATION DU MCD

Lorsque le MCD est terminé, on doit procéder à quelques vérifications avant la validation. Elle permet de détecter certaines incohérences dans la construction des modèles. Dans le cadre de notre cours nous appliquerons les règles de normalisation du MCD suivantes:

1. Normalisation des entités et des relations

- 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 être unique ;
- Le nom d'une entité/association est écrit en lettres majuscules.

2. Normalisation des identifiants

- Chaque entité doit posséder un identifiant ;
- Un identifiant ne doit pas être composé de plusieurs propriétés pour les entités ;

3. Normaliser les attributs

- Toutes les propriétés d'une entité ou d'une association doivent être 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. Éliminer les associations fantômes

Il faut éliminer les associations fantômes c'est-à-dire les associations du type (I,I)-(I,I). Une association binaire ne devrait porter des cardinalités I,I aux deux extrémités.

Ce jeu de cardinalité traduit généralement une erreur de conception.

5. 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 MCD ou modèle Entité/Association est un modèle chargé de représenter sous forme graphique les informations manipulées par le système. Ce formalisme comporte quatre concepts de base. Deux concepts sont structuraux, l'entité et la relation; le troisième concept est descriptif, c'est la propriété; le quatrième qualifie la liaison entre entité et relation, c'est la cardinalité. La démarche de construction du MCD se résume en trois phases majeures : le recensement de toutes les données du système d'information, la recherche des liens qui les unissent et représentation graphique des données du domaine d'étude.

Le MCD obtenu une fois validé par la vérification d'un certain nombre de règles, permet de passer au modèle logique de données (MLD).

JEU BILINGUE

| Français | Anglais | Français | Anglais | Français | Anglais |
|------------|------------|-------------|--------------|-------------|------------|
| Entité | Entity | Association | Relationship | Identifiant | Identifier |
| Occurrence | Occurrence | Cardinalité | Cardinality | Attribut | Attribute |

EXERCICES D'APPLICATION

Exercice I : Répondre par vrai ou faux

1. Le modèle entité-association est un schéma théorique qui vise à rendre compte des relations existant entre divers éléments d'un système.
2. La cardinalité précise le nombre de fois minimum et maximum qu'une instance de relation participe à une association.
3. Une entité possède toujours au moins un attribut.
4. Une association est un lien sémantique ayant deux brins et deux couples de cardinalités.
5. Une association s'effectue toujours entre deux entités.

6. Une association ne peut avoir de propriétés.
7. La cardinalité minimale est toujours 1 ou n.
8. La cardinalité maximale est toujours 0 ou 1.

Exercice 2

On s'intéresse à la gestion de consommation des boissons au niveau d'une entreprise. Un consommateur possède un nom et une adresse et il peut consommer plusieurs marques de boissons. Une boisson appartient à une marque et une seule qui possède un code unique et un libellé. Les boissons possèdent une référence (un code), un libellé et un prix unitaire et sont stockées dans des magasins identifiés par des références et possèdent un nom et une adresse. Dans un magasin, on peut trouver plusieurs marques de boissons. Les consommateurs fréquentent divers magasins.

Questions

1. Établir le dictionnaire de données
2. Réaliser le modèle conceptuel de données.

Exercice 3

Pour la gestion de l'organisation d'un examen certificatif, on considère les règles de gestion suivantes :

RG 1 : Les élèves doivent obligatoirement remplir une fiche d'inscription. Cette fiche comprend le nom, la date de naissance, l'établissement de l'élève et le nom de l'examen. Un établissement est défini par son code, son nom, son adresse et la ville.

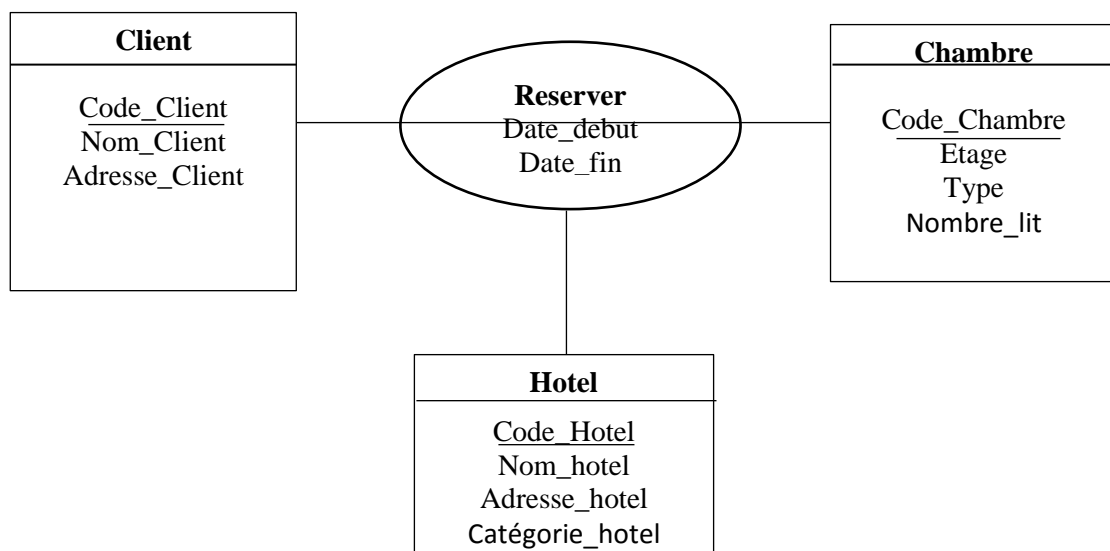
RG 2 : Chaque examen, comprend une série d'épreuves qui lui est propre, chacune dotée d'un coefficient d'un libelle et d'un code.

RG 3 : A chaque épreuve, les candidats obtiennent une note sur 20. A l'issu de la correction des copies, un nombre total de points est calculé à partir des notes obtenues et des coefficients. Si le total des points est d'au moins 210, le candidat est déclaré admis ; sinon, il est ajourné.

Question

Établir le modèle conceptuel de données.

Exercice 4 : Soit le schéma entité-association ci-dessous d'un système de réservation d'un groupe hôtelier disposant d'une centrale de réservation nationale.



Les règles de gestion sont les suivantes :

RG1 : Un client fait une réservation déterminée pour un hôtel, une chambre et dispose d'une date de début et une date de fin d'occupation.

RG 2 : Un client peut réserver plusieurs chambres à la même date mais il doit réserver au moins une chambre.

RG 3 : Un hôtel contient plusieurs chambres de différents types (single, double, ...etc.)

RG 4 : Chaque hôtel correspond à une catégorie particulière (*, **, ***, ...etc.).

Question

Déterminer les cardinalités minimales et maximales de chaque entité participant à cette association.

COMPETENCES VISEES

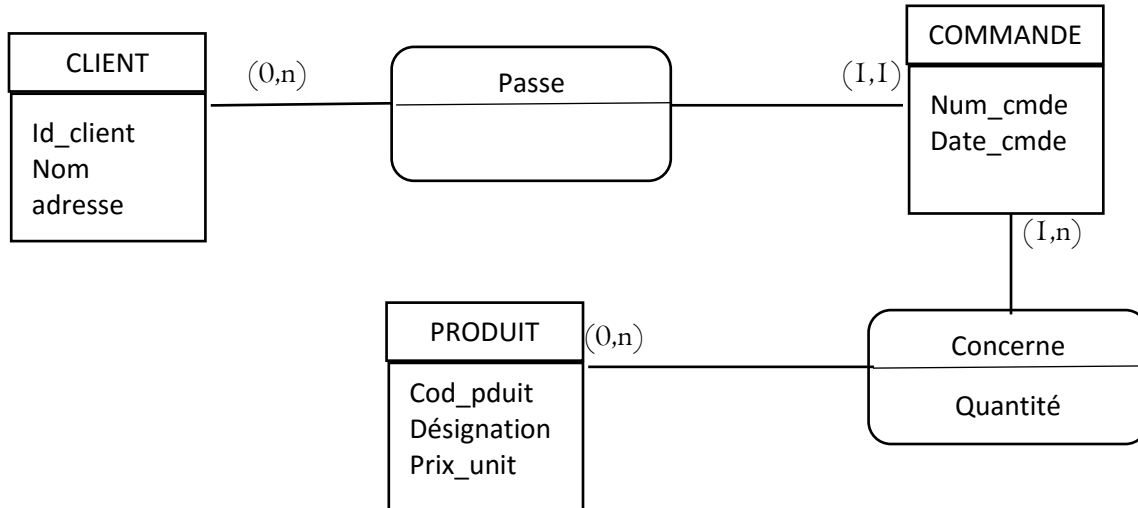
A la fin de cette leçon, l'apprenant doit être capable de :

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

SITUATION PROBLEME

Enoncé

Votre camarade de classe aimerait automatiser la gestion des commandes dans un supermarché et pour ce fait, il a produit le MCD suivant :



Très curieux, il apprend qu'il doit traduire son MCD en MLD afin de produire la base de données associée à la gestion de ses commandes.

Consignes

A partir de vos lectures sur la conception d'un SI, aider votre camarade en donnant progressivement, en fonction des règles de passage du MCD au MLD que vous avez identifié dans vos lectures, la liste des tables du MCD, en précisant l'identifiant et les attributs de chaque table ;

CONTENU DE LA LEÇON

Le passage du MCD (modèle conceptuel de données) au modèle relationnel ou MLD (modèle logique de données) ne se fait pas au hasard. Il existe un certain nombre de règles qui vous permettent de réaliser cette opération. Ces règles sont regroupées ici en quatre traitements :

1. Le traitement des entités

- ✓ Chaque entité du MCD est transformé en table dans le MLD
- ✓ Les propriétés de l'entité deviennent les attributs de la table
- ✓ L'identifiant de l'entité devient clé primaire de la table

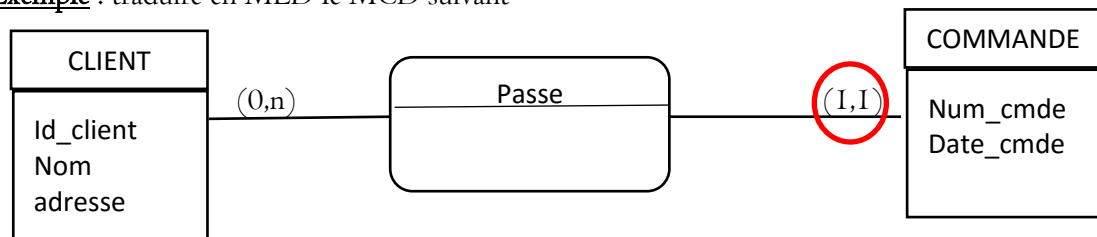
Exemple : donnons la liste des tables (étape 1) du MCD fournit à la situation problème

- CLIENT (Id_client, Nom, adresse)
- COMMANDE (Num_cmde, Date_cmde)
- PRODUIT (Cod_pduit, Désignation, Prix_unit)

2. Le traitement des associations binaires (I,n)

Cette association correspond à une paire de cardinalité (x,n) (y,I) , avec x et y valant 0 ou 1. Dans ce cas, la table issue de l'entité coté cardinalité (y,I) reçoit comme clé étrangère l'identifiant de l'entité liée : il y a migration de la clé primaire de l'entité père vers l'entité fils.

Exemple : traduire en MLD le MCD suivant

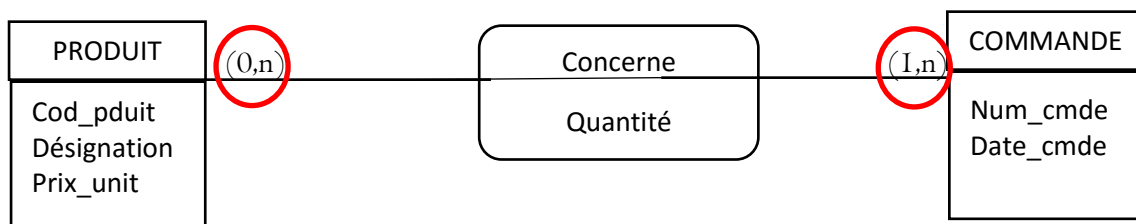


- ✓ CLIENT (Id_client, Nom, adresse)
 - ✓ COMMANDE (Num_cmde, Date_cmde, # Id_client)
- ← Clé étrangère

3. Le traitement des associations binaires (n, n)

Cette association correspond à une paire de cardinalité (x,n) (y,n) , avec x et y valant 0 ou 1. Dans ce cas, l'association **devient une relation**, et sa clé primaire est formée par la concaténation (juxtaposition) l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles de l'association deviennent des attributs de la nouvelle table.

Exemple : traduire en MLD le MCD suivant



- ✓ PRODUIT (Cod_pduit, Désignation, Prix_unit)
- ✓ COMMANDE (Num_cmde, Date_cmde)
- ✓ Concerne (Cod_pduit, Num_cmde, Quantité)

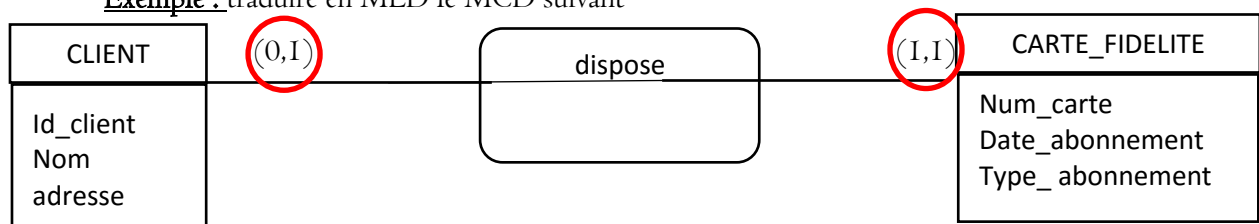
4. Le traitement des associations binaires (I,I)

Cette association correspond à une paire de cardinalité $(0,I)$ (I,I) , ou $(0,I)$ $(0,I)$, la cardinalité (I,I) (I,I) étant interdite lors de la conception du MCD ;

4.1 Relation binaire $(0,I)$ - (I,I)

On duplique la clé de la table basée sur l'entité à cardinalité $(0,I)$ dans la table basée sur l'entité à cardinalité (I,I) .

Exemple : traduire en MLD le MCD suivant

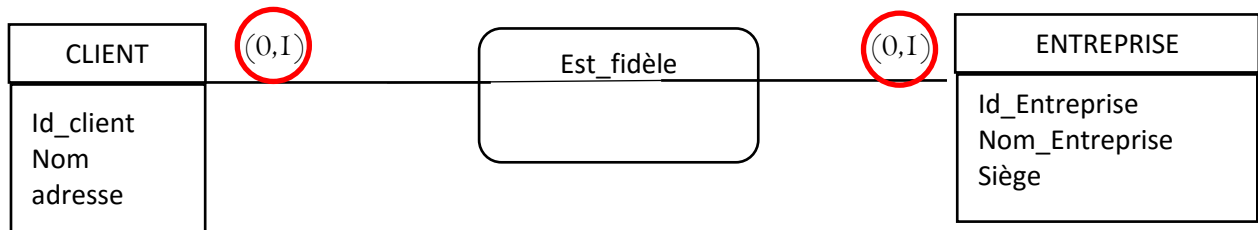


- ✓ CLIENT (Id_client, Nom, adresse)
- ✓ CARTE_FIDELITE (Num_carte, Date_abonnement, Type_abonnement, #Id_client)

4.2 Relation binaire $(0,I)$ - $(0,I)$

Dans ce cas, l'association **devient une relation**, et sa clé primaire est formée par la concaténation (juxtaposition) l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles de l'association deviennent des attributs de la nouvelle table.

Exemple : traduire en MLD le MCD suivant



- ✓ CLIENT (Id_client, Nom, adresse)
- ✓ ENTREPRISE (Id_Entreprise, Nom_Entreprise, Siège)

JEU BILINGUE

- ✓ Association : association ;
- ✓ Cardinalité : cardinality ;
- ✓ Clé primaire : primary key;
- ✓ Clé étrangère : foreign key

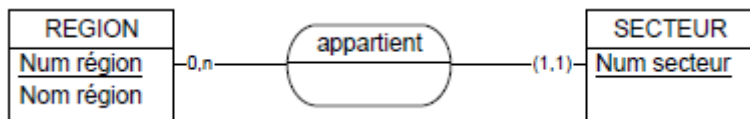
EXERCICES

Après avoir rappelé les règles de passage du MCD au MLD, donner le MLD correspondant à chacun des MCD suivants :

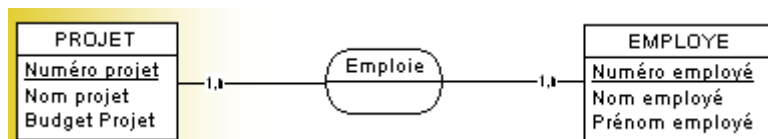
1)



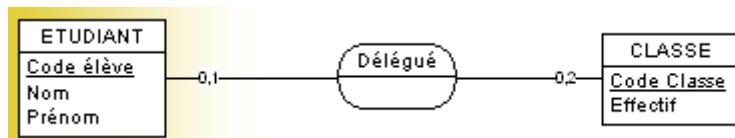
2)



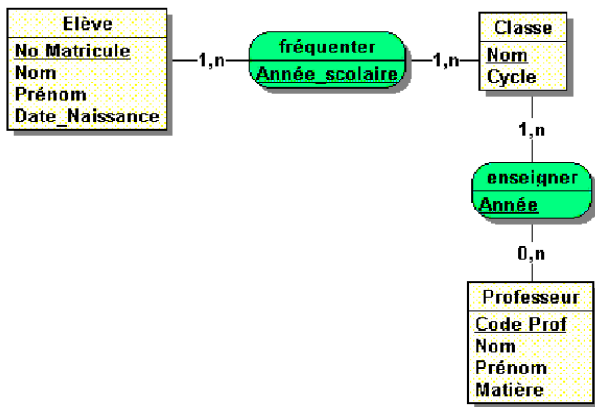
3)



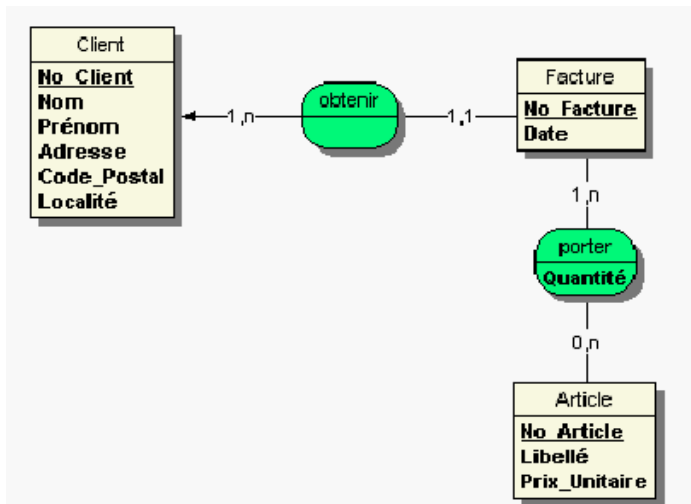
4)



5)



6)



COMPÉTENCES VISÉES :

- Définir SGBD et donner leurs rôles
- Citer quelques exemples de SGBD ;
- Lister les opérations de définition des données d'une base de données (création, suppression, modification) ;
- Lister les opérations de manipulation des données d'une base de données (insertion, modification, sélection, suppression) ;

SITUATION PROBLEME

Le censeur dispose d'un tableau sur lequel est organisé les emplois de temps de tous les enseignants de l'établissement et trouve pénible de consulter ce tableau lorsqu'il faut avoir par exemple pour un professeur donné la liste des classes et des heures d'enseignement au courant de la semaine. Ayant entendu parler des SGBD, il se dit que cela pourrait être la solution et fait appel à vous.

- Qu'est-ce qu'un SGBD et à quoi cela peut-il servir ?
- Donner en quelques exemples que le censeur pourrait utiliser
- Lister quelques fonctions d'un SGBD que le censeur devra certainement utiliser

CONTENUE DE LA LEÇON

INTRODUCTION

Dans un fichier, les informations sont stockées de manière séquentielle (les unes à la suite des autres) ; ceci rend l'accès à une donnée particulièrement difficile lorsque la taille du fichier est considérable ; une alternative serait d'utiliser les bases de données. **Une base de données** désigne un ensemble des données stockées de manière structurée et accessible directement par plusieurs utilisateurs. Pour créer et gérer les bases de données, on a recours à une catégorie de logiciel appelé **SGBD** (Système de Gestion des Bases de Données).

I. Généralités sur les SGBD

Les SGBD permettent de définir la structure de la base de données (table, champs, ...), de manipuler les enregistrements et de contrôler les accès à la base. D'une manière générale, un SGBD permet de créer et supprimer une base de données, d'ajouter, de modifier et de rechercher une donnée qui se trouve éventuellement dans une base de données spécifique.

I.I Définition

Un SGBD est donc un ensemble de programmes qui permettent l'administration ainsi que l'accès en lecture et en écriture des bases de données. On parle également de serveur de données.

I.2 Rôle et intérêt des SGBD

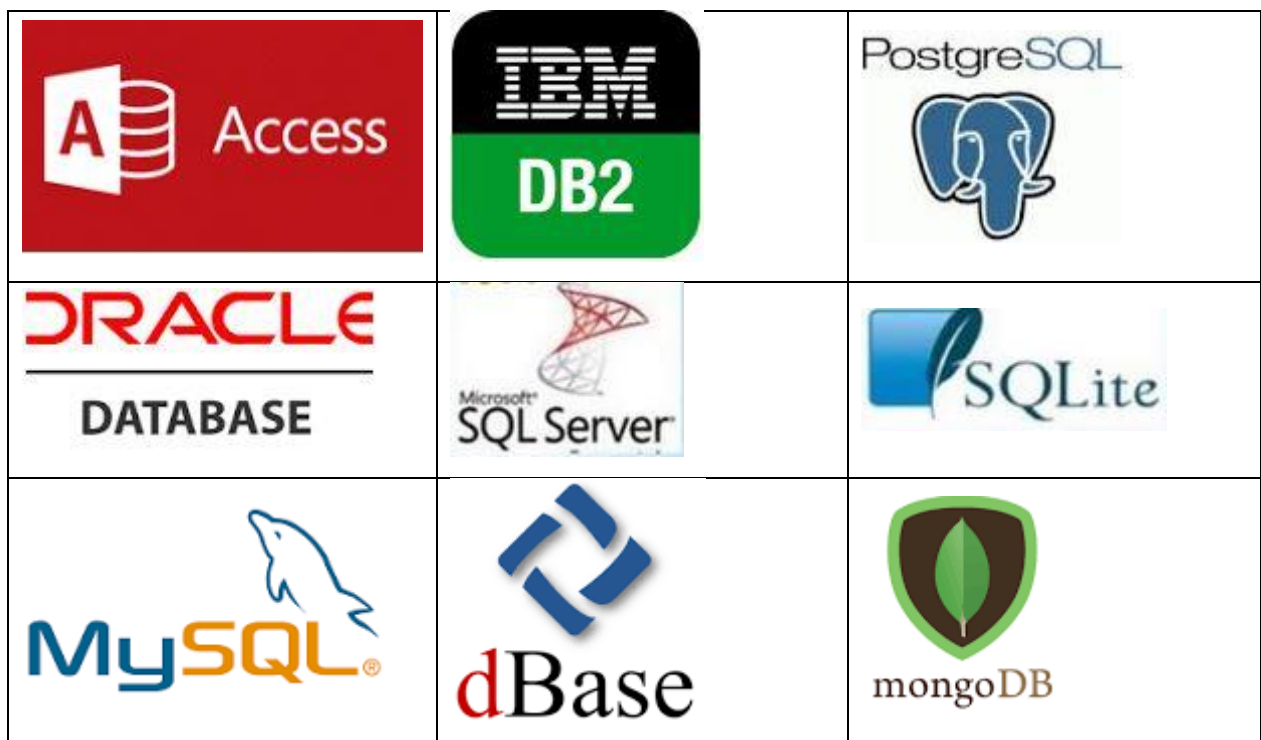
Les SGBD offrent aux utilisateurs les outils nécessaires pour mettre à jour leurs données ; La gestion des données a toujours causé d'énormes problèmes à l'homme et les SGBD ont été créés dans l'objectif de les résoudre. Voici les principaux rôles des SGBD :

- permettre l'accès aux données indépendamment de la façon dont elles ont été sauvegardées
- permettre l'accès rapide aux données
- offrir une interface d'accès aux données conviviale et commune à tous
- assurer la non redondance des données
- assurer la cohérence des données (en évitant par exemple que la suppression d'une donnée affecte toute la base)
- contrôle l'accès aux données
- sauvegarde et assure la récupération des données en cas de pannes
- Etc.

I.3 Exemples

Il existe plusieurs SGBD parmi lesquels certains sont propriétaires et payants tandis que d'autres sont gratuits. Le choix d'un SGBD dépend de l'administrateur et souvent de la qualité ou la quantité de données. Parmi les plus utilisés on peut par exemples citer :

- MySQL
- Oracle
- Ms Access
- DB2
- SQL Server
- PostgreSQL
- MongoDB
- Dbase
- ...



Quelques SGBD représentés par leur logo

II. Quelques opérations dans une base de données (aucune requête)

II.1 La définition des données

Elle permet la définition et la mise à jour des structures de la base (tables, champs, etc).

Elle est réalisée par un ensemble de commandes appelé Langage de Définition des Données (LDD); les opérations ici peuvent être :

- ✓ La création de la base de données ou des tables
- ✓ L'ajout, suppression ou modification d'une table ou d'un champ
- ✓ La suppression de la base de données

II.2 La manipulation des données

Elle permet de manipuler les enregistrements de la base et de les mettre à jour et est réalisée par un ensemble de commandes appelé Langage de Manipulation des Données (LMD). Une partie du LMD est le Langage d'Interrogation des Données (LID) qui permet de rechercher des informations en interrogeant la base ; ici on retrouve les commandes permettant

- ✓ L'insertion de nouvelles données et enregistrements
- ✓ La modification ou mise à jour des données
- ✓ La sélection ou affichage des données
- ✓ La suppression d'enregistrements

II.3 Le contrôle de l'accès aux données

Il permet de définir des droits d'accès aux différents utilisateurs de la base et est réalisé par un ensemble de commandes appelé Langage de Contrôle des Données (LCD).

CONCLUSION

Les SGBD sont des logiciels complexes permettant de gérer efficacement les données d'une structure et il en existe plusieurs. Ils permettent notamment la définition des données, la manipulation des données et le contrôle d'accès aux données. Ici l'accès aux données est direct alors que dans un fichier l'accès est séquentiel.

JEU BILINGUE : Traduire les mots et expressions suivants dans la langue correspondante

| FRANÇAIS | ENGLISH |
|-----------------|-----------------|
| Base de données | Database |
| Table | Table |
| Champ | Field ou column |
| Requête | Query |

| | |
|----------------|-------|
| SGBD | DBMS |
| Enregistrement | Tuple |

EXERCICES

1. Que signifie le sigle SGBD
2. Définir et donner 3 rôles d'un SGBD
3. Après avoir citer quatre SGBD, énumérer deux opérations réalisables dans une base de données au moyen des SGBD respectivement :
 - a) Dans la manipulation des données
 - b) Dans la définition des données
4. Distinguer si dessus (parmi les SGBD énuméré dans le cours), les SGBD payants des SGBD gratuits
5. Donner deux différences entre un SGBD et un fichier

UE I8 : Requêtes SQL de définition et de manipulation des données 3H

Compétences :

A la fin de cette leçon l'élève doit être capable de :

- Définir SQL et donner son rôle
- Créer et supprimer une base de données
- Créer et supprimer une table
- Modifier une table

Situation problème

Une base données contient des tables, selon vous quelles sont les différentes opérations qu'on peut effectuer dans une table ?

I. DÉFINITION ET RÔLE

SQL (Structured Query Language) est un langage de définition, de manipulation, d'interrogation, et de contrôle pour les bases de données relationnelles conçu en 1970 par la société IBM.

Il permet notamment de définir les structures des données, de spécifier les utilisateurs et leurs droits ou d'interroger les bases de données existantes.

SQL est un langage de définition de données (LDD), c'est-à-dire qu'il permet de créer des tables dans une base de données relationnelle, ainsi que d'en modifier ou en supprimer.

I. CRÉATION ET SUPPRESSION D'UNE BASE DE DONNÉES

a) Création d'une base de données

CREATE DATABASE Nom ;

b) Suppression d'une base de données

DROP DATABASE Nom ;

III. CRÉATION ET SUPPRESSION D'UNE TABLE

a) Création de table

La création d'une table se fait en définissant le **nom de la table** ainsi la liste de ses **attributs** et des **types de données**. Les principaux types de données sont présentés dans le tableau ci-après

| Type de donnée | Syntaxe | Description |
|---------------------|---------------|----------------------------------------------------------------|
| Type alphanumérique | CHAR(n) | Chaîne de caractères de longueur fixe n ($n < 16383$) |
| Type alphanumérique | VARCHAR(n) | Chaîne de caractères de n caractères maximum ($n < 16383$) |
| Type numérique | NUMBER(n,[d]) | Nombre de n chiffres [optionnellement d après la virgule] |
| Type numérique | SMALLINT | Entier signé de 16 bits (-32768 à 32757) |
| Type numérique | INTEGER | Entier signé de 32 bits (-2E31 à 2E31-1) |
| Type numérique | FLOAT | Nombre à virgule flottante |

| | | |
|--------------|-----------|---------------------------------|
| Type horaire | DATE | Date sous la forme 16/07/99 |
| Type horaire | TIME | Heure sous la forme 12:54:24.85 |
| Type horaire | TIMESTAMP | Date et Heure |

```
CREATE TABLE Nom_Table (
ChampI TypeI [contrainte],
ChampI Type2 [contrainte]
...
ChampI TypeN [contrainte]
CONSTRAINT nomContrainte typeContrainte);
```

Exemple1

```
CREATE TABLE Matiere (
CodeMat VARCHAR(3),
NomMat VARCHAR(20) NOT NULL,
CONSTRAINT PRIMARY KEY(CodeMat));
```

| CodeMat | NomMat |
|---------|--------|
| | |

Création d'une table nommée matière contenant deux champs et ayant pour clé primaire **CodeMat**

Exemple2

```
CREATE TABLE Note (
Matricule VARCHAR(5),
CodeMat VARCHAR(3),
Note FLOAT,
CONSTRAINT
PRIMARY KEY (Matricule,CodeMat),
FOREIGN KEY(Matricule) REFERENCES Eleve(Matricule),
FOREIGN KEY(CodeMat) REFERENCES Matiere(Codemat));
```

Céation d'une table nmmée Note contenant trois champs ayant pour clé primaire **Matricule et codeMat**, le champ codeMat etant également clé étrangère.

b) Suppression d'une table

DROP TABLE Nom_table

IV. MODIFICATION DES TABLES

| Type de modification | Syntaxe | Commentaire |
|----------------------|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Ajouter un champ | ALTER TABLE Nom_Table ADD Champ Type ; | Permet d'ajouter un champ à une table |
| Supprimer un champ | ALTER TABLE Nom_Table DROP COLUMN Champ; | Permet de supprimer un champ dans une table |
| Renommer un champ | ALTER TABLE Nom_Table CHANGE Ancien_champ Nouveau_Champ Nouveau_Type; | Permet de changer le nom d'un champ. Elle combine la suppression et l'ajout d'un champ |
| Modifier un champ | ALTER TABLE Nom_Table MODIFY Champ Nouveau_Type ; | Permet de modifier le type d'un champ |

V. EXERCISES D'APPLICATION

- I. Vous êtes administrateur de bases de données dans une société de vente de produits en ligne et la base de données créée par vos soins contient la table **clients**(**idclient**, **nom**, **prenom**, **sexe**) ci-dessous.
- a) Donnez la requête SQL ayant pour but de créer cette table
 - b) Quelle est la requête à utiliser pour ajouter le champ **adresse** dans cette table.
 - c) Quelle est la requête à utiliser pour ajouter le champ **prenom** dans cette table.
2. Soit la table suivante ELEVE(Matricule, Nom, prenom, age, codeClasse) où codeClasse est la clé étrangère.
- a) Ecrire la requête SQL permettant de créer cette table.
 - b) On souhaite renommer le champ Matricule en codeEleve écrire la requête correspondante.
 - c) Quelle est la requête permettant de supprimer le champ age.

COMPÉTENCES VISÉES

A la fin de cette leçon, l'apprenant doit être capable de :

- ✓ Écrire, exécuter les requêtes SQL de manipulation de données sur une table
- ✓ Utiliser les opérateurs logiques (AND, OR), de comparaison (=, <, >, <=, >=) ;

SITUATION PROBLÈME

Après avoir suivi avec beaucoup d'attention le cours sur les requêtes SQL de définition des données, votre camarade de classe qui souhaite automatiser la gestion de son supermarché, a créé une base de données contenant plusieurs tables parmi lesquelles : *Client(Id_Clt, nom, adresse)*, *produit(Cod_Pduit, désignation, prix_unit)*.

Votre camarade souhaite ajouter des clients et des produits dans sa base de données, modifier les informations sur un client existant dans la base de données, supprimer certains clients et afficher le résultat de ces requêtes. A partir de vos lectures sur les requêtes SQL, aidez votre camarade en répondant aux questions suivantes :

1) Sachant que la syntaxe SQL d'insertion de données est `INSERT INTO nomTable VALUES (valeur1, valeur2, ...) ;` insérer le client SIMO résident Yaoundé ayant pour identifiant Clt001

2) Sachant que la syntaxe SQL d'insertion de données est `UPDATE nomtable SET col1 = "valeur1", col2 = valeur _2, ... WHERE condition ;` modifier les informations du client SIMO sachant qu'elle réside désormais à Douala

3) Afficher la liste des clients qui résident à Douala, en utilisant la syntaxe SQL suivante : `SELECT liste attributs FROM nomTable WHERE condition ;`

Afficher la liste des produits dont le **prix_unit** est supérieur à 10000

4) Ecrire la requête SQL permettant de supprimer tous les produits dont **prix_unit** est compris entre 5000 et 10000, en utilisant la syntaxe suivante : `DELETE FROM nomTable WHERE condition.`

CONTENU DE LA LEÇON

I. REQUETES DE MANIPULATION DES DONNEES

I. Requête SQL d'insertion de données

L'insertion de données renvoie à l'ajout d'un enregistrement dans une table. La syntaxe SQL est la suivante :

```
INSERT INTO nom_table [(nomCol1, nomCol 2, ...)]VALUES (val1, val2, ...);
```

NB : La liste des noms de colonne est optionnelle. Lorsqu'elle est omise, la liste des valeurs est donnée dans l'ordre de création de la table.

Exemple : insertion du client SIMO résident Yaoundé ayant pour identifiant Clt001 : `INSERT INTO Client VALUES ("Clt001", "SIMO", "Yaoundé") ;`

ou INSERT INTO Client(nom, adresse, Id_Clt) VALUES ("Clt001", "SIMO", "Yaoundé") ;

2. Requête SQL de modification de données

Pour modifier les enregistrements d'une table, on se sert de la commande UPDATE. Sa syntaxe est la suivante :

```
UPDATE nomtable
SET col1 = "valeur1", col2 = "valeur_2", ...
WHERE condition ;
```

Exemple : modification des informations du client SIMO sachant qu'elle réside désormais à Douala

```
UPDATE Client
SET adresse = "Douala"
WHERE Id_Clt="Clt001" ;
```

3. Requête SQL de sélection de données

Pour afficher les enregistrements d'une table, on se sert de la commande SELECT. Cette sélection peut concerner toutes les colonnes ou certaines colonnes de la table, et peut être fonction d'aucune, d'une ou même de plusieurs conditions. La syntaxe est la suivante :

```
SELECT liste attributs
FROM nomTable
[WHERE condition] ;
```

NB : lorsque la sélection concerne tous les attributs d'une table, la liste des attributs peut être remplacée par « * ».

Exemple : Affichage de la liste des clients qui résident à Douala,

```
SELECT * FROM Client WHERE adresse="Douala" ; ou SELECT Id_Clt, nom,
adresse FROM Client WHERE adresse="Douala" ;
```

Exemple Afficher la liste des produits (désignation, prix_unit) dont le prix_unit est supérieur à 10000
SELECT désignation, prix_unit FROM Client WHERE prix_unit > 10000 ;

4. Requête SQL de suppression de données

Pour supprimer un ou plusieurs enregistrements d'une table, on se sert de la commande DELETE. Sa syntaxe est la suivante :

```
DELETE FROM nom_table [WHERE condition] ;
```

NB : En l'absence de clause WHERE, toutes les lignes de la table sont supprimées.

Exemple : Ecrire la requête SQL permettant de supprimer tous les produits dont prix_unit est comprise entre 5000 et 10000 :

```
DELETE FROM Produit WHERE prix_unit >= 5000 AND prix_unit <= 10000 ;
```

Exemple : Ecrire la requête SQL permettant de supprimer tous les produits dont **prix_unit** est inférieure à 500 ou supérieur à 10000 :

```
DELETE FROM Produit WHERE prix_unit<500 OR prix_unit>10000 ;
```

II. LES OPERATEURS EN LANGAGE SQL

I. Opérateurs de comparaisons

Il existe plusieurs opérateurs de comparaisons. La liste ci-jointe présente quelques uns des opérateurs les plus couramment utilisés.

| Opérateur | Description |
|-----------|----------------------------------------------------------------------|
| = | Égale |
| < > | Pas égale |
| != | Pas égale |
| > | Supérieur à |
| < | Inférieur à |
| >= | Supérieur ou égale à |
| <= | Inférieur ou égale à |
| IN | Liste de plusieurs valeurs possibles |
| BETWEEN | Valeur comprise dans un intervalle donnée (utile pour les nombres ou |
| LIKE | Recherche en spécifiant le début, milieu ou fin d'un mot. |

2. LES OPERATEURS LOGIQUES AND & OR

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.

- L'opérateur AND permet de s'assurer que la condition1 ET la condition2 sont vraies:

```
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 ;
```

NB : 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 prochaines commandes, nous allons considérer la table « produit » suivante :

| id | nom | categorie | stock | prix |
|----|------------|--------------|-------|------|
| 1 | ordinateur | informatique | 5 | 950 |
| 2 | clavier | informatique | 32 | 35 |
| 3 | souris | informatique | 16 | 30 |
| 4 | crayon | fourniture | 147 | 2 |

Opérateur AND

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 ;
```

Résultat :

| id | nom | categorie | stock | prix |
|----|------------|--------------|-------|------|
| 1 | ordinateur | informatique | 5 | 950 |
| 3 | souris | informatique | 16 | 30 |

Opérateur OR

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' ;
```

Résultats :

| id | nom | categorie | stock | prix |
|----|------------|--------------|-------|------|
| 1 | ordinateur | informatique | 5 | 950 |
| 2 | clavier | informatique | 32 | 35 |

3. L'OPERATEUR IN

L'opérateur logique IN est une méthode simple pour vérifier si une colonne est égale à une valeur OU une autre valeur OU une autre valeur et ainsi de suite, sans avoir à utiliser de multiple fois l'opérateur OR.

Syntaxe

Pour chercher toutes les lignes où la colonne « nom_colonne » est égale à 'valeur 1' OU 'valeur 2' ou 'valeur 3', il est possible d'utiliser la syntaxe suivante :

```
SELECT nom_colonne FROM table
WHERE nom_colonne IN ( valeur1, valeur2, valeur3, ... ) ;
```

Cette syntaxe correspond encore à :

```
SELECT nom_colonne
FROM table
WHERE nom_colonne=valeur1 OR nom_colonne=valeur2 OR nom_colonne=valeur3 ;
```

Exemple

Imaginons une table « adresse » qui contient une liste d'adresse associée à des utilisateurs d'une application.

| id | id_utilisateur | addr_rue | addr_code_postal | addr_ville |
|----|----------------|-----------------------------|------------------|------------------|
| 1 | 23 | 35 Rue Madeleine Pelletier | 25250 | Bournois |
| 2 | 43 | 21 Rue du Moulin Collet | 75006 | Paris |
| 3 | 65 | 28 Avenue de Cornouaille | 27220 | MousseauxNeuille |
| 4 | 67 | 41 Rue Marcel de la Provoté | 76430 | Graimbouville |
| 5 | 68 | 18 Avenue de Navarre | 75009 | Paris |

Si l'on souhaite obtenir les enregistrements des adresses de Paris et de Graimbouville, il est possible d'utiliser la requête suivante :

```
SELECT *
FROM adresse
WHERE addr_ville IN ('Paris', 'Graimbouville');
```

Résultats :

| id | id_utilisateur | addr_rue | addr_code_postal | addr_ville |
|----|----------------|-----------------------------|------------------|---------------|
| 2 | 43 | 21 Rue du Moulin Collet | 75006 | Paris |
| 4 | 67 | 41 Rue Marcel de la Provoté | 76430 | Graimbouville |
| 5 | 68 | 18 Avenue de Navarre | 75009 | Paris |

JEU BILINGUE :

- ✓ **INSERT** : insérer
- ✓ **INTO** : dans
- ✓ **VALUES** : valeurs
- ✓ **UPDATE** : mettre à jour
- ✓ **SET** : ensemble
- ✓ **WHERE** : où
- ✓ **SELECT** : sélectionner
- ✓ **FROM** : de
- ✓ **DELETE** : supprimer

EXERCICES

Soit la table Elève (matricule, nom, prenom, age, classe, moyenne, genre, statut) suivante :

- 1) Insérer 10 enregistrements dans cette table, ayant respectivement pour matricule *Elv001*, *Elv002*, *Elv003*, *Elv004*, *Elv005*, *Elv006*, *Elv007*, *Elv008*, *Elv009*, *Elv010*
- 2) Mettez à jour le nom de l'élève *Elv002* en **OUAFO**
- 3) Modifier le nom de l'élève ayant pour matricule *Elv001* en **SIMO**, son âge en 20 ans, et le statut en **redoublant**.
- 4) Modifier le prénom de l'élève ayant pour matricule *Elv010* en **PEGUY**, son âge en 17 ans, sa classe est **terminale** et le statut en **nouveau**.
- 5) Afficher la liste de tous les élèves
- 6) Afficher la liste des élèves redoublants

- 7) Afficher la liste des filles redoublantes
- 8) Afficher la liste des élèves de terminale dont l'âge est inférieur à 15 ans ou supérieur à 25 ans
- 9) Supprimer de la table tous les élèves dont l'âge est supérieur à 21 ans et qui sont redoublants
- 10) Donner le résultat de la requête suivante :

SELECT * FROM Elève WHERE moyenne >= 10 AND classe = "Terminale". Que fait cette requête ?

UE 20 : Implémentation une base de donnée dans Ms Access 2H

Compétences visées :

- Créer une base de données vide sous Ms Accès 2010
- Créer une table en mode graphiques dans la base de données sous Ms Accès 2010
- Établir les relations entre les tables d'une base de données
- Créer des requêtes de manipulation des tables

Situation problème : Pour votre stage de vacances dans la quincaillerie de votre oncle, vous constatez la difficulté qu'il a à faire tous ses comptes journaliers dans un cahier. Après avoir suivi les enseignements sur les bases de données, vous souhaitez réaliser une base de données pour faciliter les travaux de votre oncle.

Consigne :

1. Définir: Bases de données; SGBD
2. Donner quelques exemples de SGBD

INTRODUCTION GENERALE

Base de données : Une base de données, comme on la définit, est un ensemble structuré de données enregistré sur des mémoires secondaires créé et tenu à jour pour les besoins d'un ensemble d'utilisateurs. Les bases de données contiennent des informations nécessaires au fonctionnement des entreprises, des sociétés et des établissements (gestion comptable, gestion commerciale, gestion de production, gestion des stocks,...)

I. RAPPEL DE QUELQUES CONCEPTS DE BASE D'UNE BASE DE DONNEES

- Une base de données est un ensemble ou une collection structurée de données.
- Il existe plusieurs types de bases de données (BD's) qui se distinguent par la façon dont les données sont structurées.
- BD's relationnelles, objet, hiérarchiques, réseau, fonctionnelles, déductives ...
- 95% des BD's sont relationnelles :
 - Oracle, société éditrice de logiciels pour bases de données est la deuxième plus grosse société après Microsoft.
- Une BD relationnelle est composée d'un ensemble de **tables** (ou relations).
- Une table est composée de
 - Lignes qu'on appelle **enregistrements** (ou tuples)
 - Colonnes représentant chacune **un champ** (ou un attribut).

- Chaque table a un nom.
- Chaque champ a un nom et un type : Texte, numérique, date, ...

Exemple de table

Nom de la table → **Commandes**

3 champs

4 Enregistrements

| NumCommande | DateCommande | Montant |
|-------------|--------------|---------|
| 27 | 13/2/2007 | 120 |
| 65 | 12/1/2008 | 34 |
| 2 | 10/06/2006 | 27 |
| 34 | 14/12/2007 | 1500 |

Les types de champs

N°Commande est du type *numérique entier*

Montant est du type *numérique réel*

DateCommande est du type *date*

Quelques contraintes

- Deux tables d'une même base de données ne peuvent pas avoir le même nom.
- Deux champs de la même table ne peuvent pas avoir le même nom.
- Un même champ peut être présent dans plusieurs tables.

Exemple de tables d'une base de données

| NumAuteur | NomAuteur |
|-----------|-----------|
| 1 | Dupont |
| 2 | Durand |
| 3 | Martin |

| NumLivre | TitreLivre |
|----------|------------|
| 10 | La monnaie |
| 25 | La finance |

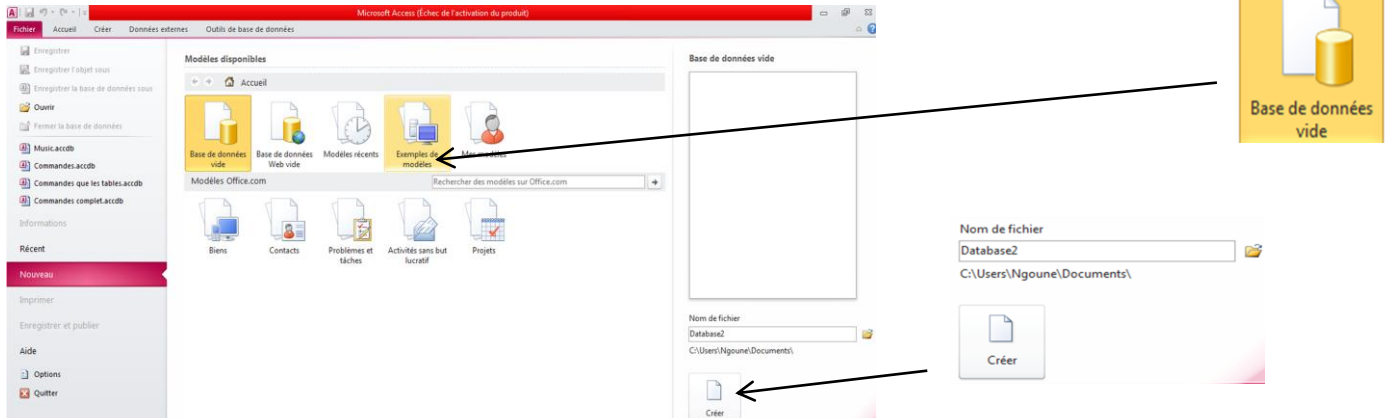
| NumLivre | NumAuteur |
|----------|-----------|
| 10 | 1 |
| 25 | 1 |
| 25 | 3 |

SGBD

- Les logiciels qui permettent de gérer des bases de données sont appelés « *Systèmes de Gestion de Bases de Données* ».
- Acces est un SGBD relationnel.
- MySQL, SQL Server, Postgres, Oracle, DB2, ...
- Un SGBD permet de
 - ✓ Créer une BD
 - ✓ Modifier la structure des tables
 - ✓ Interroger la BD
 - ✓ Modifier la BD
 - ✓ ...

II. REALISATION D'UNE BASE DE DONNEES

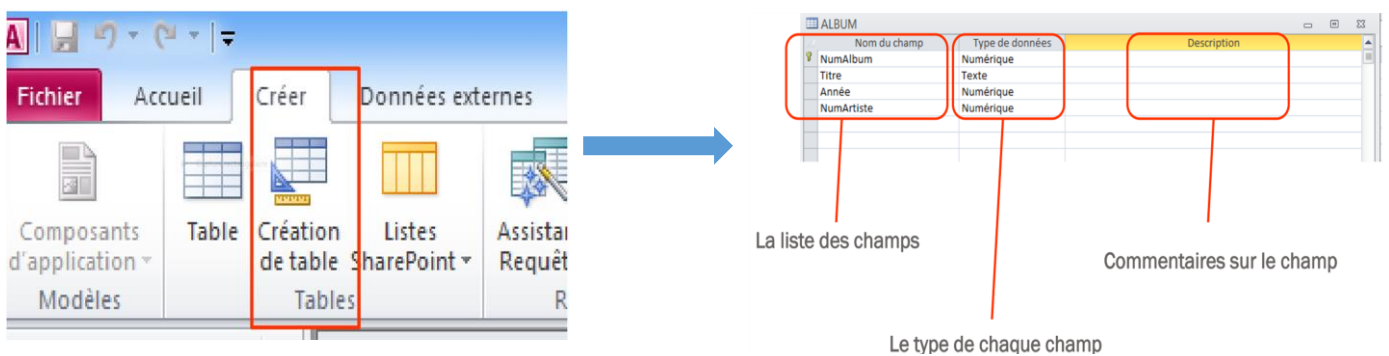
I. Créer une Base de données vide



Donner un nom à votre base de données

2. Créer Les tables

Création d'une table en mode création



- Une fois qu'on a fini de taper tous les champs composant la table,
- Fermer la fenêtre de création,
- Access demande alors le nom de la table,
- Access va demander aussi de préciser la **clé primaire**. On reviendra plus tard sur cette notion,
- Ça y est, notre table « Album » est créée,
- Pour afficher son contenu, il suffit de cliquer deux fois dessus,
- On peut ensuite saisir, supprimer, modifier des enregistrements.

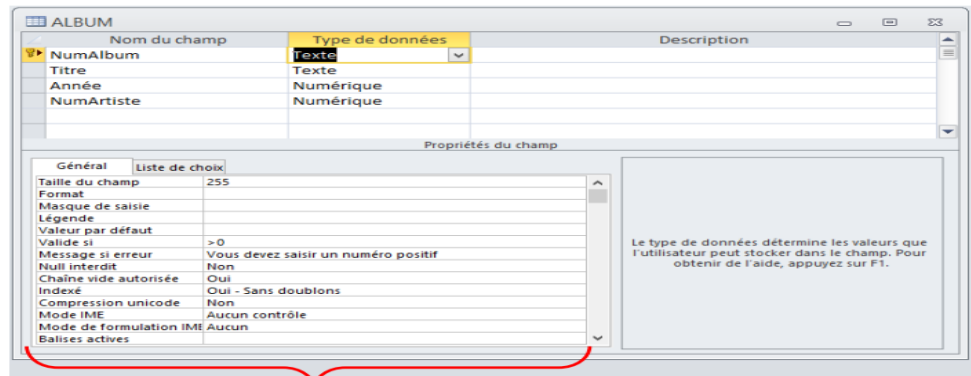
Les types des champs

- **NuméroAuto** : numéro incrémenté à chaque insertion d'un nouvel enregistrement.
- **Numérique**: C'est à l'utilisateur de taper sa valeur. Par défaut, il s'agit d'un entier mais on peut préciser si c'est un réel.
- **Texte** : c'est une chaîne de caractères. On peut préciser sa taille.
- **Oui/Non** : ça correspond aux champs qui ne peuvent prendre que l'une des deux valeurs OUI ou

NON.

- **Date/Heure** : Type des champs qui indiquent une notion de temps. Plusieurs formats sont disponibles.
- D'autres types encore mais rarement utilisés.

Propriétés d'un champ

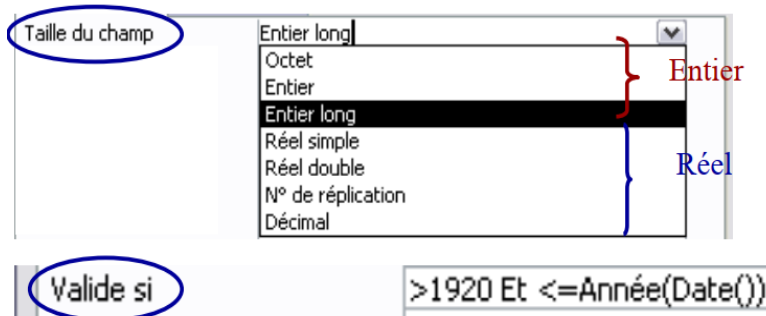


Les propriétés qu'on peut préciser pour un champ

Propriétés du type Texte

- Les plus utilisées :
 - **Taille du texte** : en nombre de caractères maximum,
 - **Valeur par défaut** : valeur prise par ce champs si l'utilisateur, lors de l'insertion d'un nouvel enregistrement ne précise pas de valeur,
 - **Null interdit** : Est-ce que l'utilisateur est obligé de donner une valeur pour ce champ ou pas,
 - **Indexé** : permet d'optimiser la recherche sur ce champ. On peut utiliser cette propriété pour exiger que les valeurs du champ doivent être uniques (sans doublons),
 - **Liste de choix** : permet de préciser l'ensemble des valeurs correctes.

Propriétés du type numérique



Notion de clé primaire

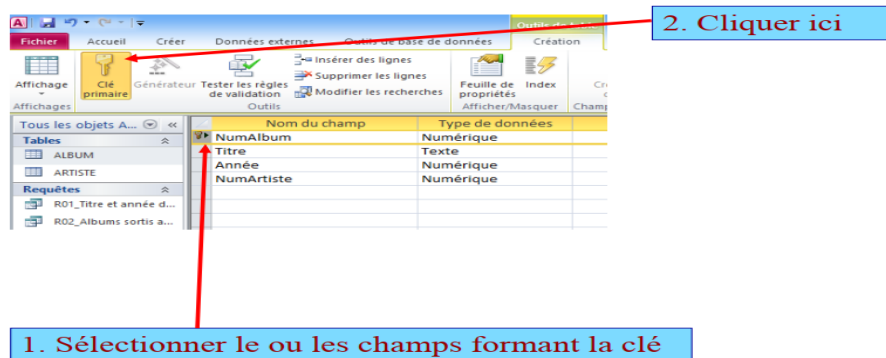
- Une clé primaire dans une table est un champ ou un ensemble de champs qui permet **d'identifier** chaque ligne dans la table. Deux enregistrements ne doivent pas avoir la même valeur pour la clé

- Par exemple: dans la table Album, le champ **NumAlbum** est une clé primaire car deux albums ne doivent pas avoir le même numéro.
- Le champ Année n'est pas une clé primaire car deux albums peuvent avoir la même année de sortie.

Conséquences de la déclaration d'une clé primaire

- Le système (Access) va **refuser l'insertion** d'un nouvel enregistrement si cela viole l'unicité de valeur pour la clé primaire. S'il y a déjà un enregistrement avec la même valeur pour la clé primaire, le système **va refuser la modification** de la clé primaire si la nouvelle valeur existe déjà.

Déclaration d'une clé primaire



1. Sélectionner le ou les champs formant la clé

- Un champ (ou ensemble de champs) est clé étrangère dans une table s'il fait référence à une clé primaire dans une autre table.
- Par exemple : le champ **NumArtiste** est clé primaire dans la table Artiste, ce champ est donc clé étrangère dans la table **Album**.
- On parle dans ce cas d'intégrité référentielle.

Conséquences de la déclaration d'une clé étrangère

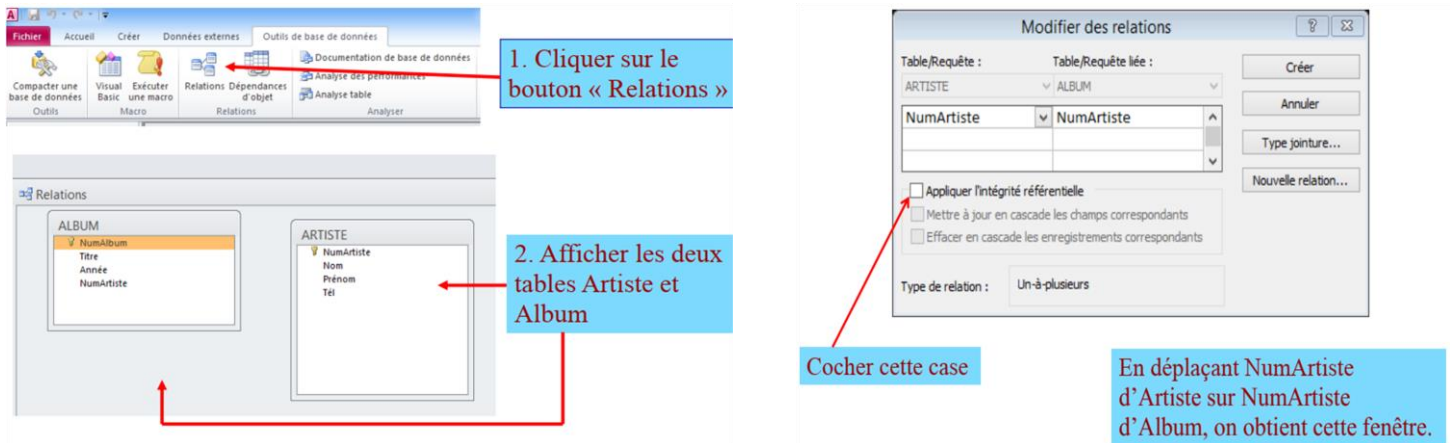
- Le système **va refuser l'insertion d'un album** si son NumArtiste n'existe pas dans la table Artiste.
- Le système **va refuser la suppression d'un artiste** s'il existe dans la table Album des enregistrements qui lui sont associés.
- Le système **va refuser la modification d'un NumArtiste dans Album** si la nouvelle valeur n'est pas présente dans Artiste.
- Le système **va refuser la modification d'un NumArtiste dans Artiste** s'il y a déjà des albums qui sont associés à l'ancienne valeur.

3. Établir Les relations

Pour créer la relation entre la table ELEVE et la table NOTE on peut suivre la démarche suivante :

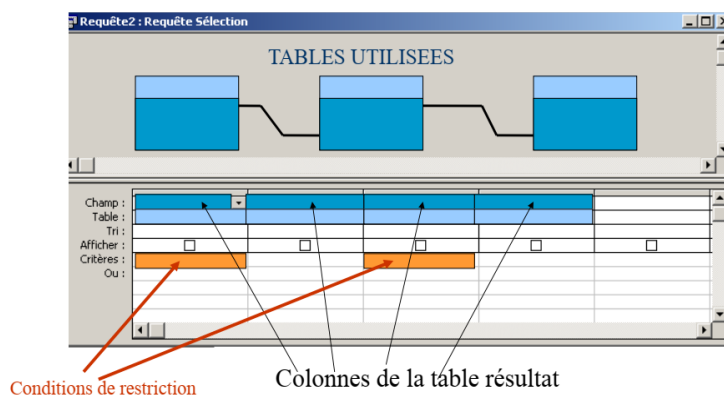
- Ouvrir le menu «Outils» puis cliquer sur la commande «Relations...»
- Ajouter les tables Album et Artiste en question

- Sélectionner la clé primaire NumArtiste de la table Artiste et le glisser, en maintenant le bouton gauche de la souris, vers la clé étranger de la table Album
- Vérifier, dans la boîte de dialogue suivante, les champs puis cocher la case «Appliquer l'intégrité référentielle»



III. CREATION DE REQUETES

INTERFACE GRAPHIQUE DE FORMULATION DES REQUETES



I. Les requêtes de sélection

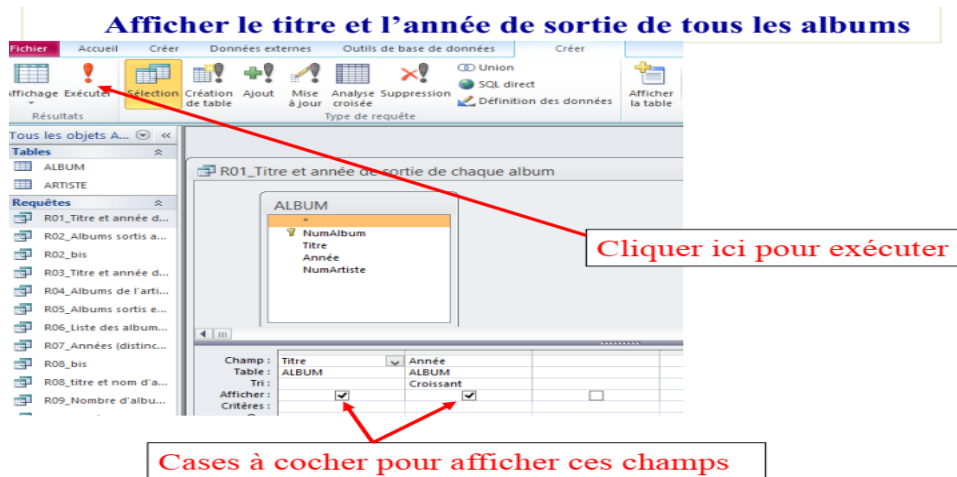
La sélection représente l'outil courant de recherche d'informations dans les bases de données. D'une manière générale, une requête de sélection :

- S'applique soit à une seule table, soit à plusieurs tables liées par des relations ;
- Permet de sélectionner les lignes par application d'un ou de plusieurs critères portant sur un ou sur plusieurs champs ;
- Permet de choisir les colonnes que l'on veut conserver (comme la sélection simple);
- Peut enregistrer le résultat recherché sous forme d'une table.

Exemple: Donner la liste des titres et années de sortie de tous les albums.

Pour créer cette requête, on peut suivre la démarche suivante :

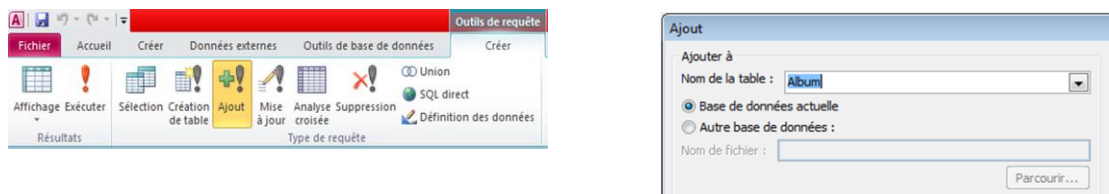
- Choisir l'objet « Requetes » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau »
- Choisir le mode de création ; par exemple « Mode création »
- Dans la fenêtre « Afficher la table », ajouter la table Album puis la fermer



2. Les requêtes d'ajout

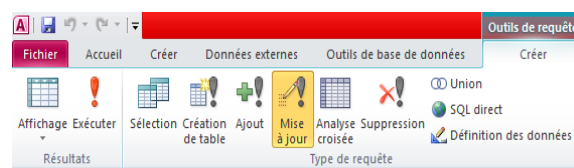
Une requête d'ajout ajoute un groupe d'enregistrements d'une ou de plusieurs tables à la fin d'une ou de plusieurs tables

- Choisir l'objet « Requetes » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode « Mode création »
- Ajouter la table Album qui représente la source des informations de la requête
- On remarque l'ajout du nouveau menu « Requête » à la barre des menus.



3. Les requêtes de mise à jour

Une requête de Mise à jour apporte des changements globaux à un groupe d'enregistrements dans une ou plusieurs tables.



IV. CREATION DE FORMULAIRES

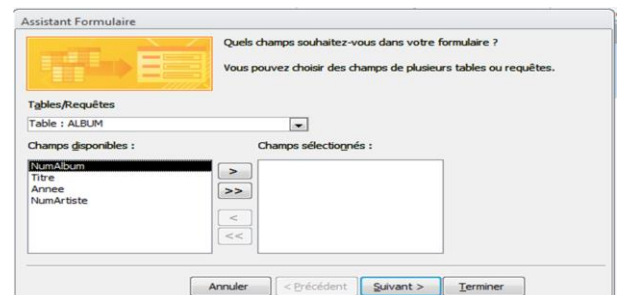
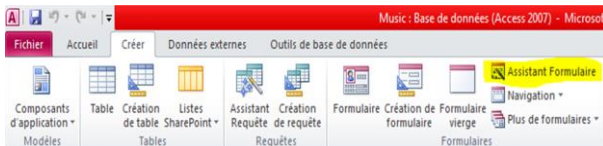
Un formulaire est un type d'objet d'une base de données utilisé essentiellement pour saisir et afficher les données dans une base de données. On peut également utiliser un formulaire comme :

- Menu général qui ouvre d'autres formulaires, des requêtes et des états dans la base de données,
- Boîte de dialogue personnalisée qui permet aux utilisateurs d'entrer des données et de les utiliser.

Procédure de création d'un formulaire

Le formulaire à créer est utilisé pour saisir les données dans la table ALBUM. Pour cela, il est possible d'utiliser l'assistant des formulaires comme indique la démarche suivante :

- Choisir l'objet « Formulaires » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir l'un des modes de création (par exemple, assistant formulaire).

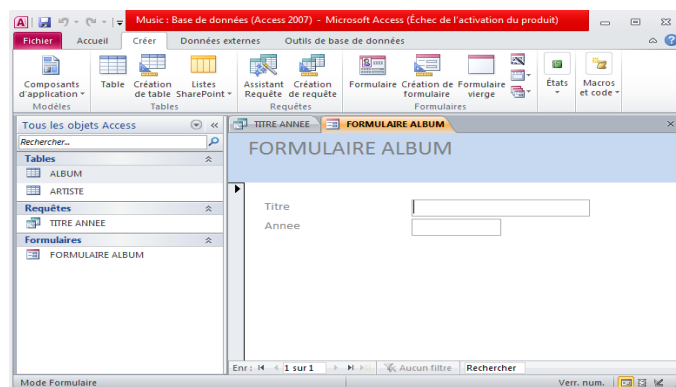


- Ajouter les champs qu'on veut afficher dans le formulaire.
- Choisir la liste des champs, le style et le nom du formulaire puis l'enregistrer.

Le formulaire s'affiche en mode affichage avec un état initial.

On peut modifier (ajout, suppression, mise en forme, ...) les objets du formulaire en basculant du mode affichage au mode création.

Après toutes ces étapes, on obtient



EXERCICE

Soit le schéma de la base de données "compétition" suivante :

PAYS (CodePays, NomPays)

EQUIPE (CodeEquipe, NomEquipe, DirecteurSportif)

ETAPE (NumeroEtape, Date, VilleDep, VilleArr, NbKm)

COUREUR (NumeroCoureur, NomCoureur, #CodeEquipe, #CodePays)

PARTICIPER (#NumeroCoureur, #NumeroEtape, TempsRealise).

1. Créer les tables de la base de données.
2. Créer un formulaire de saisir pour les entités COUREUR, EQUIPE, ETAPE et PARTICIPER.
3. Créer un formulaire qui joue le rôle d'un menu principal de la base de données.
4. Créer un état permettant l'impression de tous les coureurs tunisiens

Jeu bilingue : Traduire en français ou en anglais les mots ou expression du tableau ci-dessous :

| Français | Anglais |
|-----------------|--------------|
| Base de données | Data Base |
| Relation | Relationship |

Module 3 : ALGORITHMIQUE ET PROGRAMMATION

UA 6: ALGORITHMIQUE 6H

UE 2I : Généralités Sur Les Algorithmes Et Structure Des Données 3H

Compétences:

- Enoncer les généralités sur les algorithmes
- Décrire les structures de contrôle ;
- Citer cinq structures de données ;
- Déclarer un tableau ;
- Parcourir un tableau pour effectuer la lecture et l'affichage ;

Situation problème

Lors de votre stage pratique au sein de l'entreprise LIS (Laboratoire informatique du sahel), il vous est demandé d'aider l'informaticien de cette entreprise à concevoir un système de gestion de personnels. A cet effet ce dernier vous demande de l'aider à trouver un moyen algorithmique que vous pourriez utiliser pour stocker les informations sur les personnels.

Consigne :

1. Rappeler la définition et les parties d'un algorithme.
2. Ecrire la structure d'un algorithme.
3. Pouvez-vous utiliser le tableau pour stocker les informations sur les personnels de cette entreprise? Justifier.
4. Rappeler la syntaxe de déclaration d'un tableau
5. Comment lire et afficher les données d'un tableau.
6. Enumérer 04 autres structures de données que vous pourriez utiliser pour résoudre ce problème.
7. Faire la différence entre le tableau et le type enregistrement.

RESUME

I. Rappels

Un **algorithme** est une suite ordonnée d'instructions qui indique la démarche à suivre pour résoudre un problème précis.

La science qui étudie les algorithmes est l'**algorithmique**.

Les objets (ou données) qui seront manipulés dans l'exécution d'un algorithme sont entre autres des variables et des constantes.

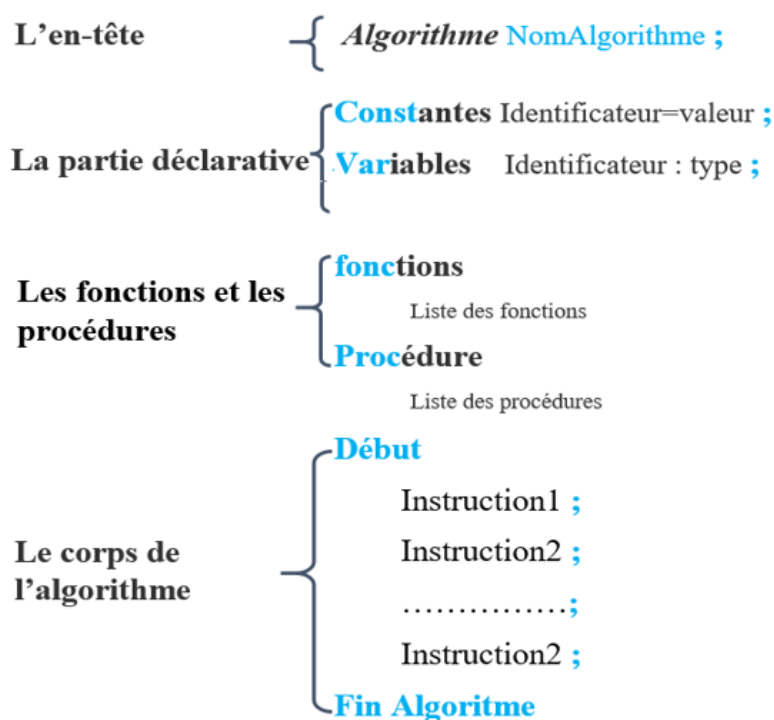
Une **constante** : représente des objets (nombre, caractères,) dont la valeur ne peut pas être modifiée pendant l'exécution de l'algorithme.

Les **variables** : représentent les objets dont la valeur peut être modifiée au cours de l'exécution de l'algorithme.

Un algorithme a généralement 03 parties :

- **L'entête** : Elle permet tout simplement d'identifier l'algorithme en précisant son nom.
- La **partie déclarative** : Elle présente la liste exhaustive des objets, grandeurs utilisés et manipulés dans l'algorithme.
- **Le corps d'algorithme** : Il est délimité par les termes DEBUT et FIN. Il contient un ensemble d'instructions (les tâches) à exécuter selon un ordre précis.

En résumé, la structure d'un algorithme est donnée par le schéma ci-dessous :



Le corps de l'algorithme, comporte toutes les tâches possibles énumérées selon un ordre précis pour une exécution automatique d'un problème quelconque. Ces tâches sont décrites par les instructions.

En algorithmique, quelques instructions de base sont :

- **Les instructions de sortie ou d'affichage** : utilisées pour afficher un message ou le contenu d'une variable. Ce sont : **Ecrire ()** et **Afficher ()**
- **Les instructions d'entrée ou de lecture** : utilisées pour lire les valeurs des variables saisies au clavier. On a comme instructions de lecture : **saisir ()** et **Lire ()**
- **L'affectation** : Elle permet d'attribuer une valeur à une variable. Elle est notée **Variable prend valeur** ou **Variable ← valeur**.

Exemple 1 :

L'algorithme qui calcule et affiche la somme de deux nombres entiers saisi au clavier est :

Algorithme Somme
Var a, b, S: entier ;
Début
Ecrire ("Entrer le premier nombre") ;
Lire(a) ;
Ecrire ("Entrer le deuxième nombre") ;
Lire (b) ;
 $S \leftarrow a+b$;
Ecrire ("La somme de ", a, "et" , b ,"est s = ",S) ;
Fin.

2. Structures des contrôle

Les opérations élémentaires relatives à la résolution d'un problème peuvent, en fonction de leur enchaînement, être organisées suivant quatre familles de structures algorithmiques fondamentales : Structures linéaires, structures alternatives, structures de choix et structure itératives (ou répétitives).

| Structures | Description | Syntaxe |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Linéaires ou séquentielles | Se caractérise par une suite d'actions à exécuter successivement dans l'ordre énoncé. Les actions successives sont mentionnées les unes après les autres. | Début Action 1 ; Action 2 ; ; Action n ; Fin |
| Alternatives ou conditionnelles | L'exécution d'une action distincte ne dépend que du résultat d'un test effectué sur la condition qui peut être une variable ou un événement. Si la condition est vérifiée, seule la première action est exécutée ; Si non , seule la deuxième action est effectuée. | Si condition Alors Action 1 ; Si non Action 2 ; FinSi |
| Itératives ou répétitives | Boucle Pour La structure itérative répète l'exécution d'une opération ou d'un traitement. | Pour i allant de val_début à val_fin pas val_pas faire |

| | | | |
|--|------------------------|----------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| | | Elles sont encore désignées sous le nom de boucle. | Action1 ; Action 2 ; FinPour |
| | Boucle Répéter | | Répéter Action1 ; Action2 ; Jusqu'à (condition) ; |
| | Boucle Tant que | | Tant que (condition vraie) Faire Action1 ; Action2 ; FinTantque |

Exemple 2 : L'algorithme qui Calcul x à la puissance n où x est un réel non nul et n un entier positif ou nul.

Algorithme Puissance ;

Var x, p : réel;

n, i : entier;

Début

Ecrire (" Entrez la valeur de x ");

Lire (x);

Ecrire (" Entrez la valeur de n ");

Lire (n);

p ← 1

Pour i allant de 1 à n **Faire**

p ← p*x ;

FinPour

Ecrire (x, "à la puissance ", n, " est égal à ", p);

Fin

3. Structures des données

Une **structure des données** est une méthode utilisée pour stocker et organiser les données dans un ordinateur de façon à les utiliser efficacement.

On distingue plusieurs structures de données en algorithmiques parmi lesquels :

- Les **Enregistrements** : Un enregistrement est un type de données défini par l'utilisateur et qui permet de grouper un nombre fini d'éléments (ou champs) de types éventuellement différents.

- Les **Piles** : Une pile est une structure de données dans laquelle on peut ajouter et supprimer des éléments suivant la règle du dernier arrivé premier sorti ou encore LIFO (Last In First Out).(Empiler ; Dépiler ; Vider ; Détruire ;Initialiser etc.)
- Les **Files** : Une file est une structure de données dans laquelle on peut ajouter et supprimer des éléments suivant la règle du premier arrivé premier sorti ou encore FIFO (Last In First Out).
- Les **Listes** : Une liste chaînée est un ensemble de cellules liées entre elles par des pointeurs. (Chaque cellule est une structure contenant les champs suivants : une ou plusieurs données comme dans n'importe quelle structure ; un pointeur suivant sur la cellule suivante).
- Les **Tableaux** : Un tableau est une structure de donnée ayant une taille fixe et qui permet de manipuler les données de même type

a. Utilisation des tableaux

Un **tableau** est une structure de donnée séquentielle qui permet de stocker un certain nombre d'éléments repérés par un index.

Les tableaux vérifient généralement les propriétés suivantes :

- Tous les éléments ont le même type de base ;
- Le nombre d'éléments stockés est fixé ;

La syntaxe de déclaration d'un tableau est donnée ci-dessous :

Nom_tableau = **Tableau** [borne_inf .. borne_sup] de type_élément ;

Exemple : declaration d'un tableau qui va contenir 20 nombres entiers.

Tab= Tableau[I..20] d'entiers ;

Le remplissage d'un tableau avec ses différents éléments peut se faire à l'aide de l'affectation ou par la lecture des valeurs saisi aux claviers.

Exemple :

- Tab[2]=3 permet d'ajouter la valeur 3 dans la case 2 du tableau nommé Tab.
- Pour remplir le tableau Tab de taille 4 avec les valeurs saisies par l'utilisateur, on écrira :

Pour i allant de 1 à 4 faire

Lire (Tab[i]) ;

FinPour

Pour afficher les différents éléments d'un tableau **Tab** de taille **n** les uns après les autres, on écrira :

Pour i allant de 1 à n faire

Ecrire (Tab[i]) ;

FinPour

NB : on peut aussi afficher directement l'élément d'un tableau connaissant son indice. Par exemple, l'élément se trouvant à l'indice 4 du tableau **Tab** sera affiché comme suit :

Ecrire (Tab[4])

b. Utilisation des enregistrements.

Il est possible de créer nos propres types, puis de déclarer des variables ou des tableaux d'éléments de ce type.

Pour ce faire, il faut déclarer un nouveau type, fondé sur d'autres types existants. Après l'avoir défini, on peut dès lors utiliser ce type structuré comme tout autre type normal en déclarant une ou plusieurs variables de ce type. Les variables de type structuré sont appelées **enregistrements**.

Contrairement aux tableaux qui sont des structures de données dont tous les éléments sont de même type, les enregistrements sont des structures de données dont les éléments peuvent être de différents types et qui se rapportent à la même entité sémantique. Les éléments qui composent un enregistrement sont appelés champs.

Avant de déclarer une variable enregistrement, il faut avoir au préalable défini son type, c'est à dire le nom et le type des champs qui le composent. Le type d'un enregistrement est appelé type structure. Un enregistrement est défini de la manière suivante.

Nom_enregistrement **Enregistrement**:

Champ1 : type_champ1

Champ2 : type_champ2

.....

Champ n : type_champ n

Fin **Enregistrement**;

Exemple :

Personne : **Enregistrement**

Nom : Chaîne

Prénom : Chaîne

Age : Entier

Fin **Enregistrement**

Une fois qu'on a défini un type structuré, on peut déclarer des variables enregistrements exactement de la même façon que l'on déclaré des variables d'un type primitif.

Syntaxe : Nom Var : Nom_Enregistrement

Exemple : P : Personne

Les enregistrements sont représentés par plusieurs zones de données, correspondant aux champs.

Exemple : la variable P de type personne que nous avons défini ci-dessus sera représentée comme suit :

| | Nom | Prénom | Age |
|----|----------------------|----------------------|----------------------|
| P: | <input type="text"/> | <input type="text"/> | <input type="text"/> |

La manipulation d'un enregistrement se fait à travers ses champs. Comme pour les tableaux, il n'est pas possible de manipuler un enregistrement globalement, sauf pour affecter un enregistrement à un autre de même type (ou le passer en paramètre). Par exemple, pour afficher un enregistrement il faut afficher tous ses champs un par un.

Alors que les éléments d'un tableau sont par l'intermédiaire de leur indice, les champs d'un enregistrement sont accessibles à travers leur nom, grâce à l'opérateur '.'

Nom Var.Nom Champ représente la valeur mémorisée dans le champ de l'enregistrement Par exemple, pour accéder à l'âge de la variable P, on utilise l'expression : **P.Age**

Remarque : la lecture d'une telle expression se fait de droit à gauche : l'âge de la personne P.

Attention : le nom d'un champ est **TOUJOURS** précédé du nom de la variable déclarée avec le type enregistrement auquel il appartient. On ne peut pas trouver un nom de champ tout seul, sans indication de la variable.

Les champs d'un enregistrement, tout comme les éléments d'un tableau, sont des variables à qui on peut faire subir les mêmes opérations (affectation, saisie, affichage...).

Exercices

1. Ecrire un algorithme qui calcule n !
2. Ecrire un algorithme qui affiche la table de multiplication d'un nombre
3. Ecrire un algorithme qui sauvegarde 50 nombres saisi par l'utilisateur dans un tableau puis les affiche.
4.
 - a. Créer type enregistrement qu'on peut utiliser pour sauvegarder les informations sur un élève sachant que ce dernier possède un nom, un prénom, une date et lieu de naissance, l'âge et le sexe.
 - b. Ecrire un algorithme qui permet d'enregistrer deux éléments qui compare leurs âges et affiche le plus âgé.

UE 22 : Recherche séquentielle dans un tableau 2H

Exemples d'action :

- Ecrire un algorithme de recherche séquentielle

Situation problème : Après avoir étudié les structures de données dans les leçons précédentes, on vous demande à présent de retrouver une valeur dans un tableau dont vous ignorez le contenu.

Consigne :

1. Citer deux structures de données
2. Définir tableau et citer ses éléments caractéristiques
3. Comment savoir qu'une valeur est présente dans un tableau?
4. En quoi consiste la recherche séquentielle?

Résumé

La recherche séquentielle ou linéaire est un algorithme qui permet de trouver une valeur dans une structure de données (notamment un tableau dans notre cas).

I- Principe

Le principe est simple, on va parcourir les cases du tableau dans l'ordre croissant des indices jusqu'à ce qu'on trouve l'élément recherché, ou bien jusqu'à ce qu'on arrive à la fin du tableau, sans avoir trouvé l'élément recherché (dans ce cas l'élément recherché ne se trouve pas dans le tableau).

II- Algorithme

Soit le tableau suivant de taille 4 nommé Tab :

| | | | |
|---|---|---|---|
| 6 | 3 | 1 | 9 |
|---|---|---|---|

Soit à rechercher la valeur 1 dans ce tableau.

Algorithme : RechercheSequentielle

Var : valeur, i : entier ;

Tab : Tableau [1..4] d'entier ;

Début

//remplissage du tableau

Tab[1] ← 6 ;

Tab[2] ← 3 ;

Tab[3] ← 1 ;

Tab[4] ← 9 ;

//valeur à rechercher et initialisation de

l'indice

valeur ← 1 ;

indice ← 1 ;

// on compare valeur et tab[indice] pour trouver valeur dans Tab

TANTQUE (valeur <> Tab[indice] && indice <= 4) **FAIRE**

indice ← indice + 1 ;

FINTANTQUE

SI (Tab[indice] = valeur) **ALORS**

ecrire("la valeur se trouve à l'indice", indice);

SINON

ecrire(" valeur non présente");

FINSI

FIN

Exercice

Soit le tableau suivant nommé Note :

| | | | | | | |
|----|---|----|---|----|----|---|
| 12 | 5 | 17 | 3 | 13 | 15 | 2 |
|----|---|----|---|----|----|---|

Ecrire un algorithme qui recherche et affiche l'indice de la note 6

Jeu bilingue

Valeur = value

UE 23 : Recherche du maximum/minimum dans un tableau IH

Pré requis

- Déclarer un tableau à une dimension
- Parcourir un tableau à une dimension
- Afficher les éléments d'un tableau

Compétences visées :

A la fin de cette unité d'enseignement, l'élève doit mobiliser les connaissances lui permettant de :

- Rechercher le plus grand élément d'un tableau
- Rechercher le plus petit élément d'un tableau
- Rechercher et afficher le maximum et minimum des valeurs d'un tableau à une dimension

Situation problème :

Votre enseignant de mathématiques dispose des notes des élèves de sa classe de terminale C. Il aimerait bien avoir en sa possession une petite application qui va lui permettre après chaque évaluation de traiter (calculer la moyenne, rechercher et afficher la plus petite note, rechercher et afficher la plus grande note, ...) de façon automatique ses différentes notes après saisie. Pour cela, il sollicite votre aide.

Consigne :

1. Quelle structure de données déjà vu en cours allez-vous utiliser pour résoudre ce problème ?
Vous décidez de stocker les différentes notes de la classe dans un tableau NOTES.
2. Rechercher et donner la plus grande valeur du tableau NOTES.
3. Rechercher et donner la plus petite valeur du tableau NOTES.

Contenu de l'UE :

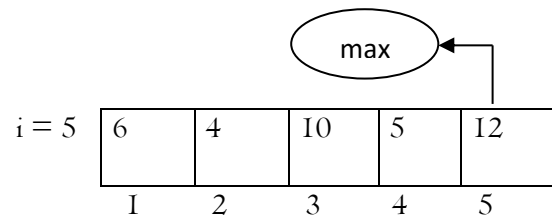
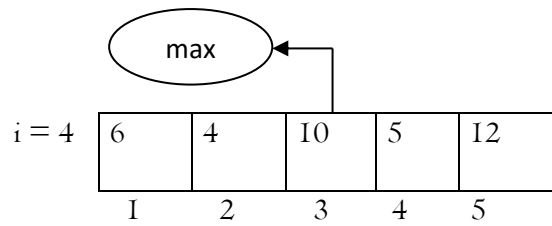
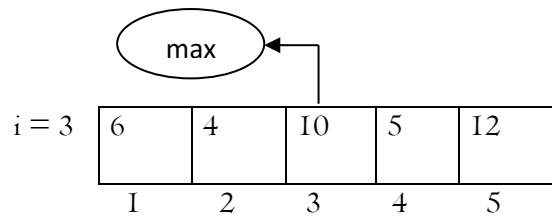
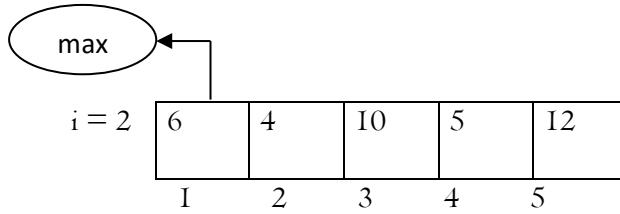
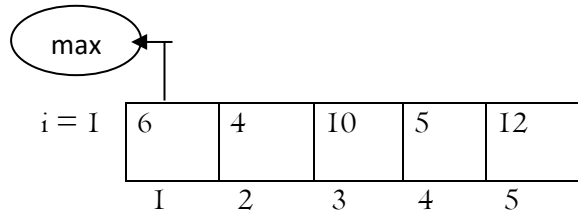
INTRODUCTION

L'un des exercices classiques d'initiation à la programmation consiste à rechercher le maximum et le minimum dans un tableau d'entiers. La recherche s'effectue en considérant le premier élément du tableau comme étant le maximum (respectivement. minimum), et en parcourant le tableau itérativement. Si à l'itération i la valeur se trouvant à l'indice i est supérieure (respectivement. inférieure) à la valeur considéré maximale (respectivement. minimale) au préalable, la valeur à l'indice i devient alors le maximum (respectivement. minimum).

I. Recherche du plus grand élément d'un tableau

a) Principe

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 1. Lorsque la valeur à l'indice i est plus grande que *max*, *max* devient alors la valeur à l'indice i



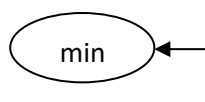
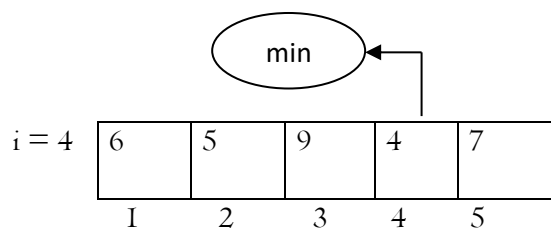
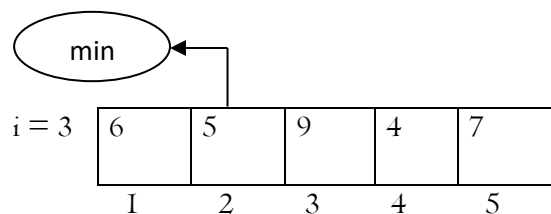
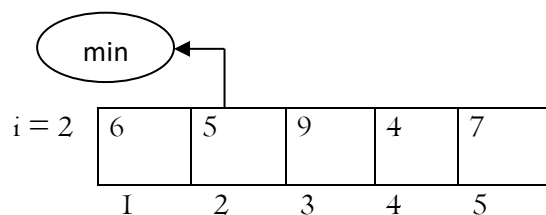
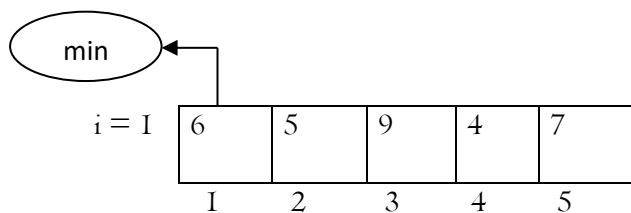
b) Algorithmme

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> Algorithm Recherche_Max var TAB : tableau[I..n] de entier i, max : entier début /*Saisie de la taille du tableau*/ Ecrire("Donner la taille du tableau :") Lire(n) /*Saisie des éléments du tableau*/ Pour i ← I à n faire débutPour écrire("Donner l'élément N°", i) lire TAB[i] finPour </pre> | <pre> /*Recherche du plus grand élément dans le tableau TAB*/ max ← TAB[I]; Pour i ← 2 à n faire débutPour si (TAB[i] > max) max ← TAB[i]; finsi finPour /*Affichage du plus grand élément du tableau TAB*/ Ecrire("Le plus grand est : ", max); fin. </pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

2. Recherche du plus petit élément d'un tableau

a) Principe

L'exemple suivant détaille les différentes itérations de la recherche du minimum dans un tableau d'entiers : La variable min prend initialement la valeur se trouvant à l'indice I. Lorsque la valeur à l'indice i est plus petite que min, min devient alors la valeur à l'indice i .



| | | | | | |
|---------|---|---|---|---|---|
| $i = 5$ | 6 | 5 | 9 | 4 | 7 |
| | 1 | 2 | 3 | 4 | 5 |

b) Algorithme

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> Algorithme Recherche_Min var TAB : tableau[1..n] de entier i, min : entier début /*Saisie de la taille du tableau*/ Ecrire("Donner la taille du tableau :") Lire(n) /*Saisie des éléments du tableau*/ Pour i ← 1 à n faire débutPour écrire("Donner l'élément N°", i) lireTAB[i] finPour </pre> | <pre> /*Recherche du plus petit élément dans le tableau TAB*/ min=TAB[1]; Pour i ← 2 à n faire débutPour si (TAB[i]<min) min←TAB[i]; finsi finPour /*Affichage du plus petit élément du tableau TAB*/ Ecrire("Le plus petit est : ", min); fin. </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

3. Algorithme du maximum/minimum d'un tableau à une dimension

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> Algorithme MinMax var TAB : tableau[1..n] de entier i, min, max : entier début /*Saisie de la taille du tableau*/ Ecrire("Donner la taille du tableau :") Lire(n) /*Saisie des éléments du tableau*/ Pour i ← 1 à n faire débutPour écrire("Donner l'élément N°", i) lireTAB[i] finPour /*Recherche du max/min dans le TAB*/ </pre> | <pre> max=TAB[1]; min=TAB[1]; Pour i ← 2 à n faire débutPour si (TAB[i]<min) alors min←TAB[i]; sinon max←TAB[i]; finsi finPour /*Affichage */ Ecrire("Le plus petit est : ", min); Ecrire("Le plus grand est : ", max); fin. </pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Jeu bilingue

Traduire en français ou en anglais les mots du tableau ci-dessous :

| Français | Anglais |
|----------|---------|
| Tableau | Array |
| Minimum | Minimum |
| Maximum | Maxmum |

Activités d'intégration

I. On considère le tableau suivant :

| | | | | | | | | | |
|---|---|---|----|---|----|---|----|----|----|
| T | 6 | 4 | 10 | 5 | 12 | 9 | 13 | 19 | 15 |
|---|---|---|----|---|----|---|----|----|----|

- Quelle est la taille du tableau T ?
 - Donner les indices du premier et du dernier élément de ce tableau.
 - Déclarer de deux façons le tableau T.
 - Ecrire le bout d'algorithme permettant de :
 - Saisir la taille du tableau.
 - Afficher le contenu du tableau.
 - Parcourir le tableau.
 - En considérant le tableau T, déterminé :
 - La position du minimum
 - La valeur du maximum
 - La position du maximum
 - La valeur du minimum
2. Soit un tableau de 12 nombres (entiers ou réels) compris entre 0 et 20. Ce sont les notes des élèves d'une classe.
- Toutes les questions de cet exercice sont d'abord à rédiger en pseudo-code avec un papier et un crayon. Une fois que le professeur a contrôlé votre travail, tester sur machine en C.
- Écrire une fonction **maxi**, dont le tableau est un paramètre, qui **retourne** le maximum du tableau. De même pour une fonction **mini**.
 - Écrire une procédure **indice**, qui recherche un nombre dans le tableau et **affiche** les indices où le nombre apparaît dans le tableau.
 - Écrire une fonction **moyenne**, qui retourne la moyenne des éléments du tableau.
 - Écrire une fonction **moyplus**, qui a pour paramètres le tableau et un nombre, et qui retourne le pourcentage de notes du tableau supérieures ou égales à ce nombre.
 - Écrire un programme principal utilisant les fonctions ci-dessus, qui fait saisir 12 notes dans un tableau, puis affiche la note maximale et le numéro des élèves qui ont la meilleure note, la note

minimale et le numéro des élèves qui ont la moins bonne note, la moyenne de la classe, le pourcentage de notes supérieures ou égales à 10, puis à 16.

3. En vue du conseil de classe, un professeur souhaite effectuer des statistiques sur sa classe de terminale qui compte 25 élèves.
 - a) Ecrire un algorithme qui permet la saisie des 25 notes dans un tableau.
 - b) Modifier cet algorithme pour qu'il calcule et affiche la moyenne des notes.
 - c) Modifier l'algorithme précédent pour qu'il renvoie la note maximale et la note minimale de la série des de notes.
 - d) Modifier l'algorithme pour qu'il affiche le nombre de notes en dessous de 10/20.

UA 7: PROGRAMMATION EN C 8H
UE 24 : Les langages de programmation et IDE IH

Compétence visée

- ❖ Citer quelques langages de programmation
- ❖ Citer quelques IDE
- ❖ Installer un IDE

Situation-problème

Une start-up de la place recrute des programmeurs pour le développement de ses applications. Vous postulez et lors de l'entretien d'embauche les questions suivantes vous ont été posées.

1. Définir langage de programmation, IDE
2. Citer quelques exemples de langages de programmation et d'IDE
3. Décrire les étapes d'installation d'un IDE

I. LES LANGAGES DE PROGRAMMATION

I. Définitions

La **programmation** est la codification des instructions indiquant les opérations que doit effectuer un ordinateur ou une machine électronique pour traiter des données.

Le **programme** est une suite d'instructions enregistrées dans la mémoire d'un ordinateur lui permettant d'exécuter des tâches données.

Le **langage de programmation** est un langage permettant d'écrire des algorithmes traduits en programme exécutant ces algorithmes.

2. Les types de langages de programmation

Les langages de programmation peuvent être classés en deux catégories à savoir :

- **Les langages interprétés** : un langage est dit interprété lorsqu'il est traduit au fur et mesure par un programme auxiliaire (interpréteur) pour le rendre intelligible par le processeur. Parmi les langages interprétés, nous pouvons citer entre autres : **PHP, Basic, Prolog, Perl, MATLAB** etc.
- **Les langages compilés** : un langage informatique est dit compilé lorsqu'il est traduit une fois pour toute par un programme annexe (compilateur) afin de générer un fichier exécutable. Comme exemples de langages compilés nous avons : **C, C++, COBOL, Pascal, Fortran** etc.

NB : il existe néanmoins d'autres langages de programmation qui appartiennent aux deux catégories : ce sont les langages intermédiaires.Exemples : Java, LISP

II. LES IDE

I. Définition et exemples d'IDE

Un IDE (Integrated Development Environment) est logiciel qui rassemble des outils permettant de développer d'autres logiciels. Les outils d'un IDE peuvent être :

- Un éditeur de code intelligent ;
- Un simulateur permettant de tester l'exécution de son logiciel
- Un compilateur qui va transformer le code source en code binaire
- Un débogueur aidant à la correction des bugs.

Il existe de nombreux IDE et nous pouvons citer :



Eclipse



Code ::Blocks



NetBeans



Android Studio



Dev-C++



Scilab

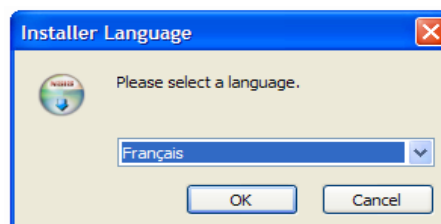
2. Installation d'un IDE

L'installation d'un IDE suit les mêmes règles que celles des autres logiciels c'est-à-dire avoir le fichier exécutable, l'ouvrir ou l'exécuter en tant qu'administrateur puis suivre la procédure jusqu'à la fin de l'installation.

Cas de Dev-C++

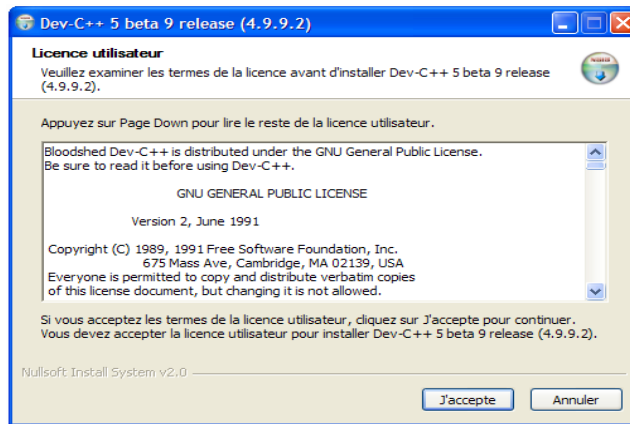
Pour installer Dev-C++, il faut :

- ✓ Double-cliquer sur son set-up ;
- ✓ choisir la langue par défaut dans lafenêtre qui apparait, puis cliquer sur "OK"

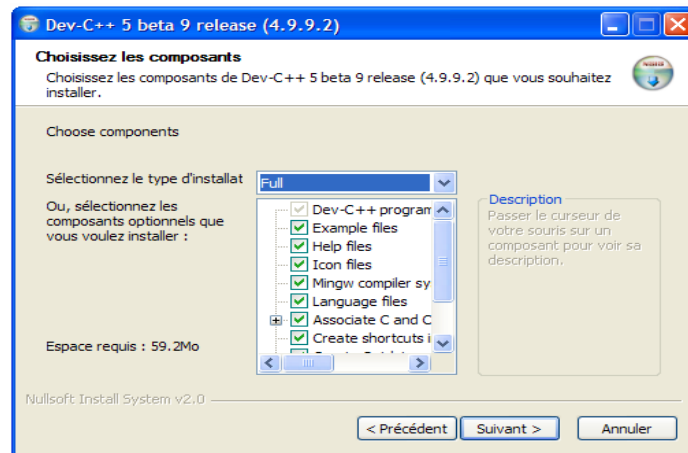


Fenêtre du choix de la langue

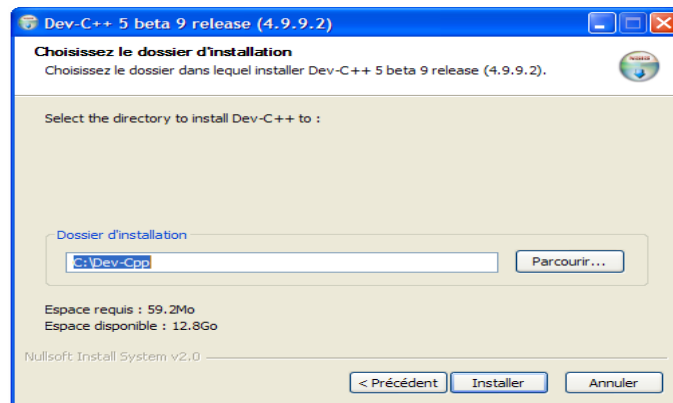
- ✓ Accepter les termes de la licencedans la nouvelle la fenêtre qui apparait en cliquant sur "J'accepte"



- ✓ Choisir le type d'installation (Full recommandé) puis cliquer sur “Suivant”

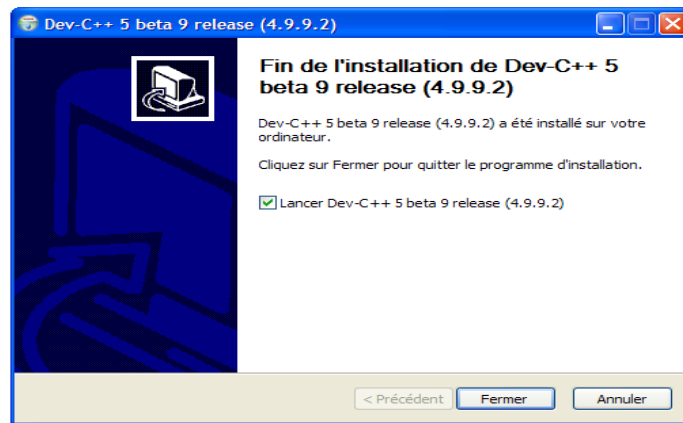


- ✓ Choisir ensuite le dossier d'installation et cliquer sur “Installer” et patienter quelques instants



Fenêtre de l'emplacement d'installation

- ✓ Cliquer sur sur “Fermer dans la fenêtre de fin d'installation ;



Bilingual game

1. Translation from English to French

Integrated Development Environment (IDE) : Environnement de Développement Intégré (EDI)

Language: langage

Programming language : Langage de programmation

2. Définir langage de programmation en anglais

Programming language is a formal language comprising a set of instructions that produce various kinds of output.

Activités d'intégration

- 1- Définir : Langage de programmation, programme, code source, IDE.
- 2- Citer les types langages de programmation et établir la différence entre ces derniers.
- 3- Enumérer quatre exemples d'IDE.
- 4- Enumérer les outils que l'on peut retrouver dans un IDE.
- 5- Pourquoi dit-on que Java est un langage intermédiaire ?

UE 25 : Ecrire un programme en C 2H

Compétences Visées :

- Structure d'un programme C
- Déclarer les variables et les constantes en C
- Utiliser les fonctions d'entrée/sorties en C
- Écrire son premier programme C

pré-requis

- Connaître le rôle des langages de programmation
- installer un IDE
- expliquer les notions de compilation et d'interprétation des programmes

Situation problème

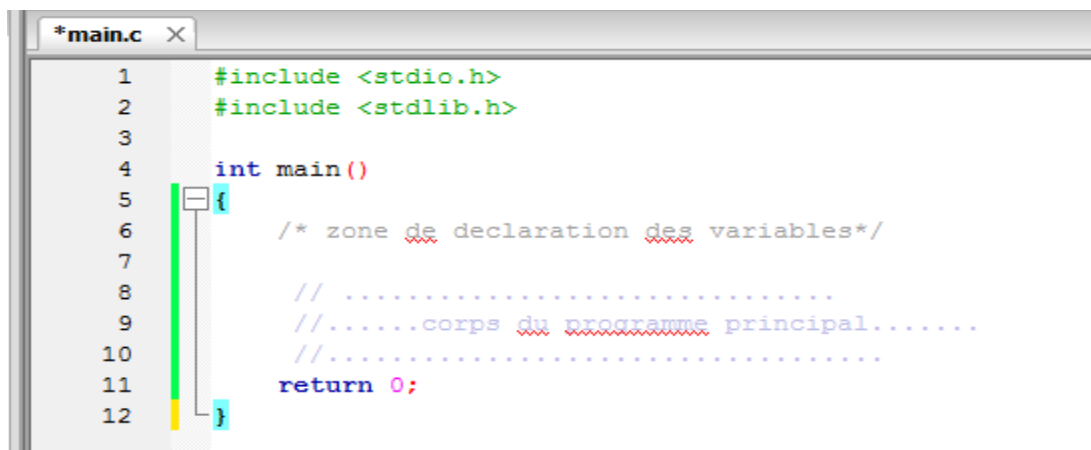
Pour son programme C, quelle syntaxe Nono Boris doit-il utiliser pour transcrire les opérateurs arithmétiques, les variables déclarées ainsi que les fonctions de lecture et d'écritures présents dans son algorithme en langage C? Quelle est la structure minimale à avoir pour commencer son programme C?

INTRODUCTION

Le langage C est un langage dont la syntaxe est légèrement complexe. Le langage C est très sensible à la casse c'est à dire qu'il fait la différence entre la majuscule et la minuscule. Chaque instruction en langage C doit se terminer par un point virgule.

I. STRUCTURE D'UN PROGRAMME C

De manière générale, un programme C consiste en la construction de blocs individuels appelés fonctions qui peuvent s'invoquer l'un l'autre. Pour pouvoir s'exécuter, un programme C doit contenir une fonction principale appelée **main** qui sera le point d'entrée de l'exécution (c'est à dire la première fonction invoquée au démarrage de l'exécution). Toutes les autres fonctions sont en fait des sous-programmes.



```
*main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      /* zone de declaration des variables*/
7
8      // .....
9      //.....corps du programme principal.....
10     //.....
11     return 0;
12 }
```


Cette structure minimale comporte les parties suivantes:

- ✓ Importation des bibliothèques. Ce sont des ensembles de fonctions déjà écrites et intégrées dans l'environnement et prêtes à être utilisées par le programmeur. (dans l'exemples ci(-dessus on a deux bibliothèques standards: **stdio.h** (permettant de gérer les fonctions d'entrées/sorties) et **stdlib.h** (pour la gestion des fichier) il existe aussi les bibliothèques **Maths.h** (pour la gestion des fonctions mathématiques) **String.h** (pour la gestion des chaines de caractères) **time.h** (pour la manipulation de la date et de l'heure)
- ✓ La fonction principale **main**. Dont le prototype est donné par `int main()`. Ici **int** représente le type de retour de la fonction c'est à dire un entier. Le mot **void** est utilisé lorsque la fonction ne retourne rien. De même lorsqu'elle ne prend aucun argument, on peut utiliser le mot void entre parenthèses du main. **Exemple:** `int main(void)`
- ✓ La zone de déclaration des variables
- ✓ le corps de la fonction principale proprement dite

II. TYPES DE DONNEES

Le langage C est un langage typé. Toute variable, est d'un type précis. Le type d'une variable définit la façon dont elle est stockée en mémoire. Les types de base les plus utilisés sont:

Les entiers : Par défaut, les entiers permettent de stocker des valeurs de signe quelconque. Si on préfixe un type entier par *unsigned*, on le restreint à des valeurs uniquement positives. Les types entiers en C sont: **int, short, long**. L'espace des valeurs est récapitulé dans le tableau ci-dessous:

| Type | Taille (Octet) | Nbre Valeur | Valeur Min | Valeur Max |
|----------------|----------------|-------------|-------------|----------------|
| short | 1 | 2^8 | -128 | +127 |
| int | 2 | 2^{16} | -32768 | +32767 |
| Long | 4 | 2^{32} | -2147483648 | +2 147 483 647 |
| unsigned short | 1 | 2^8 | 0 | 255 |
| unsigned int | 2 | 2^{16} | 0 | 65535 |
| unsigned long | 4 | 2^{32} | 0 | +4 294 967 295 |

Réels ou flottants : Les flottants servent à représenter les réels. Ils utilisent la représentation à virgule flottante. Les types réels en C sont: **float, double**.

| Type | Taille (Octet) | Valeur Min | Valeur Max |
|-------------|----------------|-------------------------|-----------------------|
| float | 4 | $-3,4 \cdot 10^{-38}$ | $3,4 \cdot 10^{38}$ |
| double | 8 | $-1,7 \cdot 10^{-308}$ | $1,7 \cdot 10^{308}$ |
| Long double | 4 | $-3,4 \cdot 10^{-4932}$ | $3,4 \cdot 10^{4932}$ |

Caractères : On utilise le mot-clé **char** pour désigner une variable de type caractère. Il est codé sur 1 octet. Il existe des caractères particuliers dont les plus usuels sont:

| Caractère | signification |
|-----------|---------------|
| \t | tabulation |
| \n | Saut de ligne |
| \b | Backspace |

Chaîne de Caractères : les chaînes de caractères sont vues comme un pointeur sur des caractères et sont donc de type `char *`. Les chaînes de caractères sont vues en C comme un tableau de caractères.

III. LES VARIABLES

Une variable est un emplacement de la mémoire dans lequel est stockée une valeur. Les variables sont définies par un type et un identificateur. L'identificateur (nom) permet d'identifier l'emplacement de la mémoire représenté par cette variable. Le type détermine la taille de la variable et donc la taille de l'emplacement mémoire réservé par la machine. Il permet également de déterminer les opérations pouvant être effectuées sur la variable. Pour utiliser une variable, il faut la déclarer. En Langage C, la déclaration de variables intervient juste après l'accolade suivant `main()`.

Déclaration des variables

La syntaxe de déclaration d'une variable en C est: `type nom_de_la_variable;`

exemple

```
int a,b ; //Déclaration de 2 variables a et b de type entier
char i ; //Déclaration d'une variable i de type caractère
double Pi = 3.14159 ;
char * chaine = "je me forme";
```

Déclaration des tableaux

Un tableau permet de déclarer une variable qui doit se présenter comme une collection de valeurs. On numérote alors ces valeurs à partir de 0. Par exemple,

```
//Déclaration d'un tableau d'entier 4 éléments :
int tab[4]; // tab est le nom de la variable tableau
//Déclaration d'un tableau de réels 7 éléments :
float V[7]; // V est le nom de la variable tableau
//Déclaration d'une chaîne de caractère :
char ch[10]; //la variable ch aura 10 caractères au max
```

Pour affecter des valeurs à un tableau, on doit à un index à la cellule du tableau correspondante tout en sachant que le premier indice du tableau est 0.

Exemple

```
tab[0]=5 ;
tab[1]=20 ;
tableauI[3]=-17 ;
```

IV. CONSTANTES

Une constante est une valeur portant un nom. En langage C les constantes en C sont non typées, on les définit dans l'entête de la source, juste en dessous des #include avec la syntaxe suivante:

```
#define Nom_de_la_variable valeur
```

Exemple

```
#define N 25 // noter l'absence du point virgule
```

Le préprocesseur va remplacer tous les noms N par la valeur 25.

V. LES OPERATEURS

Opérateurs de comparaison

| Opérateurs | symbole | Opérateurs | symbole |
|-------------------|---------|-------------------|---------|
| Affectation | = | Supérieur ou égal | >= |
| Égalité | == | Supérieur | > |
| différent de | != | Et logique | && |
| Inférieur ou égal | =< | Ou logique | |

Opérateurs arithmétique

| Opérateurs | Noms | Opérateurs | Noms |
|------------|----------------|------------|----------------|
| + | addition | ^ | Puissance |
| - | Soustraction | % | Modulo |
| * | Multiplication | ++ | Incréméntation |
| / | Division | -- | Décréméntation |

On a aussi les opérateurs complexes : += ; -= ; *= ; /=

```
Var a=8 ;
```

a+=2 équivaut a=a+2 a contient 10

a-=2 équivaut a=a-2 a contient 6

a*=2 équivaut a=a*2 a contient 16

a/=2 équivaut a=a/2 a contient 4

VI. ECRITURE ET LECTURE DES DONNEES

➤ L'écriture permet d'afficher les données à l'écran. Pour afficher un message à la console, on utilise la fonction **printf** : la syntaxe est la suivante: **printf(format,param_1,param_2, ...,param_n);**

Le paramètre format est une chaîne de caractères entre guillemets qui contient les caractères à afficher tels quels et des codes de format qui indiquent le type de l'information à afficher. Ces codes commencent par un %.

Les paramètres param_1 à param_n entre virgules sont les informations qui seront affichées à la place des codes de formats.

| Spécification du code format | | | |
|------------------------------|-------------------------|------|-------------------------------|
| %d | Un entier | %s | Une chaîne de caractères |
| %f g G | Un réel double ou float | %e,E | Réel en notation scientifique |
| %c | Un caractère | | |

exemple1: afficher le message "bonjour, je programme en C".

```
printf("bonjour , je programme en C");
```

exemple2: afficher la valeur de la variable entière X "

```
printf("la valeur de x est: %d", X);
```

➤ La lecture permet de récupérer une valeur saisie par un utilisateur. L'instruction en Langage C pour lire sur la console est **scanf** la syntaxe est la suivante:

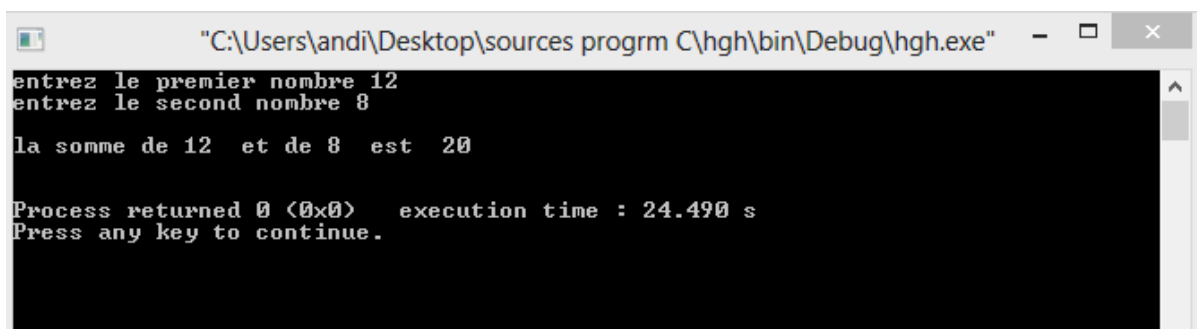
```
int scanf(format, &donnee_1, &donnee_2, ..., & donnee_n);
```

Cette fonction retourne le nombre de données correctement lues. Les & représentent les adresses des variables auxquelles il faut affecter les données lue au clavier.

Exemple : scanf("%d", &x)

Exercice d'application: écrire son premier programme C

Écrire un programme C qui demande à l'utilisateur deux nombres puis calcule et affiche la somme. Le résultat doit ce présenter comme la capture ci-dessous.



```
"C:\Users\andi\Desktop\sources programm C\hgh\bin\Debug\hgh.exe"
entrez le premier nombre 12
entrez le second nombre 8
la somme de 12 et de 8 est 20
Process returned 0 (0x0) execution time : 24.490 s
Press any key to continue.
```

résolution

```
#include <stdio.h>
int main()
{
    int a,b,S;
```

```
    printf("entrez le premier nombre ");
    scanf("%d",&a);
    printf("entrez le second nombre ");
    scanf("%d",&b);
```

```
S=a+b;
printf("\nla somme de %d et de %d est
%d",a,b,S);
```

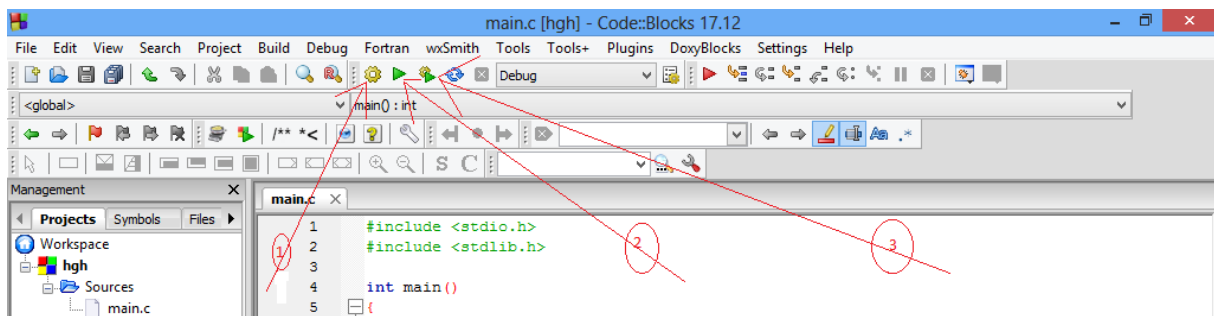
```
return 0;
}
```

Le programme C que nous avons écrit doit être compilé et exécuté en cliquant sur l'un des boutons ci-dessous:

bouton 1: pour compiler

bouton2: pour exécuter

bouton 3: compile puis lance automatiquement l'exécution



CONCLUSION

Toute variable C est typée, de même que toute valeur de retour d'une fonction. Mais il peut arriver qu'aucune valeur ne soit disponible. Pour exprimer l'idée de "aucune valeur", on utilise le mot-clé **void**. Il est commode pour faciliter la compréhension de son programme par d'autres personnes d'y insérer un texte explicatif qui dit ce que fait le programme en un endroit bien précis et qui n'est pas pris en compte pendant l'exécution du programme.

EXERCICES

- 1) Donnez la structure minimale d'un programme C
- 2) Donnez la syntaxe de déclaration d'une variable en C
- 3) Donnez la syntaxe de déclaration d'une constante en C
- 4) Donnez le rôle de l'importation des bibliothèques au début de tout programme C

UE 26 : STRUCTURES DE CONTROLE EN LANGAGE C IH

COMPETENCES VISEES :

A la fin de cette leçon, chaque élève de la classe de Tle C/D sera capable d'écrire les structures de contrôle en C.

SITUATION PROBLEME :

Votre petit frère de la classe de 1^{ère} D veut écrire un petit programme en langage C. Cependant, il ne connaît pas comment écrire la syntaxe de la structure de contrôle qu'il a choisi en langage C. Sachant que vous avez fait un cours en 1^{ère} sur le C, il sollicite votre aide. En vous basant sur la documentation et le cours vu en 1^{ère} C/D l'an passé sur le C, montrez lui comment écrire les structures de contrôle en langage C.

INTRODUCTION

Nous avons au quotidien des problèmes qui nous poussent à effectuer des choix. Pour résoudre ces problèmes en algorithmique ou en langage C, on utilise les structures de contrôle. Dans cette leçon, nous allons nous intéresser aux structures de contrôle en langage C.

I) Les structures alternatives (if...else)

Encore appelé les conditions, elles permettent de tester les variables ou des expressions. Nous avons plusieurs types que l'on peut citer :

I) L'instruction if (Si (expression) Alors)

La syntaxe est la suivante :

| | | |
|------------------------------------------|----|-------------------------------------------------|
| <i>if(expression)</i> | ou | <i>if(expression) {</i> |
| <i>instruction ; // dans le cas</i> | | <i>instructions ; // s'il y en a plusieurs.</i> |
| <i>// d'une instruction ou</i> | | <i>}</i> |
| <i>// des instructions sur une ligne</i> | | |

Elle est traduite comme suit :

- D'abord « expression » est évaluée.
- Si elle est vraie, on exécute « instruction ».
- Sinon on saute « instruction » pour exécuter la suite du programme.

Exemple : écrire un programme en C qui teste la variable âge d'un utilisateur en lui disant s'il est majeur. (Âge supérieur ou égale à 18 ans).

```
if (age >= 18)
{
    printf ("Vous êtes majeur !");
}
```

2) L'instruction if else (Si (expression) alors ... sinon ...)

L'instruction if/else, signifie que si « expression » est vraie, on exécute alors « instruction1 » sinon (else), dans le cas contraire, on exécute « instruction2 ». À noter que « instruction » peut être formée aussi d'un bloc { } ou d'autres combinaisons de if/else.

Sa syntaxe est la suivante :

| | | | |
|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>if (expression) instruction1; else instruction2;</pre> | <p><i>Ou :</i></p> <pre>if (expression) { instruction1; else instruction2; }</pre> | <pre>if (expression1) instruction1 else if (expression2) instruction2 else instruction3</pre> | <p><i>Ou :</i></p> <pre>if (expression1) { instruction1 } else if (expression2) { instruction2 } else { instruction3 }</pre> |
|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|

Exemple : écrire un programme en langage C qui permet de résoudre une équation du second degré.

Solution :

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    float delta, a, b, c;
    printf("\n Entrer les valeurs de a, b et c de l'équation");
    scanf("%f %f %f ", &a,&b,&c);
    delta =b*b - 4*a*c;
    if (delta > 0) {
        printf("\n Il existe deux solutions dans R, x1 et x2");
    } else
        if (delta < 0) {
            printf("\n L'équation n'a pas de solution dans R");
        } else {
            printf("\n Il existe une solution double x1=x2=x0 ");
        }
    return 0;
}
```

II) Les structures itératives

I) La boucle Pour (for (...))

La syntaxe est la suivante :

```
int compteur;
for (compteur = 0; compteur < 10; compteur++)
{
```

```

// bloc d'instruction ;
}

```

- **La première est l'initialisation** : cette première instruction est utilisée pour préparer notre variable compteur. Dans notre cas, on initialise la variable à 0.
- **La seconde est la condition** : comme pour la boucle « *while* », c'est la condition qui dit si la boucle doit être répétée ou non. Tant que la condition est vraie, la boucle « *for* » continue.
- **Enfin, il y a l'incréméntation** : cette dernière instruction est exécutée à la fin de chaque tour de boucle pour mettre à jour la variable compteur. La quasi-totalité du temps on fera une incréméntation, mais on peut aussi faire une décrémentation (variable--) ou encore n'importe quelle autre opération (variable += 2 ; pour avancer de 2 en 2 par exemple).

Exemple : écrire un programme en C qui permet de dire bonjour les amis 10 fois.

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    int i;
    for (i = 0; i < 10; i++)
        printf("\n Bonjour les amis ");
    return 0;
}

```

2) La boucle tanque (while ())

La syntaxe est la suivante :

```

while (/* Condition */)
{
    // Instructions à répéter
}

```

Exemple : réécris le programme précédent en utilisant cette fois ci la boucle while.

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    int i = 0;
    while (i < 10)
        printf("\n Bonjour les amis ");
        i++;
    return 0;
}

```

3) La boucle répéter jusqu'à (do...while ())

La syntaxe est la suivante :


```

Do
{
    // bloc d'instruction ;
} while (condition) ;

```

Cette boucle s'exécutera toujours au moins une fois. En effet, le test se fait à la fin comme vous pouvez le voir.

Exemple : réécris le programme précédent en utilisant cette fois-ci la boucle « do while »

```

#include <stdio.h>                printf("\n Bonjour les amis ");
#include <stdlib.h>                i++;
int main(int argc, char *argv[]) } while (i < 10)
{
int i = 0;                        return 0;
do{                                }

```

CONCLUSION

Dans cette leçon nous avons remarqué que les structures de contrôle sont à la base de tous les problèmes. Nous avons aussi vu que les boucles permettent de répéter une série d'instructions plusieurs fois. Parmi ces boucles certains sont plus adaptées que d'autres selon le cas.

JEU BILINGUE

| Mots en anglais | Signification en français |
|-----------------|---------------------------|
| if | Si |
| else | Sinon |
| while | Tant que |
| Do while | Faire tant que |
| For | Pour |
| Print | Imprimer |

EXERCICES

Exercice 1 :

Ecrire un programme en langage C qui permet de donner la nature de l'eau en fonction de sa température. Sachant que (pour $t \leq 0$ Eau glacée ; $0 < t < 100$: Eau liquide ; $t > 100$: Eau gazeuse ;)

Exercice 2 :

Soit à traduire en langage C les algorithmes ci-dessous :

| | | |
|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Pour i allant de I à nbre faire</p> <p>som ← som + i ;</p> <p>écrire("la somme est : ", som) ;</p> <p>Finpour</p> | <p><u>Algorithme</u> divisio_entiere</p> <p><u>Var</u> p, q, i: entier ;</p> <p><u>Début</u></p> <p>Ecrire ('entrer un nombre') ;</p> <p>Lire (p) ;</p> <p>Ecrire ('entrer un autre nombre') ;</p> <p>Lire (q) ;</p> <p>i ← 0 ;</p> <p><u>Tant que</u> p ≥ q <u>faire</u></p> <p>p ← p - q ;</p> <p>i ← i + I ;</p> <p><u>fin</u>tantque ;</p> <p>Écrire ('le résultat est : 'i) ;</p> <p><u>Fin</u> algorithme</p> | <p><u>variables</u> N, i, somme : entiers</p> <p><u>début</u></p> <p>écrire("donner la valeur de N") ;</p> <p>lire(N) ;</p> <p>i ← 0 ;</p> <p>somme ← 0</p> <p><u>Répéter</u></p> <p>somme ← somme + i ;</p> <p>i ← i + I ;</p> <p><u>Jusqu'à</u> i ≥ N</p> <p>écrire("la somme est : ",somme) ;</p> <p><u>fin</u></p> |
|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

UE 27 : UTILISER LES TABLEAUX EN C IH

Compétences visées :

- Définir et déclarer un tableau en C
- Parcourir et accéder à un élément précis du tableau en C.

Situation problème :

Lors d'une visite dans le bureau de votre père, vous vous rendez compte qu'il est en désordre, il vous demande de l'aider à mettre de l'ordre et rendre accessible tout ses dossiers

- Que proposez vous a votre père pour mettre de l'ordre ? **Réponse** : Ranger
- Quel matériel utiliser pour ranger les dossiers ? **Réponse** : placard, carton, tableau ...
- Après avoir rangé les dossiers, il vous demande de chercher un dossier bien précis parmi tant d'autres, comment vous y prendre ? **Réponse** : parcourir et comparer le dossier en question aux autres.

I- DECLARATION D'UN TABLEAU EN C

a- Définition et rappel

Le tableau est une collection homogène de données ordonnées et de taille statique ; il fut un ensemble D'octets permettant de représenter une liste d'éléments de même type. Chaque élément du tableau est repère par un indice (son rang dans le tableau).

Exemple : soit un tableau T

| | | | | | |
|----|-----|-----|-----|----|----|
| 45 | 555 | 478 | 400 | 48 | 78 |
|----|-----|-----|-----|----|----|

Ce tableau est un tableau des éléments de même type "entier" de taille 6 et dont le premier indice est 0 jusqu'à 5. 478 est l'élément d'indice 2 et 48 est l'élément d'indice 4.

b- Déclaration en C

Pour la déclaration d'un tableau en C, nous avons deux manières : **la déclaration brute** (c'est-à-dire sans initialisation) et **la déclaration avec initialisation**

- Déclaration brute

Pour déclarer un tableau de façon brute en C la syntaxe est la suivante : **type Identificateur [taille du tableau]**

La taille du tableau est le nombre de ses éléments, elle ne peut être une variable, elle doit être une constante définie avant ou lors d la déclaration.

Exemple : pour déclarer un tableau de 75 entiers, la syntaxe est la suivante : **Int tab [75]**

Ci est un tableau **d'entier** dont la taille est **75** et se nomme **tab**

- Déclaration avec initialisation

La déclaration avec initialisation se fait presque de la même façon celle précédente, mais a ce niveau nous devons préciser tous les éléments du tableau a la déclaration entre guillemet ; la syntaxe est la suivante : **type identificateur [taille]={val1, val2, ..., valN}**

Exemple : float manu [6]={0, 6.5, 9, 34, 45, 23}, ceci est un tableau dénommé manu de 5 réels initialisé a 0, 6.5, 9, 34, 45 et 23

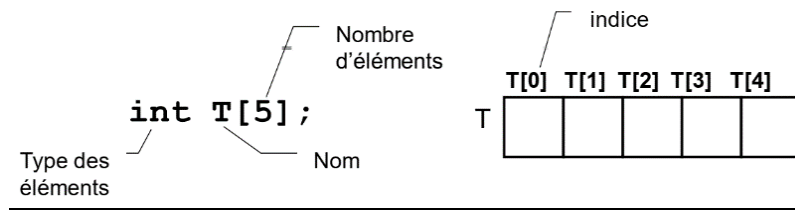


Figure 3: Déclaration tableau en C

II- Accéder à un élément du tableau en C

Après déclaration du tableau, l'on peut se trouver dans le besoin d'écrire ou lire une valeur à une position précise du tableau ou de parcourir pour afficher les éléments du tableau.

a- Accéder à un élément précis du tableau en C

Syntaxe: <NomTableau> [<indice>]

Exemple 1: Pour un tableau T de taille N:

T[0] pour accéder au premier élément et T[N-1] pour accéder au dernier élément

Exemple 2 :

En considérant le tableau déclaré dans la remarque, nous avons :

Tab[0] → retourne 1 et Tab[3] → retourne 4

| | Tab[0] | Tab[1] | Tab[2] | Tab[3] | Tab[4] |
|-----|--------|--------|--------|--------|--------|
| Tab | 1 | 10 | 9 | 4 | 5 |

b- Affectation dans un tableau

Pour écrire a une position du tableau, on procède suivant la syntaxe suivante :

Syntaxe : <NomTableau> [<indice>] = valeur ;

Exemple :

```
int T[4]; //Déclaration d'un tableau de taille 4
T[0] = 23; //Écrire 23 dans la première case du tableau
T[1] = 3; //Écrire 3 dans la deuxième case du tableau
T[2] = 0; //Écrire 0 dans la troisième case du tableau
T[3] = 4; //Écrire 4 dans la quatrième case du tableau
```

c- Parcourir un tableau en C

Les éléments d'un tableau sont à lire et a afficher élément par élément, pour parcourir donc un tableau, nous avons besoin d'utiliser l'instruction for () {...} et l'instruction d'affichage printf ().

Exemple : soit le tableau manu défini ci-dessus, le programme en C permettant d'afficher ces éléments est :

```

#include <stdio.h>
void main()
{
    float manu[6]={ 0, 6.5, 9, 34, 45, 23};
    int i;
    for (i=0 ;i<5 ; i++) /*parcours le tableaux élément par élément*/
    {
        printf("%.1f \n", manu[i]);
    }
}

```

Jeux bilingues : donner la signification des termes suivants en Anglais : *tableau, entier et nombre décimal*, pour () faire, si et sinon

Activiter d'intégration :

- 1- Définir tableau
- 2- définir en algorithmique un tableau d'entier de 10 éléments
- 3- traduire cet algorithme en langage C
- 4- créer un tableau de d'entier regorgeant les 12 premiers nombres entiers
- 5- Écrire un code en C, qui permet d'afficher l'élément à l'indice 6 du tableau en C

UE 28 : Traduire un algorithme de recherche séquentielle en C 2H

COMPETENCES VISEES :

A la fin de cette leçon, chaque élève de la classe de Tle C/D sera capable de traduire un algorithme de recherche séquentielle en langage C.

SITUATION PROBLEME :

Votre petit frère de la classe de I^{ère} D a écrit un algorithme qui crée un tableau et y stocke de manière aléatoire des nombres, puis, demande un nombre à un utilisateur et recherche si le nombre est dans le tableau. Cependant, il n'arrive pas à traduire cet algorithme en langage C et sollicite donc votre aide. En vous basant sur la documentation et le cours vu sur les structures de contrôle en langage C, aidez votre frère à traduire son algorithme de recherche en C.

INTRODUCTION

La recherche est une opération qui consiste à fouiller un élément dans un ensemble. Exemples : fouiller un stylo noir dans une trousse, un objet dans un classeur... pour effectuer cette opération, généralement nous l'effectuons de manière séquentielle (objet après objet jusqu'au dernier). Dans cette leçon nous allons traduire l'algorithme de recherche séquentielle en langage C.

Traduire un algorithme de recherche en langage C

I) Algorithme de recherche

Objectif : Rechercher une information dans un tableau

- Méthode : séquentielle
- Soit T un tableau de N éléments et val l'élément cherché
- Parcours du tableau à partir du premier élément (T[0])
- Arrêt quand élément trouvé ou si fin de tableau (T[n-1])

Algorithme recherche_sequentielle

```
{Recherche le premier indice où se trouve la valeur val parmi les N données du tableau tab ; affiche l'indice si la valeur est trouvée.}
```

```
Variables T [0, N-1], indice, val, n : entier
```

```
Début
```

```
    indice ← 0
```

```
    tant que ( val != T[indice] ET indice < N-1) faire  
        indice ← indice + 1
```

```
    ftq
```

```
    si T[indice] = val alors  
        écrire(" L'élément se trouve en : ", indice);
```

```
    sinon
```

```
        écrire(" Elément non présent ");
```

```
    fsi
```

Fin

2) Traduction de l'algorithme de recherche en langage C

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int indice,val,n, i, tab[50];
    /*où
    indice : indice courant,
    val: valeur à rechercher,
    n : dimension du tableau,
    tab[50] : tableau donnée. */

    /* Saisie des données */
    printf(" Entrer la dimension du tableau (max 50): ");
    scanf("%d",&n);
    for(i=0; i<n; i++){
        printf("Elément %d : ", i);
        scanf("%d",&tab[i]);
    }
    printf(" Elément à rechercher : ");
    scanf("%d",&val);
    while(val!=tab[indice] && indice < n-1)
    {
        indice=indice+1;
    }
    if(tab[indice]==val)
    {
        printf("L'élément se trouve à la position : %d",indice);
    }else{
        printf("L'élément %d recherché ne se trouve pas dans le tableau ",val);
    }
    return 0;
}
```

CONCLUSION

Dans la recherche séquentielle, on parcourt tout le tableau jusqu'à la fin avant d'en décider pour une donnée si elle y est ou non. Cette opération est lourde si on a un grand tableau. C'est pourquoi il existe d'autres algorithmes tels que l'algorithme de recherche dichotomique (elle s'applique sur un tableau déjà trié) dont nous vous laissons l'occasion d'en lire plus à propos sur internet.

JEU BILINGUE

| Mots en français | Signification en anglais |
|-------------------|--------------------------|
| Recherche | Search or research |
| Séquentiel | Sequential |
| Fouille de donnée | Data Mining |
| Fouiller | Search |

EXERCICES

Exercice I :

Soit l'algorithme de recherche ci-dessus : traduire en langage C cet algorithme en remplaçant la boucle tant que (While) par la boucle (Do While).

Exercice 2 :

En supposant que le tableau est déjà trié :

Donner le principe pour rechercher un élément dans ce nouveau tableau.

Exercice 3 :

Peut-on écrire l'algorithme de recherche séquentiel ci-dessus à l'aide de la boucle Pour ? si oui traduire cet algorithme en langage C en utilisant la boucle Pour (For).

UE 29 : Identifier/interpréter/corriger les erreurs de programmation IH

Compétences visées :

- ✓ Identifier les erreurs de programmation
- ✓ interpréter et corriger les erreurs de programmation

Situation problème :

Nono Boris a écrit deux programmes C qui ont des comportements différents. Le premier affiche de nombreuses erreurs après compilation le second n'affiche aucune erreur mais les résultats affichés ne sont pas corrects. Comment identifier et corriger le problème? Quel outil utilisé pour faciliter cette tâche?

INTRODUCTION

La programmation est une démarche très complexe, et comme c'est le cas dans toute activité humaine, on y commet de nombreuses erreurs. Pour des raisons anecdotiques, les erreurs de programmation s'appellent des « bugs » (ou « bogues », en France), et l'ensemble des techniques que l'on met en œuvre pour les détecter et les corriger s'appelle « debug » (ou « débogage »). L'outil logiciel utilisé pour ce travail est le **débogueur**. En fait, il peut exister dans un programme trois types d'erreurs assez différentes, que nous allons analyser dans ce cours.

I. ERREURS DE SYNTAXE

Le compilateur C ne peut exécuter un programme que si sa syntaxe est parfaitement correcte. Dans le cas contraire, le processus s'arrête et vous obtenez un message d'erreur. Le terme syntaxe se réfère aux règles que les auteurs du langage ont établies pour la structure du programme. Tout langage comporte sa syntaxe. En C par exemple toutes instruction doit se terminer par un point virgule.

Exemple dans l'exemple ci-dessous, le débogueur affiche une erreur après compilation.

```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a,b,S;
7
8
9      printf("entrez le premier nombre ");
10     scanf("%d",&a)
11     printf("entrez le second nombre ");
12     scanf("%d",&b);
13     S=a+b;
14     printf("\nla somme de %d et de %d est %d\n\n",a,b,S);
15
16     return 0;
17 }

```

Logs & others

| File | Line | Message |
|-------------------|------|---------------------------------------------------------------------------|
| C:\Users\andi\... | | === Build: Debug in hgh (compiler: GNU GCC Compiler) === |
| C:\Users\andi\... | | In function 'main': |
| C:\Users\andi\... | 11 | error: expected ';' before 'printf' |
| | | === Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) === |

Identification de la source du problème interprétation et correction

En observant bien le message renvoyé par le débogueur, une erreur est signalée à la ligne 11. mentionnant l'omission d'un point virgule avant l'instruction printf.

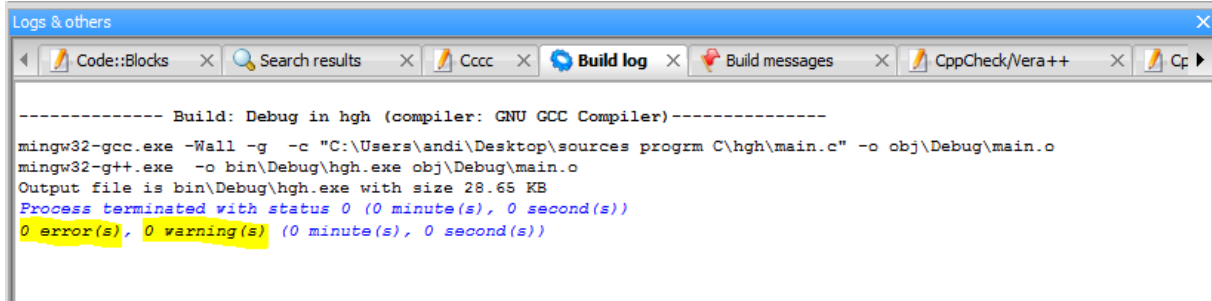
Logs & others

| File | Line | Message |
|-------------------|------|---------------------------------------------------------------------------|
| C:\Users\andi\... | | === Build: Debug in hgh (compiler: GNU GCC Compiler) === |
| C:\Users\andi\... | | In function 'main': |
| C:\Users\andi\... | 11 | error: expected ';' before 'printf' |
| | | === Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) === |

Avant l'instruction printf c'est l'instruction scanf. En fait le compilateur a eu du mal à identifier la fin de l'instruction commencé à la ligne I0 du programme. On constate bien l'absence du point virgule à la ligne I0. Le problème est résolu en ajoutant le point virgule à la fin de l'instruction puis en enregistrant le fichier ou le projet.

```
10 | scanf("%d",&a);
```

Après compilation, le débogueur ne mentionne plus d'erreur et le programme fonctionne normalement.



II. ERREURS SÉMANTIQUES

Le second type d'erreur est l'erreur sémantique ou erreur de logique. S'il existe une erreur de ce type dans un de vos programmes, celui-ci s'exécute parfaitement, en ce sens que vous n'obtenez aucun message d'erreur (rendant ainsi leur détection difficile), mais le résultat n'est pas celui que vous attendiez : vous obtenez autre chose. En réalité, le programme fait exactement ce que vous lui avez dit de faire. Le problème est que ce que vous lui avez dit de faire ne correspond pas à ce que vous vouliez qu'il fasse. La séquence d'instructions de votre programme ne correspond pas à l'objectif poursuivi. La sémantique (la logique) est incorrecte. La différence entre le texte correct et le texte erroné est souvent seulement d'un seul caractère. La découverte de telles erreurs ne peut donc se faire que par un examen très attentif du code source du programme. Ci-dessous nous donnons quelques erreurs de logiques assez fréquentes.

2.1) erreur sur une comparaison

| | |
|-----------------------------|--------------------------|
| Ce que Vous voulez | Comparer a et b |
| Ce que vous avez écrit | <code>if(a= b)</code> |
| Ce que vous avez obtenu | Une affectation de b à a |
| Ce que vous aurez dû écrire | <code>if(a= = b)</code> |

2.2) erreur avec le if

| | |
|-----------------------------|-----------------------------------------------------|
| Ce que Vous voulez | Affectation si le Test <code>a>b</code> est vrai |
| Ce que vous avez écrit | <code>if(a>b);</code> <code>a=b;</code> |
| Ce que vous avez obtenu | Une affectation de b à a dans tout les cas. |
| Ce que vous aurez dû écrire | <code>if(a>b)</code> <code>a=b;</code> |

2.3) erreur avec les priorités des opérateurs

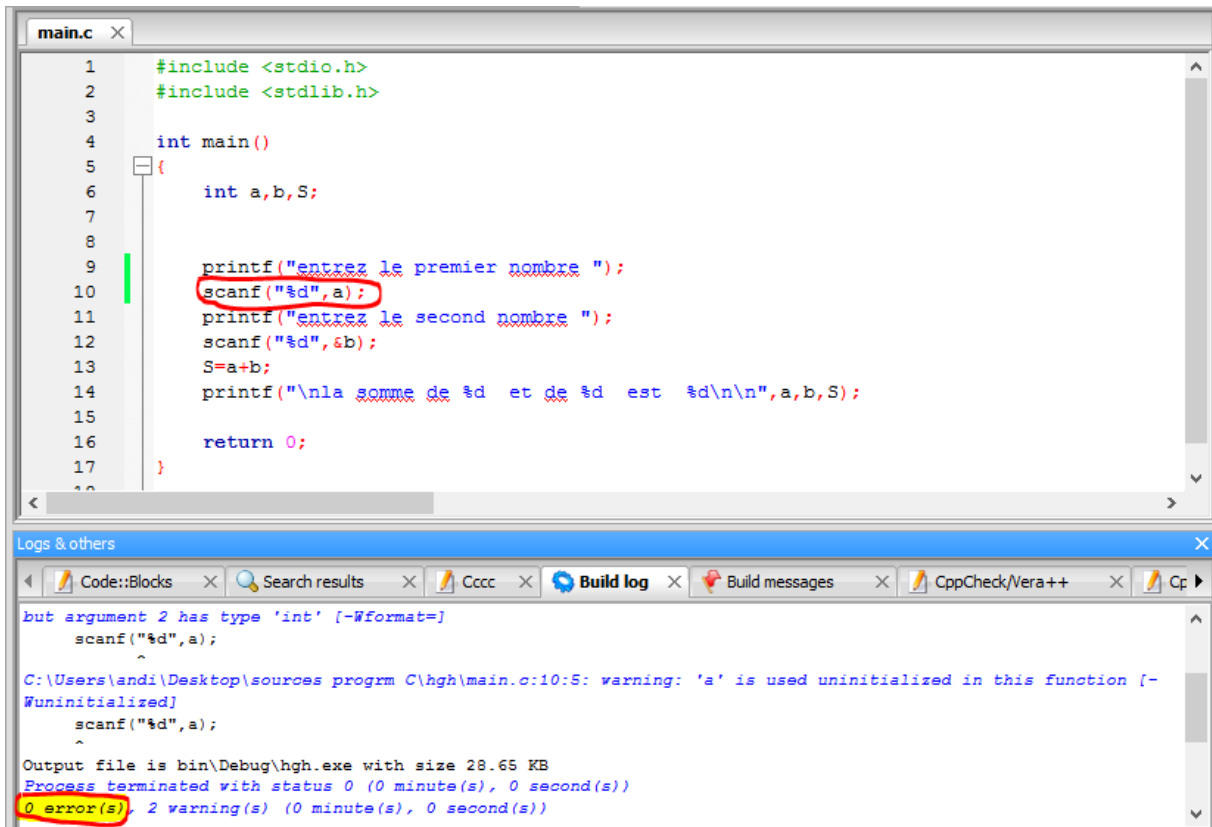
| | |
|-----------------------------|----------------------------------------|
| Ce que Vous voulez | Moyenne arithmétique de deux nombres |
| Ce que vous avez écrit | <code>a+b/2</code> |
| Ce que vous avez obtenu | Seule la valeur de b est divisée par 2 |
| Ce que vous aurez dû écrire | <code>(a+b)/2</code> |

Il existe une infinité de situations pouvant emmener un programme à bien s'exécuter mais toutes fois en renvoyant des résultats inattendus et indésirables.

III. ERREURS À L'EXÉCUTION

Le troisième type d'erreur est l'erreur en cours d'exécution (Run-time error), qui apparaît seulement lorsque votre programme fonctionne déjà, mais que des circonstances particulières se présentent (par exemple, votre programme essaie de lire un fichier qui n'existe plus). Ces erreurs sont également appelées des exceptions, parce qu'elles indiquent généralement que quelque chose d'exceptionnel s'est produit (et qui n'avait pas été prévu).

Exemple: Dans l'exemple ci-dessous, le programme compile normalement et n'affiche aucune erreur de programmation. l'exécution du programme doit bien commencer mais sera interrompu quelque instants après.



```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a,b,S;
7
8
9      printf("entrez le premier nombre ");
10     scanf("%d",a);
11     printf("entrez le second nombre ");
12     scanf("%d",&b);
13     S=a+b;
14     printf("\nla somme de %d et de %d est %d\n\n",a,b,S);
15
16     return 0;
17 }
18
```

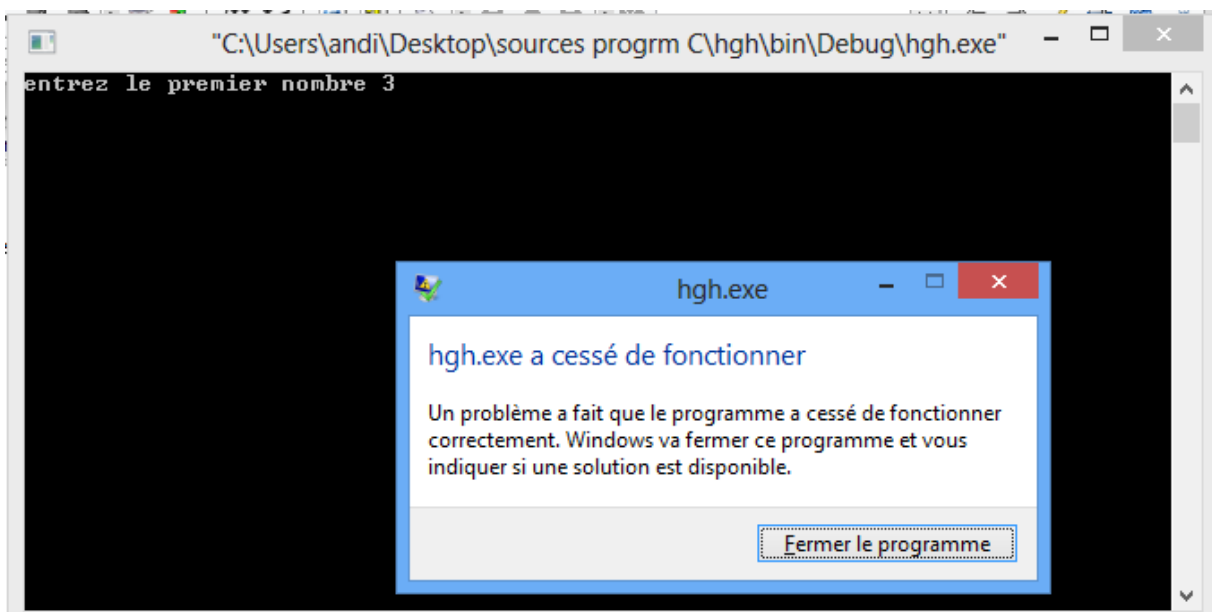
Logs & others

Code::Blocks x Search results x Cccc x Build log x Build messages x CppCheck/Vera++ x Cp

but argument 2 has type 'int' [-Wformat=]
scanf("%d",a);
^

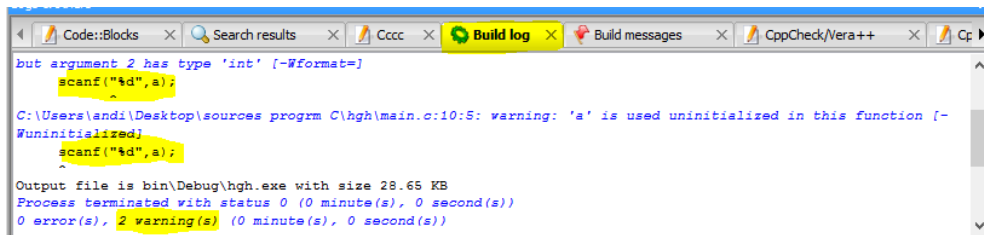
C:\Users\andi\Desktop\sources progrm C\hgh\main.c:10:5: warning: 'a' is used uninitialized in this function [-Wuninitialized]
scanf("%d",a);
^

Output file is bin\Debug\hgh.exe with size 28.65 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 2 warning(s) (0 minute(s), 0 second(s))



Identification de la source du problème interprétation et correction

En observant bien le message renvoyé par le débogueur, deux avertissements sont signalés au niveau de la fonction de lecture scanf.

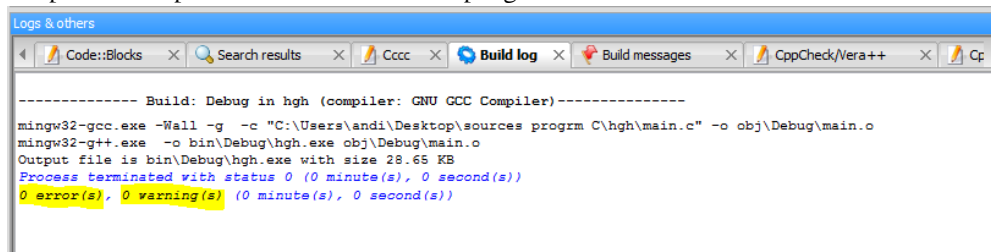


```
but argument 2 has type 'int' [-Wformat=]
scanf("%d", a);
C:\Users\andi\Desktop\sources\progmm C\hgh\main.c:10:5: warning: 'a' is used uninitialized in this function [-Wuninitialized]
scanf("%d", a);
Output file is bin\Debug\hgh.exe with size 28.65 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 2 warning(s) (0 minute(s), 0 second(s))
```

On peut donc remarquer qu'il se pose un problème sur la variable a. En fait l'adresse de la variable a (&a) n'est pas mentionné dans la fonction scanf; Le programme après la lecture de la valeur entrée par l'utilisateur n'a aucune référence pour stocker cette valeur ce qui explique l'arrêt du programme en cours d'exécution. la ligne 10 du programme devrait être donc:

```
10 scanf("%d", &a);
```

et après compilation le problème est résolu et le programme fonctionne normalement.



```
----- Build: Debug in hgh (compiler: GNU GCC Compiler)-----
mingw32-gcc.exe -Wall -g -c "C:\Users\andi\Desktop\sources\progmm C\hgh\main.c" -o obj\Debug\main.o
mingw32-g++.exe -o bin\Debug\hgh.exe obj\Debug\main.o
Output file is bin\Debug\hgh.exe with size 28.65 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

CONCLUSION

L'écriture d'un programme passe très généralement par la présence des erreurs de programmation. Leur identification, interprétation et correction est une activité assez importante pour la validité du programme. La caractéristique de beaucoup de ces erreurs s'est de ne pas provoquer de message d'erreur du compilateur, rendant ainsi leur détection difficile. La différence entre le texte correct et le texte erroné est souvent seulement d'un seul caractère. La découverte de telles erreurs se fait par un examen très attentif de la source du programme.

jeu bilingue

bogue==>bug

erreur de syntaxe ==> syntax error

débogage==>debug

erreur à l'exécution ==> run time error

programme==>program

EXERCICE1

1) Définir les termes: bug, débogage,

2) Citez trois types d'erreurs que l'on peut rencontrer dans un programme et donner une brève explication.

EXERCICE2

Le programme Ci-dessous veut permuter le contenu des variables a et b après lecture des valeurs au clavier.

```

main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a,b;
7
8      PRINTF("entrez le premier nombre ")
9      SCANF("%d",&a);
10     PRINTF(entrez le second nombre );
11     SCANF("%d",&b);
12     a=b;
13     b=a;
14     return 0;
15
16
17

```

- 1) Identifier les 07 erreurs syntaxiques présents dans ce code et apporter une correction.
- 2) Après avoir corrigé les 07 erreurs mentionnées à la question I, le programme ne permute pas les valeurs de a et de b. Identifier et corriger l'erreur de logique qui se cache dans ce code.

solution

1) identifier et corriger les erreurs

| | | |
|----------------------|---------------------------------------------------------------------------|-------------------|
| ligne 8 | Absence du point virgule. PRINTF doit être écrit en minuscule | 02 erreurs |
| ligne 9 | SCANF doit être écrit en minuscule | 01 erreur |
| ligne 10 | PRINTF doit être écrit en minuscule Absence des cotes dans printf. " " | 02 erreurs |
| ligne 11 | SCANF doit être écrit en minuscule | 01 erreur |
| ligne 16 | Absence de l'accolade fermante du main. } | 01 erreur |
| TOTAL ERREURS | | 07 erreurs |

2) identifier et corriger l'erreur sémantique

Le programme ne permute pas les valeurs car la valeur de a est écrasée et perdue lors de la première affectation.

IL faut donc déclarer une variable auxiliaire et conserver la valeur de a dans cette nouvelle variable auxiliaire puis à la fin affecter cette valeur auxiliaire à b.

On aura donc:

```

main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a,b, c;
7
8      printf("entrez le premier nombre ");|
9      scanf("%d",&a);
10     printf("entrez le second nombre ");
11     scanf("%d",&b);
12     c=a;
13     a=b;
14     b=c;
15     return 0;
16
17

```