

**Classe : Troisième**  
**Module 1 : Architecture, maintenance et tableur**  
**Chapitre 1: Décrire les périphériques**  
**Leçon 1&2: décrire les périphériques**

**Compétence :**

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

- ✓ Décrire les périphériques d'entrée/sortie
- ✓ Décrire les périphériques de communication
- ✓ Interconnecter les éléments de base d'un ordinateur
- ✓ Décrire les types de mémoire

**Situation-problème**

Votre voisin vient d'acheter son ordinateur; il désire se connecter sur internet mais il ne sait pas comment faire. Peux-tu l'aider?

**Introduction**

Un périphérique est un appareil électronique pouvant être connecté à l'ordinateur et ayant un ou plusieurs fonctions. Les périphériques d'entrées permettent de transmettre les informations à l'unité centrale, les périphériques de sortie permettent à l'unité centrale de transmettre les informations aux utilisateurs, les périphériques mixtes sont à la fois les périphériques d'entrée et de sortie; les périphériques de communication permettent la mise en réseaux d'un ensemble d'ordinateurs

I. Les périphériques d'entrée/sortie

I. L'écran

C'est un appareil qui dispose d'une zone le permettant d'afficher les images; c'est grâce à elle que l'unité centrale communique avec l'utilisateur à travers les images et les écritures. On connecte généralement l'écran à l'UC sur le port VGA

II. Le clavier

Le clavier est une base pièce électronique qui dispose un ou ensemble de boutons sur lesquels sont inscrits un ou plusieurs symboles (lettre de l'alphabet, les chiffres, ...). On connecte généralement le clavier à l'UC sur le port clavier

III. Le haut parleur

Le haut parleur est une pièce électronique capable d'émettre le son. On le connecte généralement à l'UC sur le port son

IV. Le scanner

le scanner est une pièce électronique capable de créer un fichier numérique en lisant le contenu d'un document physique (une lettre écrite sur un format par exemple). On connecte généralement le scanner à l'unité centrale sur un port USB

V. L'imprimante

L'imprimante est une pièce électronique capable de transformer un fichier numérique stocké dans une unité de stockage en un document papier. On connecte généralement le clavier à l'UC sur un port USB

VI. La clé USB

La clé USB (universal serial bus) est un périphérique de stockage de masse de petite capacité. Elle est facile à transporter et à utiliser. On le connecte à l'ordinateur sur un port USB

VII. Les lecteurs/graveurs DVD

Le lecteur/graveur DVD est une pièce électronique permettant d'inscrire les données sur un DVD ou de le lire. L'unité centrale offre généralement une zone devant le contenir; on le connecte directement à la carte mère à travers une nappe.

II. Les périphériques de communication

**Le concentrateur**

Encore appelé le Hub, le concentrateur permet de faire communiquer tous les ordinateurs d'un réseau local en reliant tous les câbles à lui. Lorsqu'un ordinateur envoie une information à un autre, le concentrateur reçoit l'information et la redistribue à tous les autres ordinateurs, mais seul l'ordinateur à qui le message est destiné peut l'ouvrir.

#### **Le commutateur**

Encore appelé switch, le commutateur 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.

#### **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 autorisée.

#### **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.

#### **Le routeur**

Un routeur est un appareil permettant 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

Le modem

Le modem est une pièce électronique permettant la communication entre ordinateurs via une ligne téléphonique. Il transforme donc un signal analogique en un signal numérique et réciproquement. On l'utilise généralement pour connecter un ordinateur sur internet. On le connecte sur l'ordinateur à travers le port USB ou le port série

### **III. Les types de mémoires**

L'ordinateur fonctionne en exécutant des instructions et données. Ces programmes doivent être dans la mémoire vive et peuvent provenir de la mémoire morte ou de la mémoire de masse. Il existe donc trois types de mémoire : les mémoires mortes, les mémoires vives et les mémoires de masse

#### **1. Les mémoires mortes**

Les mémoires mortes sont des mémoires non volatiles ( qui ne perd pas son contenu quand l'ordinateur n'est plus alimenté) dont le contenu est fixe au moment de sa fabrication et pouvant être lue plusieurs fois par un utilisateur. Un utilisateur expérimenté peut changer le contenu de certaines mémoires mortes. Les mémoires mortes sont alors classées selon la possibilité de les programmer et les effacer:

- ROM (read only memory): le contenu est fixe au moment de la fabrication et ne peut pas être modifié
- PROM(Programmable read only memory): dont le contenu peut être programmé une seule fois pour toute par l'utilisateur
- EPROM(erasable programmable read only memory): le contenu peut être programmé plusieurs fois par l'utilisateur.
- EEPROM(electrically erasable programmable read only memory): identique au EPROM mais effaçable électriquement donc plus facile

- UVPR0M( ultra violet programmable read only memory) dont le contenu peut être programme par ultra violet
- Flash EPROM donc le contenu est programmable par flash; beaucoup plus facile à utiliser que les UVPR0M

## 2. Les mémoires vives

La mémoire vive ou RAM (readom acces memory) est la mémoire dans laquelle le processeur met les instructions et données en cours d'exécution. En raison de leur rapidité, on distingue deux types de RAM:

SRAM (Static RAM):elle est beaucoup plus rapide mais coute très cher

DRAM (dynamic RAM): elle est beaucoup plus lente mais coûte très moins cher

## 3. Les mémoires de masse

Encore appelée mémoire auxiliaire ou mémoire secondaire, les mémoires de masse sont des périphériques permettant le stockage de grandes quantités d'informations. La mémoire de masse qu'on trouve dans l'unité centrale est le disque dur

## Résumé

### Exercices de consolidation

✓

### Classe : Troisième

### Module 1 : Architecture, maintenance et tableur

### Chapitre 2: Les logiciels

### Leçon 3-4: Les logiciels

### Compétence :

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

- ✓ Comparer les logiciels systèmes et les logiciels d'applications
- ✓ Citer les exemples de système d'exploitation, de logiciels d'applications et d'utilitaires
- ✓ Lister les modes d'acquisition des logiciels
- ✓ Télécharger et installer un logiciel

### Situation-problème

Thomas vient juste d'acheter son ordinateur sur lequel est inscrit free dos. Cependant son Papa exige qu'il l'aide à saisir son rapport de mission ce soir. Très confus, Thomas ne sait pas comment faire pour que son ordinateur soit capable de faire le traitement de texte. Apportez-lui votre soutien.

### Introduction

L'ordinateur a deux parties principales: la partie matérielle et la partie logicielle. La partie matérielle encore appelée hardware, est constituée de l'unité centrale et des unités périphériques; elle sert de support à la partie logicielle. Encore appelée software, la partie logicielle est l'intelligence de l'ordinateur; c'est elle qui effectue les traitements, affiche les résultats. La partie logicielle est constituée du système d'exploitation et des logiciels d'application.

## **I. Les systèmes d'exploitation**

### **1. Rôle d'un système d'exploitation**

Un système d'exploitation est un ensemble de programmes qui sert d'interface entre le matériel et les logiciels d'application. Installé dans l'ordinateur, un système d'exploitation joue donc plusieurs rôles:

- Il gère le processeur ainsi que la mémoire
- Par l'intermédiaire des pilotes (programme qui gère le contact entre un périphérique et le SE) il gère le contact avec les périphériques
- Il offre une interface à l'utilisateur
- Il gère les logiciels d'applications (offre une interface standard d'accès aux logiciels; assure la sécurité et la fiabilité)

## **2. Exemple de systèmes d'exploitation**

Les systèmes d'exploitation les plus connus sont:

Microsoft windows

Unix

Linux et ses dérivés

OS/2 d'IBM

MAC OS

## **II. Les logiciels d'application et les utilitaires**

### **1. Les logiciels d'application**

Un logiciel d'application est un logiciel conçu pour résoudre une classe de problèmes bien précise. Les logiciels d'application sont donc gérés et protégés par le système d'exploitation. On dénombre plusieurs logiciels d'application pouvant être classés en fonction du type de problème à résoudre:

Microsoft office, open office, word perfect , ... pour la bureautique

Photoshop, inkscape, .... pour le traitement d'images

Windows média, VLC, .... pour jouer les fichiers audio et video

### **2. Les utilitaires**

Un utilitaire est un logiciel conçu pour analyser, configurer, optimiser ou entretenir une pièce d'équipement informatique, un logiciel ou les informations enregistrées sur un support informatique. Exemple: l'explorateur de fichier, le panneau de configuration, winrar, le pare-feu windows...

## **III. Les modes d'acquisition d'un logiciel**

L'acquisition d'un logiciel est régit par un contrat de licence. En fonction du type de contrat, il existe plusieurs types de logiciels :

- Les logiciels propriétaires: l'utilisateur doit l'acheter et ne peut pas le modifier
- Les logiciels gratuits: l'utilisateur peut l'utiliser et le partager sans aucun frais
- Les logiciels open source: en plus d'être gratuit, l'utilisateur peut le modifier
- Les logiciels partagés: l'auteur autorise autrui à modifier son logiciel

## **IV. Téléchargement et installation de logiciel**

**Exercice pratique :** L'objectif de ce TP est de télécharger et installer notepad++

Après avoir connecté votre appareil sur internet, saisissez dans le moteur de la recherche Google "télécharger notepad++" ; choisissez le lien le plus approprié et cliquez sur le bouton "download" ou "télécharger".

Après le téléchargement, faites un clic droit sur le fichier obtenu et choisissez "installer" puis suivez la procédure d'installation.

À partir du menu "démarrer", lancez notepad++ et saisissez votre nom et classe puis enregistrez votre fichier sous votre nom

## **Résumé**

## **Exercices de consolidation**

**Classe : Troisième**  
**Module 1 : Architecture, maintenance et tableur**  
**Chapitre 3: Assurer le bon fonctionnement d'un ordinateur**  
**Leçon 3-4: Assurer le bon fonctionnement d'un ordinateur**

**Compétence :**

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

- ✓ Identifier les types de maintenance
- ✓ Définir les termes virus, antivirus
- ✓ Citer les mesures de protection des logicielles, matérielles
- ✓ Scanner une unité de stockage
- ✓ Installer, réparer et désinstaller les logiciels (système d'exploitation, suite bureautique, antivirus...)
- ✓ Mettre à jour un antivirus
- ✓ Défragmenter, vérifier les erreurs sur un volume

**Situation-problème**

Vous êtes fatigués d'aller sur le maintenancier ? Vous voulez vous-même protéger votre machine ? Vous voulez économiser votre argent ?

Ce chapitre doit vous offrir quelques astuces pour débiter.

**Introduction**

La maintenance informatique est l'ensemble des tâches ayant pour objectif l'entretien de l'outil informatique et son adaptation aux modifications de son environnement. En informatique on distingue la maintenance matérielle de la maintenance logicielle. La maintenance matérielle porte sur la carte mère, ses composantes et les différentes unités d'entrées/sorties. Les maintenances logicielles prennent généralement la forme de 'mini-programmes' appelés 'mise à jour'.

**I. Les causes de dysfonctionnement et les mesures de protection**

**1. Les causes de dysfonctionnement matériel**

Les composants du matériel informatique sont vulnérables :

- Aux problèmes électriques : coupures d'alimentation, surtensions, baisse de tension
- Aux mauvaises connexions des composantes
- A la poussière : elle cause le surchauffage des composants qui se détériorent rapidement
- Pendant les premiers mois d'utilisation : le matériel n'est pas fiable

**2. Les causes de dysfonctionnement logiciel**

Le dysfonctionnement des logiciels est en général dû aux échanges entre un ordinateur et le monde extérieur. On distingue deux générateurs de dysfonctionnement logiciel :

- **Les virus** : ils sont introduits par les utilisateurs à travers les connexions internet/réseaux, les périphériques de stockage de masse.
- **Un utilisateur malavisé ou malintentionné** : un tel utilisateur peut par mégarde ou volontairement supprimer les fichiers systèmes, surcharger ou fragmenter le disque dur, divulguer les informations confidentielles

Ces générateurs peuvent engendrer les dysfonctionnements suivants : le ralentissement voire le blocage de la machine, la suppression des documents, la modification des programmes et la diffusion d'informations privées.

**3. Les mesures de protection matérielle**

La protection du matériel passe par :

- L'achat du matériel de bonne qualité, assorti d'une garantie minimale d'un an
- L'utilisation des onduleurs, régulateurs de tension contre les problèmes électriques
- Un emplacement sain contre la poussière, l'eau, ...

#### 4. les mesures de protection logicielle

Ces mesures sont :

- installer un antivirus
- installer un pare-feu (fire wall) et bien configurer afin qu'il ne laisse pas les sites indésirables
- ne partager que les dossiers indispensables en cas de connexion réseau
- connaître les bases de la gestion des fichiers
- protéger par les mots de passe les données confidentielles
- effectuer les sauvegardes régulières de l'état du système d'exploitation ou simplement des données qui s'y trouvent

### II. Les virus et les antivirus

#### 1. les virus

Un virus est un programme informatique malveillant. Par conséquent il a pour objectif de nuire l'utilisateur de la machine destinataire. Un virus peut : détruire les fichiers, bloquer les programmes, divulguer les données confidentielles. Il existe plusieurs types de virus : les vers informatiques, les chevaux de bois, les bombes logiques, ...

#### 2. Les antivirus

Un antivirus est un logiciel conçu pour repérer les traces d'activités des virus, les bloquer, les mettre en quarantaine et rétablir éventuellement les fichiers endommagés.

Pour bien faire son travail l'antivirus utilise plusieurs techniques :

- La connaissance des caractéristiques d'un virus (sa signature)
- La détection des fichiers attaqués par un virus
  - La création des listes de renseignement sur tous les fichiers systèmes en vue de détecter d'éventuelles modifications ultérieures par un virus
  - La détection d'ordres suspects
- La surveillance des lecteurs de supports amovibles : disquettes, cd/dvd, clé-usb, ...

Un antivirus doit être mise à jour régulièrement. Les mises à jour sont des programmes supplémentaires qui permettent à l'antivirus de détecter de nouveaux virus

Pour effectuer une mise à jour, on doit connecter l'ordinateur sur internet et activer l'option mise à jour de l'antivirus

Pour faire son travail l'antivirus commence par scanner les supports de stockage. Scanner un support de stockage consiste à le parcourir pour détecter tout ce qui est suspect. Après avoir scanné le support de stockage, l'antivirus propose à l'utilisateur plusieurs options :

- Supprimer les virus
- Mettre en quarantaine les virus c'est-à-dire l'empêcher de fonctionner normalement sans le supprimer
- Réparer les fichiers attaqués.

Voici quelques antivirus : avast, norton, kapersky, ...

La procédure d'un antivirus est similaire à celui de n'importe quel logiciel à une exception près : deux antivirus ne doit pas être installé dans un même ordinateur car tout antivirus suspecte son collègue d'être un virus.

La procédure d'installation d'un antivirus est la suivante :

- Désinstaller tout autre antivirus
- Télécharger l'antivirus ou se procurer d'un cd d'installation et l'insérer dans le lecteur
- S'agréer aux termes du contrat ou annuler l'installation
- Entrer éventuellement la clé du produit (fournit par le fabricant de l'antivirus)
- Poursuivre la configuration des paramètres en suivant la procédure proposée

**Exercice d'application** : télécharger et installer la version gratuite de avast.

### **III. Gestion des disques et mémoires**

Le ralentissement et le blocage de la machine peuvent être engendrés par la fragmentation (éparpillement d'un fichier en plusieurs morceaux) ou le bourrage du disque dur.

Pour surmonter ces difficultés, l'utilisateur peut :

- Libérer l'espace dans son disque dur ; cela peut se faire de deux manières :
  - Suppression des fichiers utilisés rarement (manuellement ou automatiquement)
  - Suppression des applications utilisées rarement
- Défragmenter le disque dur : la défragmentation consiste à réunir autant que possible les espaces de stockage des morceaux d'un fichier
- Partitionner le disque dur : le partitionnement consiste à découper logiquement en plusieurs sous-unités (partitions) un disque dur ; ces sous unités sont gérées comme des disques à part entier. Ainsi le blocage d'un disque ne contenant pas le système d'exploitation ne peut pas rendre la machine lente, le plantage d'un disque ne peut pas entraîner les pertes de données dans l'autre, les données sont faciles à utiliser. On distingue deux types de partition :
  - La partition principale (celle contenant le SE)
  - Les partitions étendues celles ne pouvant contenir que les données)

La procédure de défragmentation ou de partitionnement est :

- Ouvrir la fenêtre des outils systèmes (démarrer, tous les programmes, accessoires, outils systèmes, défragmenter ou partitionner)
- Cliquer sur créer pour défragmenter ou partitionner
- Cliquer sur suivant pour défragmenter ou partitionner
- Poursuivre la procédure en respectant les consignes

### **Classe : Troisième**

#### **Module 1 : Architecture, maintenance et tableur**

#### **Chapitre 5: Utilisation des fonctions dans un tableur**

#### **Leçon 9-13: Utilisation des fonctions dans un tableur**

#### **Compétence :**

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

- ✓ Saisir ou recopier une formule
- ✓ Utiliser les fonctions de texte
- ✓ Utiliser les fonctions de date
- ✓ Utiliser les fonctions d'heure
- ✓ Utiliser les fonctions logiques
- ✓ Utiliser les fonctions trigonométriques
- ✓ Utiliser les fonctions mathématiques

#### **Situation-problème**

Votre ami vient une fois de plus vous dire qu'il est plus facile d'effectuer les fonctions mathématiques avec un tableur qu'une calculatrice scientifique. De plus il vous dit que vous pouvez utiliser d'autres fonctions très puissantes avec un tableur qui ne se trouvent même pas dans une calculatrice scientifique ? Comment allez-vous procéder de cette puissance des tableurs ?

#### **Introduction**

L'une des puissances d'un tableur est de réaliser en un seul clic une opération très longue. Pour y arriver les tableurs offrent aux utilisateurs une panoplie de fonctions. L'étude de quelques une de ces fonctions fait l'objet de cette leçon.

## **I. Saisir et recopier une formule**

### **1. Saisir une formule**

Les formules décrivent un ensemble de calcul pouvant faire intervenir plusieurs cellules. Les données de type formule commencent généralement par le signe = suivi d'une expression ou d'une formule.

Pour saisir une donnée de type formule, on commence par activer la cellule, puis saisir le signe = et enfin saisir l'expression ou la formule.

### **2. Recopier une formule**

Les propriétés de recopie d'une formule simple sont similaires à celle de la copie incrémentée. Lors de la recopie d'une formule les adresses absolues restent fixes et les autres varient en respectant les propriétés vues au chapitre précédent.

## **II. Notion de fonctions**

Les fonctions sont des formules déjà écrites qui effectuent des calculs ou des traitements à partir de données que vous leur fournissez. Elles vous évitent de réécrire ces formules.

Exemple : pour calculer la somme des cellules de C1 à C10, vous pourriez très bien écrire =C1+C2+C3+C4+C5+C6+C7+C8+C9+C10. Cela fonctionne parfaitement. Mais vous trouverez sans doute plus facile d'écrire =SOMME(C1:C10)

Pour donner des résultats, la plupart des fonctions nécessitent que vous leur fournissiez des données pour travailler. Ces données sont appelées des arguments. Exemple : la fonction SOMME( ) prend en argument les cellules C1 à C10.

Les arguments doivent figurer après le nom de la fonction, entre parenthèses et séparés par des points-virgules. Il est impératif de respecter leur ordre, car, en règle générale, chacun d'entre eux a un rôle spécifique. Il est également nécessaire de veiller au type d'argument demandé (valeurs numériques, chaînes de caractères, dates...) sous peine de voir apparaître des messages d'erreur tels que #VALEUR!

La procédure générale pour utiliser une fonction dans Excel 2003 est la suivante :

- Cliquer sur la cellule dans laquelle on veut l'utiliser
- Dans le menu insertion, cliquer sur fonction
- Dans la boîte de dialogue qui s'ouvre, choisir la catégorie de fonction, puis la fonction elle-même et cliquer sur OK
- Dans la boîte de dialogue qui va apparaître, spécifier les paramètres

## **III. Les fonctions texte**

Elles permettent d'effectuer les traitements sur les caractères d'un texte ou d'une chaîne de caractère en texte. Ils existent plusieurs fonctions textes :

- **La fonction cherche** : permet de rechercher la position d'un caractère dans une chaîne ou un texte. Sa syntaxe est la suivante : =CHERCHE(texte\_cherché;texte). L'argument texte\_cherché est le texte qu'on doit rechercher et l'argument texte est le texte dans lequel on doit effectuer la recherche. Exemple : si je veux chercher la position du mot 'bonjour' dans le texte saisi dans la cellule B7, je vais saisir : =CHERCHE('bonjour' ;B7)
- **La fonction remplacer** : permet de remplacer une chaîne de caractère par une autre. Sa syntaxe est la suivante : =REPLACER(ancien\_texte ;no\_départ ;no\_car ;nouveau\_texte) l'argument ancien\_texte est le texte qu'on doit remplacer, l'argument no\_départ est la position du premier caractère de la chaîne ancien\_texte où le remplacement par nouveau\_texte doit commencer, l'argument no\_car Nombre de caractères



d'ancien\_texte que nouveau\_texte doit remplacer, l'argument nouveau\_texte est le texte qui doit remplacer les caractères d'ancien\_texte.

- **la fonction concatener** : permet de concaténer plusieurs chaînes en une seule chaîne. Sa syntaxe est la suivante : =CONCATENER(texte1 ;texte2 ;..... ;texteN). Les arguments sont les différents textes qu'on doit concaténer.

**Exercice pratique** : dans les cellules : A2, B7, D8, G9, F5, F7, E1, E4, D3 saisir respectivement Bonjour, le, monde, des, nouveaux, ou, des, futurs, informaticiens. Dans A3 utiliser la fonction recherche pour afficher la position du syllabe ma dans la chaîne qui a été saisie dans D3 ; dans la cellule B8, utiliser la fonction remplacer pour afficher le contenu de A2 en changeant les caractères onjour par ienvenu ; utiliser la fonction concatener pour afficher en une seule phrase le contenu des cellules A2, B7, D8, F5, E1, E4, D3. Amuser vous comme vous voulez avec les autres fonctions.

#### IV. Utilisation des fonctions de gestion du temps

Elles permettent d'effectuer les traitements sur les dates et les heures.

Rappelons au préalable que vous pouvez saisir les dates dans Excel comme suit :

- 04/12/09 Excel affiche 04/12/2009.
- 04-12-09 Excel affiche 04/12/2009.
- 04 décembre 2009 Excel affiche 04-déc-2009.
- 04/12 Excel affiche 04-déc

De même vous pouvez saisir les heures dans excel comme suit :

- 23:6 Excel affiche 23:06.
- 23: est affiché 23:00.

Les tableurs nous offrent plusieurs fonctions de gestion du temps :

- **La fonction date** : affiche la date correspondante aux arguments spécifiés. Sa syntaxe est la suivante : =DATE(année ;mois ;jour) . L'argument année est l'année à afficher, l'argument mois est le mois à afficher, l'argument jour est le jour à afficher
- **La fonction annee**: affiche l'année correspondante à la date saisie en paramètre. Sa syntaxe est la suivante : =ANNEE(année) ; la fonction =AUJOURDHUI() renvoie l'année courante.
- **La fonction mois** : affiche le numéro mois correspondante à une date prise en argument. Sa syntaxe est la suivante : =MOIS(date)
- **La fonction jour** : affiche le numéro du jour du mois correspondante à une date prise en argument. Sa syntaxe est la suivante : =JOUR(date).
- **la fonction heure** : affiche l'heure correspondante à la date ou l'heure saisie en argument. Sa syntaxe est la suivante : =HEURE(date)

**Exercice pratique** : dans les cellules : A2, B7, D8, G9 C6 saisir une date, une année, un jour et une heure. Dans la cellule E1 utiliser la fonction date pour afficher la date en prenant l'année dans la cellule B7, le mois dans D8 et le jour dans G9. Dans trois cellules au choix, utiliser les fonctions mois(), jour(), annee(), heure() pour afficher respectivement le mois, le jour, l'année, l'heure à partir du contenu de la cellule A2.

#### V. Utilisation des fonctions logiques

Il est question ici d'utiliser un tableur pour afficher une valeur qui si elle vérifie certaines conditions. Exemple : afficher les clients qui viennent samedi et font des achats supérieurs à 10000Fr. Les fonctions capables de faire ces types de traitement s'appellent les fonctions logiques. Les fonctions logiques sont les fonctions qui permettent de vérifier la validité de certaines conditions. Il existe plusieurs fonctions logiques :

- **La fonction ET** : elle affiche faux si l'un des arguments est faux et vrai si tous les arguments sont vrais. Sa syntaxe est la suivante : ET(valeur\_logique1 ;valeur\_logique2 ;.... ;valeur\_logiqueN). Exemple : ET(A1>10000 ;H3= 'Dimanche')

- **La fonction OU** : elle affiche vrai si l'un des arguments est vrai et faux si tous les arguments sont faux. Sa syntaxe est la suivante :  
ET(valeur\_logique1 ;valeur\_logique2 ;... ;valeur\_logiqueN). Exemple :  
OU(A1>10000 ;H3= 'Dimanche')
- **La fonction FAUX** : elle affiche la valeur 'faux' dans la cellule. Sa syntaxe est la suivante : =FAUX() ; elle ne possède pas d'arguments.
- **La fonction VRAI** : elle affiche la valeur 'vrai' dans la cellule. Sa syntaxe est la suivante : =VRAI() ; elle ne possède pas d'arguments.

**Exercice pratique** : dans les cellules : A1, B1, C1 saisir nom, jour quantité. Dans la plage A2 :A8 saisir les noms de 7 personnes ; dans la plage B2 :B8 saisir la première lettre de chaque jour de la semaine ; dans la plage C2 :C8 saisir 7 multiples de 1000 avec au moins 2 valeurs supérieures à 10000 et 2 valeurs inférieures à 10000. Dans la plage D2 :D8 : ET( B3=M ;C3>10000) ; ET( B3=M ;C3<10000) ; OU( B3=M ;C3>10000) ; OU( B3=M ;C3<10000) ; ET( B5=M ;C5>10000) ; ET( B5=M ;C5<10000) ; OU( B5=M ;C5>10000)

## VI. Utilisation des fonctions mathématiques

Les fonctions mathématiques sont des fonctions qui permettent d'effectuer les opérations mathématiques. Ils existent plusieurs :

- La fonction MOYENNE(paramètre1; paramètre2; .... ; paramètreN) permet de calculer la moyenne de paramètre1, ...,paramètreN où paramètrei correspond à une valeur ou à l'adresse d'une cellule.
- La fonction SOMME(paramètre1; paramètre2; .... ; paramètreN) affiche la somme des paramètres
- La fonction MAX(paramètre1; paramètre2; .... ; paramètreN) affiche la plus grande valeur entre les paramètres
- La fonction MIN(paramètre1; paramètre2; .... ; paramètreN) affiche la plus petite valeur entre les paramètres
- **Exercice pratique** : reprenez l'exercice pratique ci-dessus pour les fonctions mathématiques

## VII. insertion de graphiques

La procédure d'insertion de graphiques est la suivante :

- Sélectionner la liste à représenter sous forme de graphique
- Cliquer sur le menu graphique
- Cliquer sur type de graphique
- Choisir son type de graphique
- Cliquer sur ok

## Résumé

Les fonctions sont des formules déjà écrites qui effectuent des calculs ou des traitements à partir de données que vous leur fournissez. Le bouton fonction du menu insertion affiche une boîte de dialogue qui nous offre plusieurs fonctions regroupées en catégories. Une fonction peut avoir 0 ou plusieurs arguments chacune séparée de l'autre par un point virgule. Un argument peut être une valeur ou l'adresse d'une cellule.

## Exercices de consolidation

1)

**Classe : Troisième**  
**Module 2 : Numération et algorithmes**  
**Chapitre 6: Utilisation des systèmes de numération**  
**Leçon 14-15: Initiation aux systèmes de numération**

**Compétence :**

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

- ✓ Définir les termes système de numération, base
- ✓ Lister les bases usuelles (2 – 8 – 10 – 16),
- ✓ Identifier les symboles des bases Usuelles
- ✓ Convertir base 10  $\Leftrightarrow$  base 2, 10  $\Leftrightarrow$  base 8, 10  $\Leftrightarrow$  base 16
- ✓ Convertir base 2  $\Leftrightarrow$  base 8, base 2  $\Leftrightarrow$  base 16, base 8  $\Leftrightarrow$  base 16

**Situation-problème**

On raconte que l'ordinateur ne connaît que deux nombres : 0 et 1. Pourtant nous travaillons plutôt avec 10 nombres allant de 0 à 9. Comment l'ordinateur fait-il pour nous comprendre ?

**Introduction**

Le langage que nous utilisons nous offre : un ensemble de lettres (a-z, A-Z), de symboles (+, \*, ^, ...), de nombres (0 à 9) et une règle d'écriture pour donner un sens au message que nous voulons transmettre: c'est le français. D'autres langages comme le chinois, le perse, le bamoun offrent aussi à leurs utilisateurs un alphabet (ensemble de caractères, nombres, symboles) et une règle d'écriture (la syntaxe et le sémantique). L'ordinateur quand à lui parle un langage : le binaire. L'alphabet du langage binaire ne connaît que deux nombres : 0 et 1 ; cependant il dispose aussi sa règle d'écriture.

Pour cette leçon, nous nous intéresserons uniquement aux nombres.

**I. Les bases usuelles**

Un système de numération décrit la façon dont les nombres sont représentés. Un système de numération est décrit par : un alphabet (ensemble de symboles ou de chiffres), des règles d'écritures des nombres et de juxtapositions de symboles.

**Exemple de système de numération :** système de numérotation romaine, babylonien, décimale, binaire, octale, hexadécimale

Une base est le nombre de chiffres (ou symboles) utilisés dans un système de numérotation. Le système décimale utilise 10 chiffres (0,1, 2, 3, 4, 5,6, 7, 9) : on l'appelle encore la base 10 ou base décimale. Le système binaire utilise 2 chiffres (0, 1) : on l'appelle encore la base 2 ou base binaire. Le système octal utilise 8 chiffres (0, 1, 2, 3, 4, 5, 6, 7) : on l'appelle encore la base 8 ou base octale. Le système hexadécimal utilise 16 chiffres et 6 symboles (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) : on l'appelle encore la base 16 ou base hexadécimale.

**II. Conversion des nombres en base décimale**

**1. La base décimale**

La base décimale est la base que nous utilisons couramment. Pour écrire un nombre dans cette base, on juxtapose les chiffres qu'elle autorise.

La règle d'écriture d'un nombre en décimale est simple : un chiffre donné dépasse le chiffre qui est à sa droite d'une dizaine d'unités, le chiffre le plus à droite vaut seulement une unité. **Exemple :**  $10879 = 1 \cdot 10^4 + 0 \cdot 10^3 + 8 \cdot 10^2 + 7 \cdot 10^1 + 9 \cdot 10^0 = 10000 + 0 + 800 + 70 + 9 = 10000 + 800 + 70 + 9 = 10800 + 70 + 9 = 10870 + 9 = 10879$

**2. La base binaire**

Le binaire est un mode de représentation des chiffres utilisés dans les composants électroniques. En effet l'électricité peut être présente ou absente dans un composant électronique donné : alors on représente par 1 la présence de l'électricité dans le composant et

par 0 son absence ; cette représentation correspond parfaitement au binaire qui ne connaît que deux chiffres 0 et 1.

La règle d'écriture d'un nombre en binaire est simple : un chiffre donné dépasse le chiffre qui est à sa droite de deux d'unités, le chiffre le plus à droite vaut seulement une unité.

Pour convertir un binaire en décimale, il suffit de réécrire le nombre en appliquant la règle d'écriture. **Exemple** :  $1010011 = 1 * 2^6 + 0 * 2^5 + 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 64 + 0 + 16 + 0 + 0 + 2 + 1 = 83$

### 3. La base octale

La base octale est utilisée par les informaticiens pour faciliter certains calculs qui devaient être faits en base 2.

La règle d'écriture d'un nombre en octale est simple : un chiffre donné dépasse le chiffre qui est à sa droite de huit unités, le chiffre le plus à droite vaut seulement une unité.

Pour convertir un octale en décimale, il suffit de réécrire le nombre en appliquant la règle d'écriture. **Exemple** :  $765042 = 7 * 8^5 + 6 * 8^4 + 5 * 8^3 + 0 * 8^2 + 4 * 8^1 + 2 * 8^0 = 7 * 32768 + 6 * 4096 + 5 * 512 + 0 * 64 + 4 * 8 + 2 * 1 = 256546$

### 4. La base hexadécimale

Les informaticiens utilisent de moins en moins la base octale au profit de la base hexadécimale. L'alphabet de la base hexadécimale contient 10 chiffres (0 à 9 correspondant aux chiffres de la base décimale) et 6 symboles : A ⇔ 10, B ⇔ 11, C ⇔ 12, D ⇔ 13, E ⇔ 14, F ⇔ 15. Ainsi l'alphabet de la base hexadécimale correspond aux 16 premiers décimaux.

La règle d'écriture d'un nombre en hexadécimale est simple : un chiffre donné dépasse le chiffre qui est à sa droite de 16 unités, le chiffre le plus à droite vaut seulement une unité.

Pour convertir un hexadécimale en décimale, il suffit de réécrire le nombre en appliquant la règle d'écriture. **Exemple** :  $AB08F = 10 * 16^4 + 11 * 16^3 + 0 * 16^2 + 8 * 16^1 + 15 * 16^0 = 10 * 65536 + 11 * 4096 + 0 * 256 + 8 * 16 + 15 * 1 = 700559$

**Remarque** : Le transcodage (ou conversion de base) est l'opération qui permet de passer de la représentation d'un nombre exprimé dans une base à la représentation du même nombre mais exprimé dans une autre base.

## III. La conversion d'un nombre écrit en décimale en binaire, octale et hexadécimale

### 1. Règle général

Pour convertir un nombre de la base 10 vers une base B (2, 8, 16), la règle à suivre est la division successive par b :

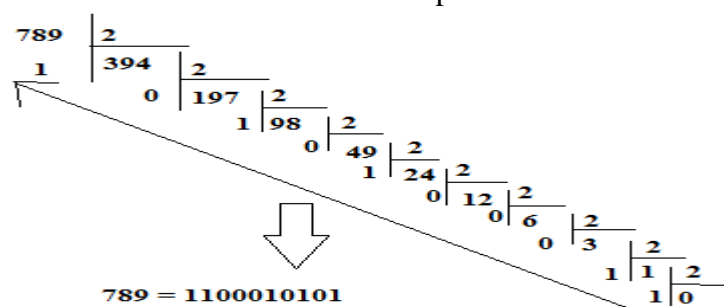
- on divise le nombre b
- on divise le quotient par la base b
- on continue de diviser le quotient par b jusqu'à l'obtention d'un quotient nul
- la suite des restes correspond aux symboles de la base visée.
- on obtient en premier le chiffre de poids faible (il a seulement une unité) et en dernier le chiffre de poids fort (il a le plus grand nombre d'unités).

### 2. Exemple de conversion d'un décimale en binaire

D'après la règle générale, on doit effectuer une suite de divisions par 2.

Convertissons le décimal

789 en base 2 :



Le nombre 789 écrit en décimal est égale au nombre

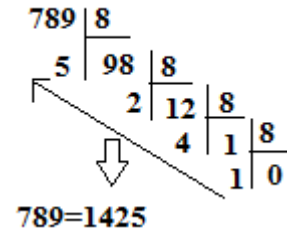
1100011 en binaire

### 3. Exemple de conversion d'un décimal en octal

D'après la règle générale, on doit effectuer une suite de divisions par 8.

Convertissons le décimal 789 en octal

Le nombre 789 écrit en décimal est égale au nombre 1425 en octal

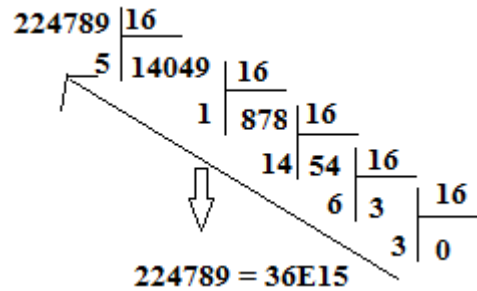


### 4. Exemple de conversion d'un décimal en hexadécimal

D'après la règle générale, on doit effectuer une suite de division par 16.

Convertissons le décimal 224789 en hexadécimal

Rappelons que 10 en décimal vaut A en hexadécimal, 11 ⇔ B, 12 ⇔ C, 13 ⇔ D, 14 ⇔ E et 15 ⇔ F. Le nombre 224789 écrit en décimal est égale au nombre 36E15 en hexadécimal



## IV. Conversion d'un nombre binaire en octal ou en hexadécimal

Pour convertir un nombre binaire en octal, on regroupe de la droite vers la gauche les chiffres de ce nombre en groupe de 3 chiffres, puis on convertit indépendamment chaque groupe de chiffres comme si on voulait le convertir en décimale ; si le groupe de chiffres le plus à gauche est inférieur à trois chiffres, on augmente les 0 à droite de façon à compléter le groupe à trois chiffres

**Exemple :** convertissons le nombre binaire 11000101011101 en octal.

11000101011101=011-000-101-011-101=3-0-5-3-5=30535. Ainsi, le nombre binaire 11000101011101 est égale à 30535 en octal.

Pour convertir un nombre binaire en hexadécimal, on regroupe de la droite vers la gauche les chiffres de ce nombre en groupe de 4 chiffres, puis on convertit indépendamment chaque groupe de chiffres comme si on voulait le convertir en décimale ; si le groupe de chiffres le plus à gauche est inférieur à quatre chiffres, on augmente les 0 à droite de façon à compléter le groupe à quatre chiffres

**Exemple :** convertissons le nombre binaire 11000101011101 en octal.

11000101011101=0011-0001-0101-1101=3-1-13=3-1-C=31C. Ainsi, le nombre binaire 11000101011101 est égal à 31C en hexadécimal.

## V. Conversion d'un nombre écrit en octal ou hexadécimal en un nombre binaire

Pour convertir un nombre écrit en octal en un nombre écrit en binaire, on décompose chaque chiffre de ce nombre en un nombre binaire écrit sur 3 chiffres ; si le résultat de la décomposition d'un chiffre est inférieur à 3 chiffres, on complète la partie gauche avec les 0 de façon à obtenir 3 chiffres.

**Exemple :** convertissons le nombre octal 37265 en binaire.  $37265 = 3-7-2-6-5 = 11-111-10-110-101 = 011-111-010-110-101 = 011111010110101 = 11111010110101$

Le nombre octal 37265 s'écrit en binaire 11111010110101

Pour convertir un nombre écrit en hexadécimal en un nombre écrit en binaire, on décompose chaque chiffre de ce nombre en un nombre binaire écrit sur 4 chiffres ; si le résultat de la décomposition d'un chiffre est inférieur à 4 chiffres, on complète la partie gauche avec les 0 de façon à obtenir 4 chiffres.

**Exemple :** convertissons le nombre octal E3A26 en binaire.  $E3A26 = E-3-A-2-6 = 1111-11-1010-10-110 = 1111-0011-1010-0010-0110 = 11110011101000100110$

Le nombre hexadécimal E3A26 s'écrit en binaire 11110011101000100110

### Résumé

Un système de numération se décrit par : un alphabet (les nombres et les symboles qui permettent de représenter les chiffres à une unité dans le système), une syntaxe (la façon d'écrire un nombre dans le système) et une sémantique (les règles qui permettent d'écrire un nombre dans le système de façon à ce qu'il ait un sens). L'ordinateur utilise uniquement la base binaire ; pour simplifier les calculs, les informaticiens utilisent les bases octales et hexadécimales. La base décimale est celle que nous utilisons chaque jour. Pour convertir un décimal en une base B, on effectue les divisions successives par B et on regroupe les restes. Pour convertir un binaire en octal (respectivement en hexadécimal), on regroupe les chiffres en groupe de 3 (respectivement en groupe de 4). Pour convertir un octal (respectivement un hexadécimal) en binaire, on décompose chaque chiffre en groupe de 3 (respectivement 4).

### Exercices de consolidation

- 2) effectuer les conversions suivantes dans les bases indiquées :  $(147)_8=(\dots)_{10}$  ;  $(1A)_{16}=(\dots)_{10}$  ;  $(1000111)_{2}=(\dots)_{10}$  ;  $(ABF15)_{16}=(\dots)_{10}$  ;  $(78A)_{16}=(\dots)_{10}$  ;  $(547)_8=(\dots)_{10}$  ;  $(152)_{10}=(\dots)_{10}$
- 3)

## Classe : Troisième

### Module 2 : Numération et algorithmes

### Chapitre 6: Utilisation des systèmes de numération

### Leçon 16: Les opérations en base 2

#### Compétence :

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

- ✓ Additionner en base 2

#### Situation-problème

On raconte que l'ordinateur ne connaît qu'un seul langage : le binaire (constitué de deux nombres 0 et 1). Pourtant il effectue de très grands calculs en peu de temps. Comment puis-je effectuer juste les opérations élémentaires dans ce langage ?

#### Introduction

Additionner, soustraire, multiplier sont les opérations arithmétiques que nous utilisons chaque jour. Dans cette leçon il est question de reprendre ces opérations en base décimale en observant les règles qui s'y cachent ; puis d'appliquer ces règles pour effectuer ces opérations en binaire, octal et hexadécimal.

#### **I. L'addition en base 10, 2**

##### **1. Addition en base 10**

Additionnons les nombres 4709 et 5628 :

Pour additionner deux nombres en base dix, on doit respecter les règles suivantes :

- Positionner les deux nombres : l'un au dessous de l'autre
- Ranger le positionnement des chiffres de façon à ce que les chiffres de même poids soit sur la même colonne ; remplir éventuellement par des 0 les colonnes vides les plus à droites
- Additionner les chiffres de même poids en commençant par le chiffre de poids le plus faible
- Positionner le résultat en respectant l'alignement des colonnes

$$\begin{array}{r} \phantom{+} \phantom{0} \phantom{4} \phantom{7} \phantom{0} \phantom{9} \\ + \phantom{0} \phantom{5} \phantom{6} \phantom{2} \phantom{8} \\ \hline 1 \phantom{0} \phantom{3} \phantom{3} \phantom{7} \end{array}$$

- Si lors d'un calcul le résultat est supérieur ou égale à 10 alors attribuer le chiffre de poids faible du résultat de ce calcul au résultat final et ramener le chiffre de poids fort (la retenue) à la colonne suivante : on doit l'augmenter à l'addition de cette colonne
- Si la colonne la plus à droite génère aussi la retenue, on pourra créer une nouvelle colonne à sa droite avec les valeurs 0 et appliquer les règles connues.

+	0	1
0	0	1
1	1	10

## 2. Addition en base 2

L'addition en base deux est similaire à l'addition en base dix. Cependant, on doit connaître comment s'additionne les nombres binaires à un seul chiffre : voici un tableau récapitulatif :

Additionnons les binaires : 11001010111 et 1011110011

Pour additionner deux nombres en base deux, on doit respecter les règles suivantes :

- Positionner les deux nombres : l'un au dessous de l'autre
- Ranger le positionnement des chiffres de façon à ce que les chiffres de même poids soit sur la même colonne; remplir éventuellement par des 0 les colonnes vides les plus à droites
- Additionner les chiffres de même poids en commençant par le chiffre de poids le plus faible
- Positionner le résultat en respectant l'alignement des colonnes
- Si lors d'un calcul le résultat est supérieur ou égale à 10 alors attribuer le chiffre de poids faible du résultat de ce calcul au résultat final et ramener le chiffre de poids fort (la retenue) à la colonne suivante : on doit l'augmenter à l'addition de cette colonne
- Si la colonne la plus à droite génère aussi la retenue, on pourra créer une nouvelle colonne à sa droite avec les valeurs 0 et appliquer les règles connues.

$$\begin{array}{r}
 11111111 \\
 011001010111 \\
 + 001011110011 \\
 \hline
 100101001010
 \end{array}$$

### Résumé

Les opérations en base deux sont similaires aux opérations en base dix à une exception près : si deux chiffres sont consécutifs dans un nombre en base deux, le chiffre à gauche dépasse le chiffre à droite de deux unités alors que dans la base dix, il le dépasse plutôt de dix unités.

### Exercices de consolidation

- 4) Posez et effectuez les opérations suivantes en binaire : 11001+11000 ; 10001+11111, 10101+10101 ; 110011+

**Classe : Troisième**  
**Module 2 : Numération et algorithmes**  
**Chapitre 2: La codification**  
**Leçon 5-8: Le codage en ASCII**

**Compétence :**

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

- ✓ Définir les termes information, donnée, code, codage, codification
- ✓ Citer les qualités d'une bonne information
- ✓ Expliquer l'importance de la codification
- ✓ Utiliser la table du code ASCII pour coder un alphabet, une lettre, une expression

### **Situation-problème**

Je vous ai dit depuis le début : mon objectif est d'utiliser le même langage que la machine. La base 2 m'a permis de pouvoir effectuer les calculs comme elle ; cependant comment écrire un texte comme elle ?

### **Introduction**

Une information est le support formel d'un élément de connaissance humaine susceptible d'être représentée à l'aide de conventions (codages) afin d'être conservée, traitée ou communiquée. Une bonne information doit être précise, fiable, pertinente, vérifiable. Une donnée est la représentation d'une information sous une forme codée destinée à faciliter son traitement. Le codage consiste à établir une correspondance entre la représentation de d'une même information dans deux langages différents. Un code est ensemble de signes ou signaux, de règles et de lois permettant la communication dans un langage ou dans un jargon donné. La codification est le fait d'utiliser un code pour communiquer. L'ordinateur et les informaticiens utilisent plusieurs codes permettant d'interpréter une suite de nombres binaires : binaire, le décimal, l'hexadécimal, l'octal, le BCD, le Gray, L'ASCII, .... Dans la suite, nous nous intéresserons uniquement au code ASCII

#### **I. Présentation du code ASCII**

Né vers 1960, le code ASCII (American Standard Codification Interchange Information) permet de représenter les nombres, les caractères alphabétiques, les signes de ponctuations et les caractères spéciaux.



		000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	'	p
0001	1	SOH	DC1		1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	10	LF	SUB	*	:	J	Z	j	z
1011	11	VT	ESC	+	;	K	[	k	{
1100	12	FF	FS	,	<	L	\	l	!
1101	13	CR	GS	-	=	M	]	m	}
1110	14	SO	RS	.	>	N	'	n	~
1111	15	SI	US	/	?	O	-	o	DEL

Il existe deux versions du code ASCII :

- la première version appelée tout simplement code ASCII permet de représenter chaque symboles sur 7 bits, il ne peut représenter ainsi que  $2^7 = 128$  symboles différents ; par conséquent il ne permet pas de représenter pas certains types de caractères comme les caractères accentués
- la première deuxième version appelée code ASCII étendu permet de représenter chaque symboles sur 8 bits, il peut représenter jusqu'à ainsi que  $2^8 = 256$  symboles différents ; il ne permet de représenter les caractères accentués

Dans le cadre de cette leçon, nous utiliserons le code ASCII simple.

Ci-dessus se trouve un tableau représentant les symboles et leur code ASCII.

## II. Utilisation de la table du code ASCII

### 1. Le codage d'un symbole se trouvant dans la table ASCII

Pour coder un symbole en ASCII, on doit :

- Repérer le symbole à coder
- A partir du symbole repéré, parcourir verticalement le tableau jusqu'à la première ligne, lire et écrire le code (sur trois bits) qui s'y trouve
- A partir du symbole repéré, parcourir horizontalement le tableau jusqu'à la première colonne, lire et écrire le code (sur quatre bits) qui s'y trouve
- Mettre les deux codes côte à côte et obtenir le code ASCII du symbole repéré

**Exemple :** coder le symbole qui permet de faire l'espace entre deux mots : Dans la table ASCII, ce symbole est identifié par : ESC. En parcourant la table verticalement depuis le symbole ESC, on trouve à la première ligne : 001. En parcourant la table horizontalement depuis le symbole ESC, on trouve à la première colonne : 1011. Le code ASCII de ESC est donc 0011011.

**Exercice pratique :** Donner le code ASCII de chaque caractère de l'alphabet (A-Za-z) et des nombres allant de 0 à 9.

## **2. Le codage d'une expression**

Pour coder une expression en ASCII, on doit coder séquentiellement chaque symbole qui constitue l'expression.

**Exemple :** coder l'expression : Bonjour le code ASCII

10000101101111110111011010101101111111010111100100011011110110011001010  
011011110001111011111100100110010100110111000000110100110010011001001

**Exercice pratique :** Coder l'expression : !!Apprendre a coder ; c'est vouloir s'ouvrir au reste du monde !!

## **Résumé**

L'ordinateur ne comprend que le binaire ; nos informations sont donc représentées sous formes binaires. Le code ASCII est l'un des codes les plus utilisés pour représenter une information en binaire. . Le code ASCII étendu permet de représenter certains symboles comme les caractères accentués en plus des symboles qu'on peut représenter avec le code ASCII.

## **Exercices de consolidation**

5)

## **Classe : Troisième**

### **Module 1 : Numération et transfert algorithme**

### **Chapitre 9: Utiliser les unités de mesure**

### **Leçon 20-21: Les unités de mesure en informatique**

#### **Compétence :**

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

- ✓ Lister les unités de mesures en informatique, les multiples et les sous multiples
- ✓ Définir les unités de mesures du matériel
- ✓ Calculer la résolution de l'écran
- ✓ Convertir dans les mesures de capacités informatiques
- ✓ Décrire les unités suivantes : DPI, Inch, Hertz, BPS, Pixel et X

#### **Situation-problème**

Votre papa veut faire une sortie avec vous ! Mais pour des intérêts clairs et nets !! Il veut acheter un seul ordinateur pour une entreprise de vingt employés qu'il vient de créer !! Pour se rassurer de votre capacité à pouvoir l'aider, il vous demande de le décrire d'abord et point par point ce qui différencie deux ordinateurs. A titre exemplaire : il vous dit que ce qui différencie deux pains n'est pas forcément le prix mais plutôt la dose de chaque élément qui entre dans la constitution du pain d'un poids fixe. Comment compter vous l'aider ? Comment évaluer la dose des éléments qui entre dans chaque composant de l'ordinateur ?

#### **Introduction**

Deux objets de même nature se différencient par les propriétés qu'ils partagent ; ces propriétés proviennent des éléments qui constituent ces objets. Un ordinateur est constitué d'un ensemble de composants. Chaque composant a une capacité qui permet de juger sa capacité d'atteindre ses objectifs. La différence entre deux composants de même nature se calcule généralement par cette capacité.

### **I. Les unités de mesure du matériel informatique**

Un ordinateur est constitué de trois parties principales : l'unité centrale, les unités d'entrée et les unités de sorties.

Ces trois parties sont constituées de plusieurs composants :

### 1. La mémoire

Une mémoire est un composant électronique permettant d'enregistrer, de conserver et de restituer des informations. Les informations peuvent être écrites ou lues. Il y a écriture lorsqu'on enregistre des informations en mémoire, lecture lorsqu'on récupère des informations précédemment enregistrées.

Les caractéristiques d'une mémoire sont :

- **La capacité** : quantité total d'informations que peut contenir la mémoire. Une information dans la mémoire est codée sous forme d'un ensemble de bits. La mémoire regroupe les informations par ensemble de 8 bits encore appelé 1 octet. C'est pourquoi la capacité d'une mémoire s'exprime en octet.
- **Le format des données** : une mémoire est constituée des cases mémoires ; une case mémoire contient des emplacements pouvant stocker les bits. Un format de données est le nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du mot mémorisable.
- **Le temps d'accès** : c'est le temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture et l'arrivée de la première information
- **Le temps de cycle** : il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture.
- **Le débit** : c'est le nombre maximum d'informations lues ou écrites par seconde.
- **Volatilité** : elle caractérise la permanence des informations dans la mémoire. Il existe deux types de mémoire : les mémoires volatiles (qui ne gardent l'information que quand l'ordinateur est allumé) et les mémoires permanentes (qui gardent les informations même quand on éteint l'ordinateur). La mémoire volatile qui se trouve dans l'ordinateur est la RAM (Random Acces Memory) ; les mémoires permanents sont : la ROM (Read Only Memory), le disque dur, ...

### 2. Le processeur

Encore appelé l'unité centrale de traitement, le processeur est le composant qui fait tous les travaux que nous demandons à l'ordinateur. Lorsqu'il est de petite taille, on l'appelle microprocesseur.

Le processeur exécute tous les travaux que nous lui soumettons sous forme d'opérations : la capacité d'un processeur est donc sa fréquence (le nombre d'opérations qu'il peut exécuter en une seconde). La fréquence d'un processeur se mesure en Hertz (hz).

### 3. L'écran

L'écran est une unité de sortie permettant à l'unité centrale d'afficher le résultat de son traitement. Il existe deux types d'écran : les écrans à tube cathodique et les écrans plats.

Les caractéristiques d'un écran sont :

- **La taille** : c'est la diagonale de l'écran ; elle s'exprime en pouces (1 pouce= 2,54cm)
- **La définition** : c'est le nombre de pixel que l'écran peut afficher. Un pixel est le plus petit élément d'une représentation visuelle auquel peuvent être affecté séparément des attributs tels que luminosité, couleur, clignotement. La définition s'exprime sous forme d'une multiplication entre le nombre de pixels horizontaux l e nombre de pixels verticaux

### 4. L'imprimante

L'imprimante est une unité de sortie qui permet d'imprimer sur le papier les résultats d'un traitement informatique.

Les caractéristiques d'une imprimante sont :

- **La vitesse d'impression** : nombre de pages tirées par minute
- **La résolution**: c'est le nombre de pixels que l'imprimante peut imprimer pour une unité de surface donnée (cette unité de surface est appelée point). La résolution se mesure donc en point par pixel (ppp) ou dot per inch (DPI)
- **La mémoire de l'imprimante** : mémoire qui se trouve dans l'imprimante ayant pour objectif de stocker les données en attente d'impression

## 5. Les lecteurs/graveurs de CD/DVD

Un lecteur est un appareil permettant la lecture, à l'aide d'un rayon laser, des informations enregistrées sur un CD-ROM ou un DVD-Rom. Un graveur est un appareil qui permet d'écrire des données sur un cd rom, à l'unité ou en petit série, par gravure au laser sur un CD ou DVD réinscriptible. Le CD-RW et le DVD-RW sont des disques optiques compacts sur lesquels on peut effacer et réinscrire jusqu'à 1000 fois des données.

La caractéristique principale d'un graveur est la vitesse d'exécution de la copie sur le cd ou dvd. Cette vitesse est mesurée en X. Exemple : 16x. X est le multiplicateur de la vitesse de gravure et correspond à 150ko/s et 1Mo/s pour les dvd.

## II. Conversion des unités de mesure informatique

Un bit est la plus petite unité d'information traitable par un ordinateur ; elle peut être 0 ou 1. Un octet est une suite de 8 bits ; en d'autres termes une combinaison de 0 et de 1 dont la quantité finale est huit.

L'octet est l'unité de mesure de la donnée ou de l'information en informatique. La capacité de tous les composants informatiques pouvant stocker se mesure en octet (o).

Les multiples de l'octet sont :

- **Le kilo-octet (ko) ou kibioctet (kio)** :  $1 \text{ ko} = 2^{10} \text{ o} = 1024 \text{ o}$
- **Le méga-octet (mo) ou mébioctet (mio)** :  $1 \text{ mo} = 2^{10} \text{ ko} = 2^{20} \text{ o}$
- **Le giga-octet (go) ou gibioctet (gio)** :  $1 \text{ go} = 2^{10} \text{ mo} = 2^{20} \text{ ko} = 2^{30} \text{ o}$
- **Le téra-octet (to) ou tebioctet (tio)** :  $1 \text{ to} = 2^{10} \text{ go} = 2^{20} \text{ mo} = 2^{30} \text{ ko} = 2^{40} \text{ o}$

**Exercice pratique:** Convertir 16384 bits en octets, ko, mo, go, to

### Résumé

La propriété principale de la mémoire est son volume (capacité) : elle s'exprime en octet ; la propriété principale du processeur est sa fréquence (en hertz) ; les propriétés principales de l'écran sont la taille (en pouce) et la définition (en ppm ou dpi).

### Exercices de consolidation

6)

### Classe : Troisième

### Module 2: Numération et Algorithme

### Chapitre 10: Exécuter un algorithme

### Leçon 22: Définition de concepts et notion d'algorithme

### Compétence :

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

- ✓ Enoncer les étapes de résolution d'un problème
- ✓ Elaborer une démarche de résolution d'un problème
- ✓ Définir les concepts algorithmiques
- ✓ Décrire le fonctionnement d'un ordinateur d'un point de vue algorithmique

## Situation-problème

Vous la mission d'indiquer le chemin à un étranger pour quitter d'un quartier à un autre à vélo. Quelques règles doivent être respectées : il ne doit plus se renseigner en cours de route, vous devez vous assurer qu'il comprend les termes que vous utilisez, vous devez être sûr qu'il arrivera à destination.

## Introduction

L'homme fait face chaque jour à une multitude de problèmes auxquels il se doit de proposer des solutions. la recherche de la solution à un problème est une démarche algorithmique, qui est loin d'être figée ou universelle, dans la mesure où elle dépend de la perception et de la conception de tout un chacun, cependant la solution reste unique.

### **I. Les étapes de résolution d'un problème**

La résolution d'un problème nécessite au moins trois étapes :

- **La collecte des données** : à partir du problème à résoudre, on essaye de le comprendre en profondeur, puis déterminer tous les mots clés nécessaires pour résoudre le problème (c'est ce qu'on appelle la collecte des données). **Exemple** : pour bouillir l'eau, il faut : une quantité d'eau, une marmite, un réchaud, ...
- **L'élaboration d'un plan de la solution** : A partir des données recensées, les résultats attendus et ses propres connaissances ; on construit un ensemble d'étapes permettant de résoudre le problème. On peut éventuellement décomposer une étape incompréhensible en sous-étapes. **Exemple** : pour bouillir l'eau il faut : verser l'eau dans la marmite, allumer le réchaud, ..., attendre que l'eau se mette à l'ébullition, ... ; l'étape allumer le réchaud peut être incompréhensible pour certain, on doit le décomposer en un ensemble de sous étapes.
- **La réalisation de la solution** : A partir du plan de la solution, de l'outil que l'utilisateur doit utiliser pour résoudre le problème, de ses propres connaissances ; on construit un système de résolution du problème utilisable par l'utilisateur. **Exemple** : Dans la première page vierge de recette de cuisine, on rédige dans l'ordre d'exécution l'ensemble des étapes à suivre pour bouillir l'eau ; on doit remplacer chaque étape compliquée par un ensemble de sous étapes élémentaires.

### **II. Notion d'algorithme**

Un algorithme est une suite finie et ordonnée d'instructions permettant de résoudre une classe de problèmes semblables en un temps fini.

Un algorithme doit être utilisé par un utilisateur pour résoudre son problème en l'absence du créateur de l'algorithme. C'est pourquoi le créateur doit connaître le langage de l'utilisateur et utiliser uniquement les instructions qu'il pourra comprendre. L'utilisateur peut être un homme, un ordinateur, une machine, ...

Un Langage de définition d'algorithme (LDA) est un langage naturel qui utilise un ensemble de mots-clés et de structure permettant de décrire de manière complète, claire, l'ensemble des opérations à exécuter sur des données pour obtenir des résultats. Ainsi pour écrire un algorithme, on doit utiliser un LDA propre à l'utilisateur. En d'autres termes chaque LDA doit être compréhensible par celui qui doit exécuter l'algorithme

Pour nous ; l'utilisateur est l'ordinateur ; nous devons donc utiliser un LDA compréhensible par l'ordinateur.

### **III. Le fonctionnement des ordinateurs**

Pour écrire et exécuter un algorithme, nous utiliserons le schéma fonctionnel de Von Neuman. Selon Von Neuman l'ordinateur est composé de quatre parties essentielles :

- **le processeur** : il reçoit du clavier les instructions, les données ; il stocke ces instructions et données mémoire, exécute l'algorithme (en accédant aux instructions et données en mémoire) et affiche le résultat à l'écran
- **la mémoire** : c'est le grenier du processeur : elle contient plusieurs cellules permettant au processeur de : stocker les instructions de l'algorithme ; exécuter l'algorithme en lisant ces instructions ; en fonction des instructions, stocker ou lire les données que l'utilisateur saisies au clavier.
- **les unités d'entrée** : elles permettent à l'utilisateur d'indiquer au processeur l'algorithme à exécuter, d'envoyer au processeur les instructions nécessaires pour exécuter l'algorithme. Dans le cadre de ce cours, nous utiliserons uniquement le clavier.
- **les unités de sortie** : elles permettent au processeur d'envoyer à l'utilisateur le résultat de l'exécution de l'algorithme. Dans le cadre de ce cours, nous utiliserons uniquement l'écran.

### Résumé

Les étapes de résolution d'un problème sont : la collecte de données, l'élaboration d'un plan de la solution, la réalisation de la solution. Un algorithme est une démarche à suivre pour résoudre un ensemble de problèmes qui se ressemblent. Un algorithme ne doit utiliser que des instructions compréhensibles par celui qui doit l'exécuter ou les opérations élémentaires.

### Exercice de consolidation

1. Quelles sont les étapes de résolution d'un problème ?
2. Qu'est-ce qu'un algorithme ?
3. Qu'est-ce qu'une instruction ?
4. Pour deux utilisateurs différents, doit-on écrire pour un même problème deux algorithmes différents ? Si oui pourquoi ?
5. Comment l'ordinateur fait-il pour exécuter un algorithme ?
6. Ecrire l'algorithme pour bouillir les œufs dans votre cuisine pour deux utilisateurs différents en sachant que le premier utilisateur connaît allumer et éteindre le feu dans votre cuisine, le second ne connaît pas.
7. Pourquoi les instructions d'un algorithme doivent être compréhensibles par un utilisateur ?
8. Pourquoi dit-on que pour un même algorithme, il existe une infinité de LDA ?
9. Comment fait-on pour communiquer avec l'ordinateur ? Comment l'ordinateur fait-il pour communiquer avec nous ?
10. Comment l'ordinateur effectue les calculs ? Où garde-t-il les résultats ?

### Classe : Troisième

#### Module 2: Numération et Algorithme

#### Chapitre 10: Exécuter un algorithme

#### Leçon 22: Variables et instructions

### Compétence :

- A la fin de cette leçon l'apprenant doit être capable de :
- ✓ Identifier les variables et leurs caractéristiques
  - ✓ Identifier les constantes
  - ✓ Identifier les instructions simples (Lecture, écriture, affectation)
  - ✓ Identifier une incrémentation/décrémentation

### Situation-problème :

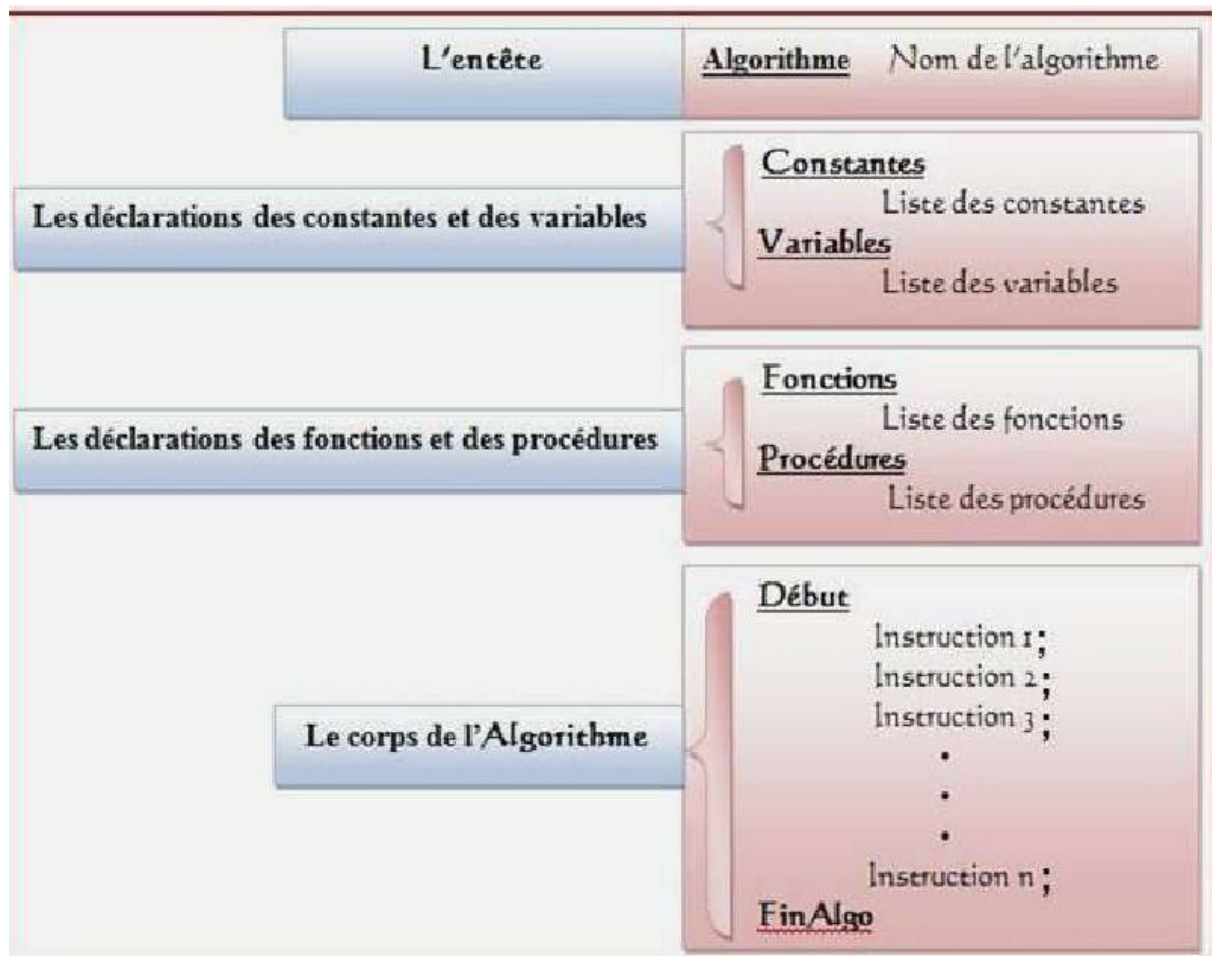
Je veux exécuter un algorithme capable de demander à l'utilisateur de saisir deux nombres au clavier ; puis mon algorithme affiche le résultat de la somme de ces deux nombres à l'écran. Que faire ?

### Introduction :

Un LDA est un langage de définition d'algorithme propre à un utilisateur. Or le processeur, cerveau de l'ordinateur ne peut utiliser qu'un LDA qu'il peut comprendre. De plus nous savons déjà que le processeur ne sait faire que des instructions arithmétiques simples (addition, soustraction, multiplication, division, reste d'une division), les instructions d'accès aux périphériques et à la mémoire. Quel est donc le LDA de l'ordinateur ? Cette leçon marque le début de l'étude du LDA de l'ordinateur.

#### **I. Structure d'un algorithme**

Un algorithme est structuré de façon suivante :



- **L'en-tête** permet tout simplement d'identifier un algorithme par un nom. Ce nom n'influence en rien le bon déroulement de l'algorithme. En général il faut donner des noms parlants à nos algorithmes, ceci pour permettre au lecteur d'avoir une idée de ce que fera l'algorithme;
- **Les déclarations de constantes, variables**, sont une liste exhaustive des objets ou des données utilisés et manipulés dans le corps de l'algorithme.
- **Les déclarations des fonctions et des procédures** sont une liste exhaustive des opérations non élémentaires c'est à dire qui contiennent d'autres instructions simples ou non. Les procédures et fonctions seront appelées dans

- **le corps de l'algorithme** : il contient les tâches (instructions, opérations) à exécuter. Ces tâches peuvent être des appels de fonction ou des instructions simples

## II. Déclaration des variables et constantes

La structure générale des algorithmes contient une zone de déclaration des variables et des constantes. Toutes les données manipulées par les algorithmes doivent être déclarées.

Une variable est une donnée qui peut être modifiée pendant l'exécution de l'algorithme. On peut représenter la variable par une boîte étiquetée pouvant contenir la donnée proprement dite. Cette donnée est d'un type que nous ne pouvons pas changer, mais nous pouvons changer sa valeur (le contenu de la boîte).

### 1. Déclaration des variables et constantes

La syntaxe de déclaration d'une variable (syntaxe minimale) est la suivante :

**Variable** : nom\_variable : type ;

Cette syntaxe contient :

- **Le nom de la variable** : comme le nom d'un être humain, il permet d'identifier la variable ; il doit commencer par une lettre
- **Le type de la variable** : il permet de préciser la qualité de données qu'on doit mettre dans la variable. On ne peut pas le changer dans la suite.

La syntaxe de déclaration de plusieurs variables est la suivante :

**Variable** : nom\_variable1 : type1, nom\_variable2 : type2 , ....., nom\_variableN : typeN ;

Une constante est une donnée dont la valeur ne peut pas être modifiée pendant l'exécution de l'algorithme. C'est une variable particulière : la valeur qui a été mise dans sa boîte reste la même pendant l'exécution de l'algorithme.

La syntaxe de déclaration d'une constante (syntaxe minimale) est la suivante :

**Constante**: nom\_constant= valeur ;

La syntaxe de déclaration de plusieurs constante est la suivante :

**Constante**: nom\_constant1=valeur1, nom\_constant2=valeur2, ....., nom\_constant3=valeur3;

### 2. Les types de données

On distingue cinq types de base :

- **Le type Entier** : On attribut le type Entier à une variable ou une constante destinée à recevoir un nombre entier naturel. **Exemple : variable** :  
reste\_division :Entier ;
- **Le type Réel** : : On attribut le type Réel à une variable ou une constante destinée à recevoir un nombre Réel. **Exemple : variable** :  
quotient\_division :Réel ;
- **Le type caractère** : On attribut le type Caractère à une variable ou une constante destinée à recevoir un caractère. **Exemple : variable** :  
première\_lettre :Caractère ;
- **Le type Chaîne\_caractère**: On attribut le type chaine\_caractère à une variable ou une constante destinée à recevoir un mot ou une chaîne de caractères. **Exemple : variable** : nom\_personne :Chaine\_caractère ;
- **Le type Booléen**: On attribut le type Booléen à une variable ou une constante pouvant recevoir juste deux valeurs (vrai ou faux). **Exemple : variable** :  
perdu:Booléen ;

## III. L'affectation et l'incrémention d'une variable

Dans le corps d'un algorithme, on trouve un ensemble d'instructions ordonnées permettant de résoudre le problème. Une instruction finit toujours par un point virgule.

L'affectation est une instruction qui permet de modifier la valeur d'une variable. Sa syntaxe est la suivante : nom\_variable ← valeur\_variable ;



**Exemple :** Exécuter cet algorithme

**Algorithme :** affecte2

**Variable :** nombre1 :Entier, nombre2 :Entier ;

**Début**

nombre1 ← 5 ;  
nombre2 ← 10 ;

**Fin**

L'incréméntation d'une variable permet d'augmenter la valeur 1 au contenu d'une variable de type Entier.

**Exemple :** Exécuter cet algorithme

**Algorithme :** affecte2

**Variable :** nombre1 :Entier, nombre2 :Entier, nombre3 :Entier ;

**Début**

nombre1 ← 5 ;  
nombre2 ← 10 ;  
nombre3 ← 15 ;  
nombre1 ← nombre1 + 1 ;  
nombre2 ← nombre2 + 1 ;  
nombre3 ← nombre3 + 1 ;

**Fin**

**NB :** Avant d'incrémenter une variable, on doit d'abord l'initialiser ; c'est-à-dire l'affecter une valeur pour la première fois

#### **IV. La lecture au clavier et l'affichage à l'écran**

Dès la première leçon, nous avons limité la communication entre l'ordinateur et son utilisateur par la lecture au clavier et l'affichage à l'écran.

##### **1. L'affichage à l'écran**

L'affichage à l'écran permet au processeur d'afficher une donnée à l'écran de l'utilisateur. Ainsi le processeur peut afficher un ordre que l'utilisateur doit exécuter, le résultat d'un problème ou un message quelconque.

La syntaxe minimale de déclaration d'une instruction d'affichage est la suivante :

**écrire(message) ;** message peut être une chaîne de caractères ou une variable.

La syntaxe générale de déclaration d'une instruction d'affichage est la suivante :

**écrire(message1, message2, ..., messageN) ;** Dès lors on peut afficher plusieurs messages à l'écran.

**Exemple :** Exécuter cet algorithme

**Algorithme :** affiche3

**Début**

écrire(« bienvenu en première ») ;  
écrire(« bienvenu en première ») ;  
écrire(« bienvenu en première ») ;

**Fin**

##### **2. La lecture au clavier**

La lecture est instruction permettant à l'utilisateur d'entrer une valeur au clavier. Ainsi le processeur peut affecter une valeur à une variable à partir des saisies effectuées au clavier par l'utilisateur.

La syntaxe minimale de déclaration d'une instruction de lecture est la suivante :

**lire(nom\_variable) ;** nom\_variable est le nom de la variable qui doit recevoir la valeur saisie.

La syntaxe générale de déclaration d'une instruction de lecture est la suivante :

**lire(nom\_variable1, nom\_variable2, ..., nom\_variableN )** ; Dès lors, on peut lire successivement plusieurs valeurs.

**NB :**

- 1) Dès que le programme rencontre une instruction Lire, l'exécution s'interrompt, attendant la frappe d'une valeur au clavier. Dès lors, aussitôt que la touche Entrée (Enter) a été frappée, l'exécution reprend.
- 2) Avant de Lire une variable, il est très fortement conseillé d'écrire des libellés à l'écran, afin de prévenir l'utilisateur de ce qu'il doit frapper ; sinon, l'utilisateur passe son temps à se demander ce que l'ordinateur attend de lui

**Exemple :** Exécuter cet algorithme

**Algorithme :** bienvenu

**Variable :** nom :Chaine\_caractère ;

**Début**

écrire(« entrer votre nom ») ;  
lire(nom) ;  
écrire(« bienvenu en première », nom) ;

**Fin**

**Résumé**

La structure minimale d'un algorithme est la suivante : l'entête, la déclaration des variables, la déclaration des constantes, la déclaration des fonctions, le corps de l'algorithme. Une variable est un objet informatique contenant une donnée pouvant être modifiée alors qu'une constante n'est pas modifiable. Les types de données que peut contenir une variable sont : le type entier, le type réel, le type caractère, le type chaîne de caractères et le type booléen.

**Exercices de consolidation**

- 1) Executer chacun de ces algorithmes

Variables A, B, C : chaines-caractères Début A ← "423" B ← "12" C ← A.B	Variables A, B, C : chaines- caractères Début A ← "423" B ← "12" C ← A & B	Variables A, B, C : Entier Début A ← 3 B ← 10 C ← A + B B ← A + B
Variables val, double : Entier ; Début Val ← 231 Double ← Val * 2 Ecrire (Val) ; Ecrire (Double) ;	Variables nb, pht, ttva, pttc : Entier Début Ecrire ("Entrez le prix hors taxes : " ) ; Lire (pht) ; Ecrire ("Entrez le nombre d'articles :") ; Lire (nb) ; Ecrire ("Entrez le taux de TVA :") ; Lire ( ttva) ; pttc ← nb * pht ; pttc ← pttc+ptva; Ecrire ("Le prix toutes taxes est : ", pttc ) ;	

**Classe : Troisième**

**Module 2: Numération et Algorithme**

**Chapitre 10: Exécuter un algorithme**

**Leçon 24 : h**

**Classe : première**

**Module 1 : Algorithme et programmation web statique**  
**Chapitre 1 : Les notions élémentaires de l'algorithme**  
**Leçon 4 : les structures alternatives**

**Compétence :**

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

- ✓ Utiliser une structure alternative
- ✓ Utiliser une structure de condition
- ✓ Utiliser la structure répétitive pour

**Situation-problème**

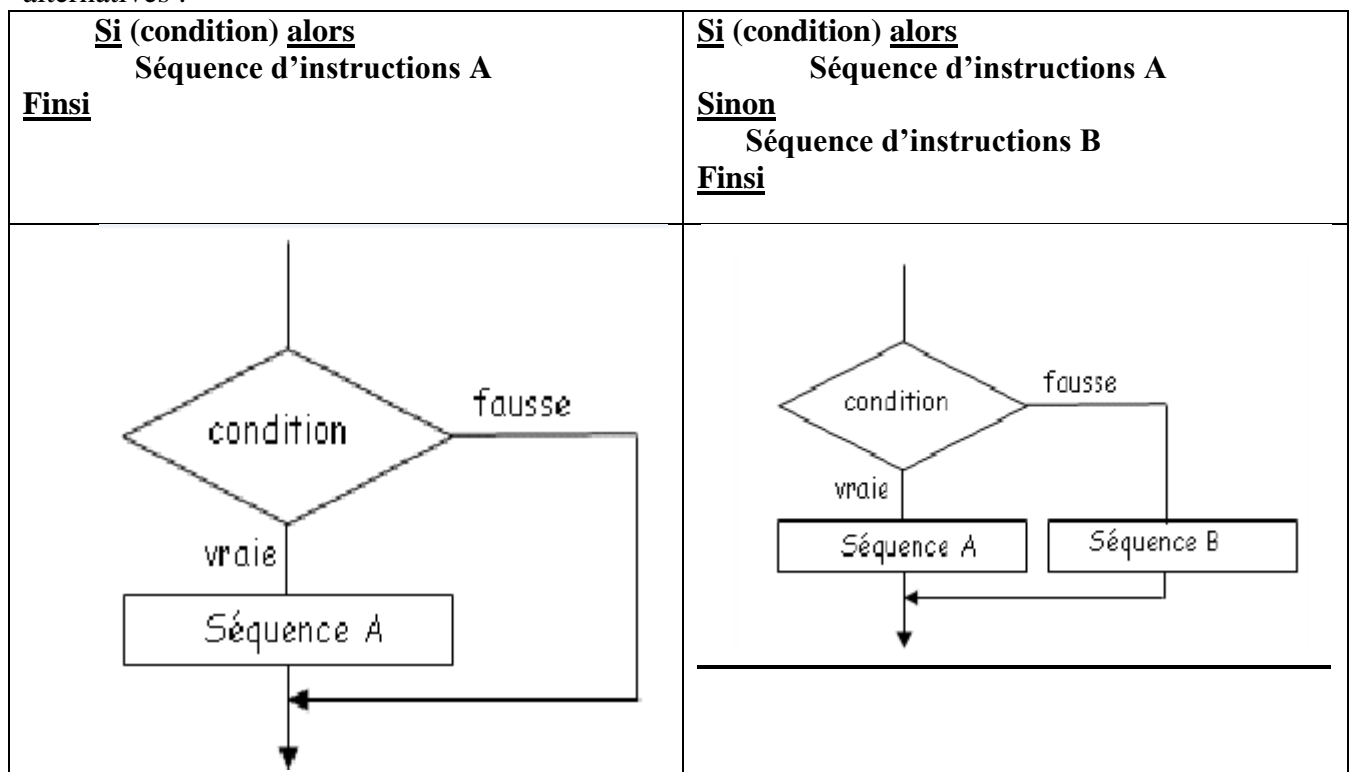
Je veux exécuter un algorithme qui permet à un élève de calculer le montant des cahiers qu'il a dans son sac. L'élève doit alors entrer le nombre de cahier (ce chiffre doit être positif), puis le montant de chaque cahier ; l'algorithme doit afficher à la fin le montant total.

**Introduction**

Dans le corps des algorithmes précédents, on résolvait les problèmes en écrivant une séquence d'instructions dont chacun se terminait par un point virgule ; on les exécutait l'une après l'autre (du premier jusqu'au dernier) : c'était des structures séquentiels. Dans cette leçon nous aborderons d'autres structures utilisées en algorithme.

**I. Les structures alternatives et les structures de condition**

Une structure alternative est une structure qui utilise une condition pour permettre d'exécuter une séquence d'instruction ou non. Il existe deux formes de structures alternatives :



La condition peut être une variable booléenne, une expression de comparaison ou une expression utilisant un ensemble d'opérateurs logiques.

Dans la première forme, si la condition est vrai, on exécute la séquence d'instructions A avant d'exécuter les instructions qui suivent le Finsi; sinon on saute directement la séquence d'instructions A pour exécuter les instructions qui suivent le Finsi. Dans la deuxième forme,

avant d'exécuter les instructions qui suivent le Finsi, on doit exécuter la séquence d'instructions A si la condition est vrai ; sinon on doit exécuter la séquence d'instructions B.

**Exemple :** Exécuter cet algorithme

**Algorithme :** vérifie\_racine

**Variable :** nombre :entier ;

**Début**

écrire(« entrer un nombre ») ;

lire(nombre) ;

**Si** (nombre $\geq$ 0) **alors**

écrire(« on peut calculer sa racine carrée ») ;

**Sinon**

écrire(« on ne peut pas calculer sa racine carrée ») ;

**Finsi**

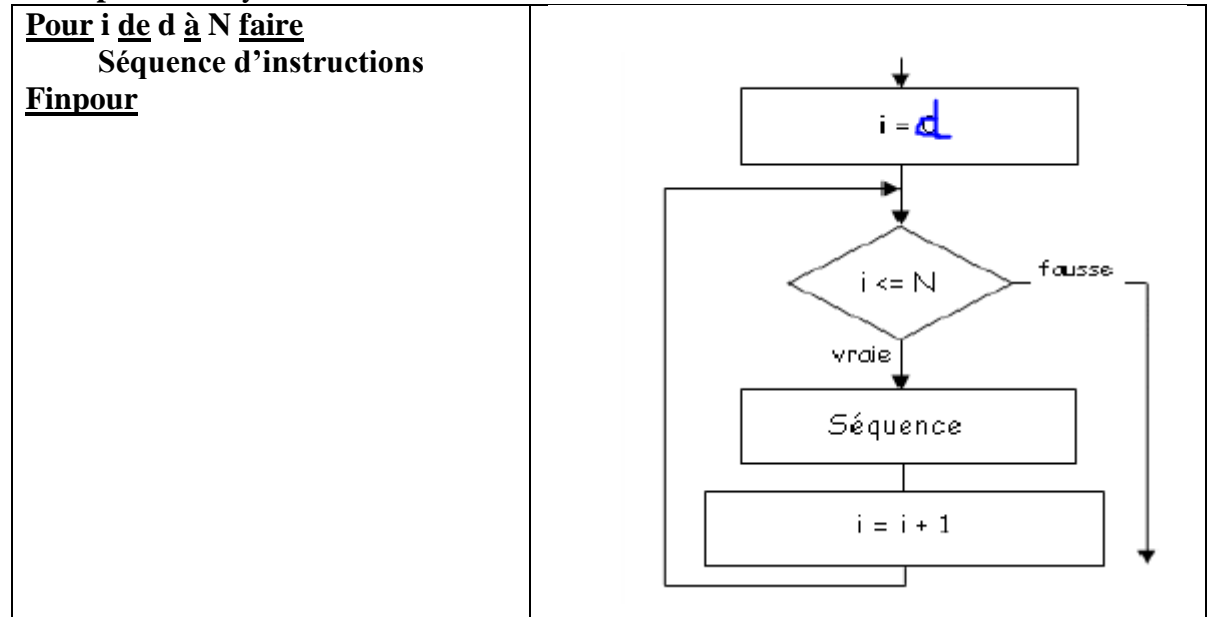
écrire(« terminé ») ;

**Fin**

## II. La structure répétitive pour

Une structure répétitive ou une boucle est une structure qui utilise une condition pour permettre d'exécuter plusieurs fois une séquence d'instructions.

**La boucle pour :** Sa syntaxe est la suivante :



La boucle pour commence par attribuer la valeur d à i ; puis si  $d \leq N$  alors elle exécute la séquence d'instructions, puis elle incrémente i et reprend le test ; si  $i > N$ , alors elle saute la séquence d'instructions pour exécuter l'instruction en dessous de Finpour.

**Exemple :** Exécuter cet algorithme

**Algorithme :** factoriel

**Variable** N, i, F: Entier ;

**Début**

$F \leftarrow 1$  ;

Ecrire ("Entrez un nombre compris entre 1 et 3") ;

Lire (N) ;

**Pour i de 1 à N faire**

$F \leftarrow F * i$  ;

**Finpour**

Ecrire (« Le factoriel de », N, « est », F) ;

Fin

### Résumé

Il existe plusieurs façons d'exécuter un ensemble d'instructions :

- **les structures séquentielles** : chaque instruction est exécutée une seule fois
- **les structures alternatives** : la première séquence est exécutée si la condition est vraie, sinon la seconde séquence est éventuellement exécutée
- **les structures répétitives (pour)** : en fonction de la condition, la séquence peut être exécutée plus d'une fois

### Exercice de consolidation