

Informatique

3^{ème} année de l'enseignement secondaire

Sections :

- Mathématiques
- Sciences expérimentales
- Sciences Techniques

Les auteurs

Rached Douari
Inspecteur Principal

Kamel Ben Rhouma
Maître Assistant à
l'ENSI

Fethi Hmida
Professeur Principal

Les évaluateurs

Moncef Gafsi
Maître Assistant à l'ENSI

Ridha Hadj Ali
Inspecteur d'informatique



Préface

Le présent manuel est conforme au nouveau programme d'informatique de 3^{ème} année des sections scientifiques à savoir la section Mathématiques, Sciences Expérimentales et Technique. Il est composé de deux parties. Chaque partie est divisée en chapitres. Certains chapitres ont été répartis en leçons. La leçon est composée d'un nombre de concepts et d'activités homogènes. Nous pensons que cette partition aidera le lecteur à se faire une représentation claire des différents objets d'apprentissage formant ce livre.

La première partie comporte trois chapitres. Le premier est intitulé « Introduction à l'informatique ». Nous y présentons l'intérêt, l'apport et l'impact de l'informatique dans les activités de la vie courante. Le second donne un aperçu sur l'architecture générale d'un ordinateur. Nous terminons cette partie par deux chapitres, l'un concernant les systèmes d'exploitation et leurs fonctions de base et le second concernant les réseaux informatiques et les services qu'ils offrent. Les apprentissages liés aux précédents chapitres seront appuyés par des activités d'ordre pratique et les cours devront se faire dans un laboratoire d'informatique.

La seconde partie est consacrée à un cours d'algorithmique et de programmation. Elle commence par un chapitre dont le rôle est de présenter les différentes étapes de résolution d'un problème. Nous avons jugé utile d'adopter une approche descendante dans l'analyse des problèmes à résoudre. Cette méthode est assez pédagogique et s'adapte bien au niveau des élèves surtout que les problèmes à résoudre sont relativement faciles et ayant des solutions claires.

Une fois la méthodologie présentée, nous passons au chapitre traitant les structures simples, les types standards et différentes représentations que leur fait la plupart des langages et particulièrement le langage Pascal. Rappelons que nous allons continuer à développer en Pascal pour plusieurs raisons de commodités, pédagogiques et formatives pour les enseignants. L'algorithmique et la programmation commence dans le chapitre réservé aux actions élémentaires d'entrée et de sortie. On y fait un rappel sur l'affection, ensuite l'action de sortie et enfin l'action d'entrée. Nous traitons ensuite la structure de contrôle conditionnelle avec sa forme simple c.à.d. l'alternative ensuite sa forme généralisée. Nous terminons ce livre par la structure de contrôle itérative et nous nous sommes restreints à l'itération complète.

Chaque concept est présenté à travers une activité permettant à l'élève de construire son savoir et de déduire lui-même le concept visé. Une brève récapitulation termine chaque leçon permettant au lecteur de retenir les points importants du cours. Nous proposons à la fin de chaque leçon ou chapitre une série d'exercices à difficulté croissante permettant aux lecteurs de s'exercer et de s'auto-évaluer.

Nous espérons que ce manuel vous apportera l'aide nécessaire pour réussir un bon apprentissage du programme correspondant. Toutefois, nous vous serons reconnaissants de bien vouloir nous faire part de vos suggestions et de vos remarques.

Les auteurs.



Sommaire

Préface

Sommaire

| | |
|-------------------|--|
| Chapitre 1 | Introduction à l'informatique |
| Chapitre 2 | Architecture d'un ordinateur |
| Chapitre 3 | Les systèmes d'exploitation |
| Chapitre 4 | Les réseaux informatiques |
| Chapitre 5 | Démarche de résolution de problèmes |
| Chapitre 6 | Les structures de données |
| Chapitre 7 | Les actions élémentaires simples |
| Chapitre 8 | Les structures de contrôle conditionnelles |
| Chapitre 9 | Les structures de contrôle itératives |

Bibliographie

Annexe



Chapitre 1

Introduction à l'informatique

Objectifs :

. Acquérir une culture
informatique

Plan du chapitre :

Leçon :
Introduction
à l'informatique

CHAPITRE 1



Leçon

Introduction à l'informatique

Objectifs spécifiques :

Acquérir une culture informatique

Plan de la leçon

I- Définitions

1. Introduction
2. Le mot "informatique"
3. Le mot "information"
4. Le mot "traitement"
5. Les dates clés de l'informatique

II- Domaines d'application

- Retenons
- Exercices

Leçon

Introduction à l'informatique

I- Définitions

I-1- Introduction

L'informatique est partout dans notre vie quotidienne, c'est un outil au service de très nombreux métiers. C'est aussi un domaine de recherche très dynamique.

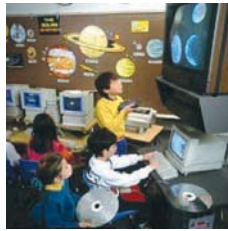
Apprendre l'usage d'un logiciel de traitement de texte ou d'un tableur relève par exemple du domaine de la bureautique, la création de sites web relève du domaine du multimédia et du graphisme.

Activité 1

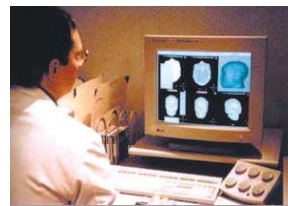
D'après les images ci-dessous, quel est l'outil informatique nécessaire et dans quel domaine pourrait-on être?



(A)



(B)



(C)



(D)

Fig.1 : Images illustrant des domaines d'application de l'informatique

| | Outil | Domaine |
|-----|-------|---------|
| (A) | | |
| (B) | | |
| (C) | | |
| (D) | | |

I-2- Le mot "Informatique"

I-2- 1 L'origine du mot "informatique"

Le mot informatique a été conçu en 1962 par le français Philippe Dreyfus qui cherchait à trouver un nom à une société dont il était l'un des créateurs. Cette société fut nommée SIA, Société d'Informatique Appliquée. Philippe Dreyfus s'est abstenu de déposer ce mot en tant que marque, lui permettant de devenir un nom commun.

I-2- 2 Définitions du mot informatique

a) D'après le dictionnaire "Robert" : " Science du traitement de l'information ; ensemble des techniques de la collecte, du tri, de la mise en mémoire, du stockage, de la transmission et de l'utilisation des informations traitées automatiquement à l'aide de programmes (logiciels) mis en oeuvre sur ordinateur."

b) D'après l'académie française : "Science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications dans les domaines techniques, économiques et sociaux."

c) D'après le dictionnaire de l'informatique et des sciences de l'information : "Science du traitement de l'information, elle ne se réduit pas à un recueil de recettes, de techniques et de savoir-faire, mais elle joue un rôle central dans de nombreuses disciplines scientifiques et dans la plupart des techniques contemporaines, son emprise sur notre environnement quotidien est de plus en plus grande."

I-3- Le mot "Information"

I-3- 1 Définition :

D'après le dictionnaire "Robert" : "Ensemble des actes qui tendent à établir la preuve d'une infraction. Renseignement sur quelque chose utile. Renseignement ou évènement qu'on porte à la connaissance d'une personne."

I-3- 2 Caractéristiques de l'information

Activité 2

Citez les principales caractéristiques de l'information. Sous quelle forme se trouve-t-elle et comment se traite-t-elle par ordinateur ?

Caractéristiques de l'information

L'information se caractérise par :

- la précision : elle doit être exacte et produite sans erreur.
- l'opportunité : elle doit parvenir au bon moment.
- l'intégralité : elle doit être complète.

- la concision : elle doit être homogène.
- la pertinence : elle doit conduire à une nouvelle connaissance de valeur.

L'information peut se présenter sous la forme :

- visuelle (texte, image fixe, tableau, dessin, graphique,...)
- auditive (séquence sonore, paroles, musique,....)
- audio visuelle (séquence animée, séquence vidéo,.....)

I-3- 3 L'unité de mesure de l'information

Pour pouvoir traiter l'information par ordinateur, celle ci doit être codée en un système numérique. Le bit (binary digit) est la plus petite unité de l'information, qui peut avoir la valeur 1 ou la valeur 0. Pourquoi ce choix ? . Regardez une lampe installée au laboratoire, cette lampe ne peut avoir que deux états : allumée ou éteinte. Au repos, la lampe est éteinte et son interrupteur dans une position initiale (position qui prend la valeur zéro). Pour l'allumer, on doit changer la position de l'interrupteur, (position qui prend la valeur 1). Les deux états donnent en fait deux informations. Si on a deux lampes et deux interrupteurs, on aura quatre états suivant le tableau ci-dessous.

| Etat | Lampe 1 | Lampe 2 |
|--------|---------|---------|
| Etat 1 | 0 | 0 |
| Etat 2 | 0 | 1 |
| Etat 3 | 1 | 0 |
| Etat 4 | 1 | 1 |

Le nombre d'états avec deux lampes est égal à $4 = 2^2$.

Activité 3

Déterminez les différents états dans le cas où nous avons trois lampes et trois interrupteurs.

I-4- Le mot "Traitement"

I-4- 1 Présentation

Il s'agit de collecter et d'enregistrer les données puis les utiliser directement ou d'appliquer des opérations de transformation telles que la classification, le tri, le calcul arithmétique, la synthèse des données. Le traitement est présenté à la machine sous forme de code et permet d'aboutir aux résultats souhaités. Ces derniers peuvent être communiqués ou stockés sur des mémoires.

I-4- 2 Définition

D'après le dictionnaire "Robert" : Déroulement systématique d'une suite d'opérations logiques et mathématiques effectuées par des moyens automatiques sur des données pour les exploiter selon un programme.

I-4- 3 Nature du traitement

Les données peuvent être traitées de plusieurs façons :

- la collecte des informations
- l'entrée des informations à l'ordinateur (saisie, numérisation, acquisition,...)
- l'enregistrement de ces informations.
- le traitement (calcul, tri, recherche, ...)
- l'exploitation des résultats, Les stockages,...

I-5- Les dates clés de l'informatique

Même si le principe de fonctionnement n'a pas changé, la technologie de fabrication des ordinateurs est passée par plusieurs générations successives, correspondant à des innovations majeures dans l'évolution du matériel et du logiciel résultant des nouvelles technologies électroniques. Voici une description chronologique de l'histoire de l'informatique classée par génération.

I-5- 1 Le première génération (1946)

Le premier ordinateur entièrement numérique, conçu en 1946 à l'université de Pennsylvanie, l'ENIAC (Electronic Numerator Integrator and Computer), fut inventé par deux ingénieurs américains John Presper Eckert (1919-1995) et John William Mauchly (1907-1980). Il contenait 18000 lampes à vide et pesait trente tonnes. Il est également le premier à traiter l'information aussi bien textuelle que numérique. Sa programmation nécessitait une intervention manuelle humaine importante. Le langage de programmation est en binaire.

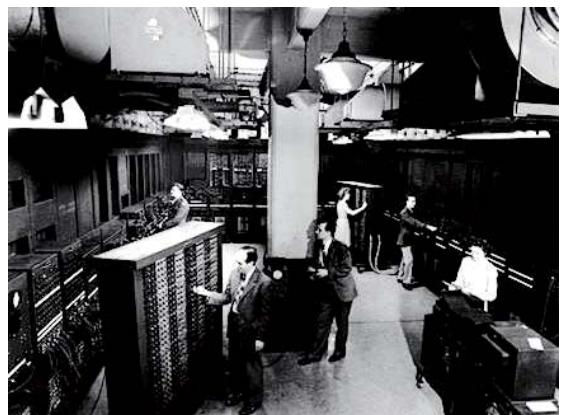


Fig. 2 : Calculateur ENIAC

I-5-2 La deuxième génération (1956)

Les lampes ont été remplacées par des transistors. Les langages de programmation ont fait leur apparition : l'assembleur puis le Fortran en 1956, Cobol en 1960 et Algol en 1960.

I-5- 3 La troisième génération (1964)

Les progrès de la technologie permettent de graver un nombre croissant de circuits électroniques pour un volume donné de silicium. C'est l'apparition des fameux circuits intégrés avec une échelle d'intégration de plus en plus poussée et la naissance d'ordinateur central d'entreprise conçu pour la gestion et le calcul numérique (IBM 360, CDC 6000, mini-ordinateur,...). D'autres langages de programmation utilisés pour la gestion et le calcul ont vu le jour. Les plus connus sont : Basic (1964), Apl (1969), Pascal (1969) et C (1972).

I-5- 4 La quatrième génération (1976)

L'intégration de plus en plus poussée donne naissance aux microprocesseurs et à la micro-informatique qui met l'informatique à la portée de tous. Cela a favorisé la naissance des micro ordinateurs individuels de bureau (Apple II, IBM PC,...) utilisés pour la bureautique (traitement de textes, tableur,...).

I-5- 5 La cinquième génération (1984)

Cette génération est caractérisée par l'augmentation des performances des processeurs (fréquence 8 MHz). Nous avons vu la naissance des micro-ordinateurs familiaux et l'apparition des systèmes d'exploitation MSDOS, MacOs,..ainsi que les réseaux locaux.

I-5- 6 La sixième génération (1992)

Des processeurs avec une fréquence égale à 33 MHz et une RAM de 640 Ko à 4 Mo ont vu le jour. Nous avons ainsi vu la naissance de logiciels pour : le Web, la messagerie électronique, le multimédia et les jeux 3D utilisés pour le traitement d'image, du son et de la vidéo.

I-5- 7 La génération actuelle

Elle est caractérisée par une plus grande échelle d'intégration et une nouvelle élévation de la fréquence d'horloge des processeurs. Cette nouvelle technologie donnent naissance à des ordinateurs portables très puissants avec des entrées/Sorties vocales et des réseaux sans fils.

II- Domaines d'applications

Activité 4

Aidé par votre professeur, remplissez le tableau ci-dessous où vous indiquez les domaines d'application d'informatique.

| à la maison | à l'école | au magasin |
|-------------|-----------|------------|
| | | |
| | | |
| | | |

Presque dans chaque secteur d'activité, l'outil informatique est devenu essentiel. Son utilisation est devenue indispensable à la maison, à l'école et au travail. Parmi ces domaines, on distingue :

- 1) **L'informatique de gestion** : Elle s'intéresse à tout ce qui est gestion des entreprises.
Exemples : Gestion des élèves, gestion du personnel, gestion du stock, ...
- 2) **La bureautique** : Permet l'élaboration des documents de différents types.
Exemples : Traitement de textes, calcul numérique, ...
- 3) **La communication** : Elle assure l'échange des données entre des ordinateurs. Dans ce cas, on utilise des réseaux informatiques.
- 4) **La médecine** : Elle est utilisée pour faire le suivi des patients ainsi que les diagnostics des maladies.
- 5) **L'informatique industrielle** : Elle permet la conduite automatique de certaines machines dans le domaine industriel grâce à des programmes (la robotique). En informatique industrielle on trouve
 - La CAO : La Conception Assistée par Ordinateur
 - Le DAO : Le Dessin Assisté par Ordinateur
 - La MAO : La Maintenance Assistée par Ordinateur.
- 6) **L'EAO** : L'Enseignement Assisté par Ordinateur.
- 7) **La linguistique** : Permet la traduction d'un texte d'une langue vers une autre.
- 8) **Les loisirs** : Utilisation des logiciels de jeux.

Retenons

L'informatique est, d'une part une science, et d'autre part un ensemble de techniques matérielles et logicielles mises en oeuvre par cette science.

L'information traitée par ordinateur peut être sous la forme visuelle (texte, dessin, image,..), auditive ou les deux à la fois.

- L'octet est une unité de l'information.

Exercices

Exercice 1

Citez les avantages d'utilisation de l'informatique dans l'enseignement.

Exercice 2

Donnez des exemples d'applications de l'informatique

- à l'aéroport
- chez le médecin
- dans les banques
- au supermarché
- dans l'industrie

Exercice 3

Que pensez vous de la définition suivante tirée du dictionnaire LE MEXIDICO : "l'informatique est d'une part une science et d'autre part un ensemble de techniques".

Chapitre 2

Architecture d'un micro-ordinateur

Objectifs :

. Identifier les principaux composants d'un micro-ordinateur et connaître leurs caractéristiques.

Plan du chapitre :

Leçon 1 :
Les principaux composants d'un micro ordinateur

Leçon 2 :
Les périphériques et les supports de stockage

Leçon 1

Les principaux composants d'un micro-ordinateur

Objectifs spécifiques

Identifier les principaux composants d'un micro-ordinateur et connaître leurs caractéristiques.

Plan de la leçon

I- Définition d'un ordinateur

1. Définition d'un ordinateur
2. Les différentes catégories d'ordinateurs

II- Les composants d'un ordinateur de bureau

1. L'unité centrale
2. Les périphériques

III- Les composantes de la carte mère

1. Les critères de choix des cartes mères
2. Les composants de la carte mère

Retenons

Exercices

Leçon 1

Les principaux composants d'un micro-ordinateur

I- Définition d'un ordinateur

Activité 1

Suite à une visite au SIB (Salon International de l'informatique et de la Bureautique), on nous a proposé les trois offres ci-dessous. En discutant entre vous et avec votre professeur et en se référant aux offres, essayez d'identifier chaque produit puis trouvez les composants communs.




| Offre 1 | Offre 2 | Offre 3 |
|--|---|--|
|  <p>Intel Pentium 4, 3 GHz Disque Dur 120 GO/7200 tpm RAM 256 Mo DDRAM Combo DVD / CDRW Contrôleur graphique Ext a 64 Mo Carte Fax Modem V92 56 KBPS Clavier Bilingue - souris avec roulette Ecran CRT HP 17" Win XP Home + Abonnement Internet</p> |  <p>Processeur Samsung 266 Mhz (2410) RAM 64 Mo Ecran 3,5 " TFT Ecran tactile et stylet Emplacement SD intégré Carte SD en option Bluetooth Microphone, haut parleur, prise d'écouteur (Lxpxh) 69,8 x 12,8 x 13,3 mm Windows Pocket PC 2003 Professionel Poids 124 g</p> |  <p>Intel Pentium M 340 1,5 GHz Mémoire Cache 512 Ko RAM 256 Mo Disque Dur 40 GO Graveur DVD/RW Dual Lan / Modem / 3ports USB / parallèle Infrarouge- PCMCIA- FireWire Ecran 15" XGA TFT WinXp home-abonnement Internet poids 2,84 Kg</p> |
| Lecteur Disquette 16,5000 TTC | Disque détachable USB 2 128 Mo 35,000 TTC | Disque dur 40 Go 85,000 TTC |
| Lecteur DVD 16X 45,000 TTC | Disque détachable USB 2 256 Mo 50,000 TTC | Disque dur 80 Go 90,000 TTC |
| Graveur DVD 12X 150,000 TTC | Disque détachable USB 2 512 Mo 110,000 TTC | Disque dur 120 Go 120,000 TTC |

Fig.1: Trois types de micro-ordinateurs

I-1- Définition d'un ordinateur

D'après le dictionnaire Robert : "Un ordinateur est une grosse calculatrice électronique dotée de mémoires à grande capacité, de moyen de traitement des informations à grande vitesse, capable de résoudre des problèmes arithmétiques et logiques complexes grâce à l'exploitation automatique de programmes enregistrés dans sa mémoire."

I-2- Les différentes catégories d'ordinateurs

I-2- 1 Les super-calculateurs (Fig. 2) : Ils sont très puissants et ont une grande capacité de stockage. Ils sont utilisés pour les grandes applications tels que : les besoins de la recherche scientifique, la gestion de grandes entreprises,...

Exemple : *Le super-calculateur Cray-1 fut le premier ordinateur capable d'effectuer 100 millions d'opérations par seconde. Ce type de calculateur est utilisé en particulier en météorologie et pour des calculs à grande échelle.*



Fig. 2

I-2- 2 Les mini-ordinateurs : Ils sont de taille et de capacité moyennes, comparés aux super-calculateurs. Ils sont conçus pour des applications spécialisées comme le contrôle de machines industrielles complexes. Ils ont été remplacés actuellement par les micro-ordinateurs.

I-2- 3 Les ordinateurs personnels (PC) (Fig.3) : Un ordinateur de bureau est composé généralement d'une unité centrale et de périphériques. Il se caractérise par son prix et par sa grande facilité d'utilisation, ce qui lui permet d'être utilisé dans tous les domaines d'application informatique.



Fig. 3

I-2-4 Les ordinateurs portables (Fig.4) : C'est un ensemble d'éléments logés dans un boîtier de petite dimension, intégrant l'écran, le clavier, l'unité centrale et une batterie. Il a l'avantage d'être opérationnel n'importe où. Son prix est généralement plus élevé qu'un ordinateur de bureau.



Fig. 4

I-2-5 Les ordinateurs de poche (Fig.5) : La miniaturisation des ordinateurs a permis aussi de réaliser des ordinateurs de poche tenant dans la main, tout en ayant une grande puissance. Les plus performants de ces ordinateurs possèdent des disques durs d'une capacité de stockage égale à plusieurs giga-octets et disposent d'un ensemble de ressources matérielles et logicielles similaires à celles des ordinateurs de bureau, ainsi que des moyens de connexion à des réseaux comme Internet.



Fig. 5

II- Les composants d'un ordinateur de bureau

Activité 2

Observez le matériel dans la salle puis dégagez les principaux composants du micro-ordinateur que vous utilisez.
Vous remarquerez qu'un ordinateur se compose d'une unité centrale et de périphériques.



II-1- L'unité centrale (Fig.6)

Activité 3

Ouvrons le boîtier et découvrons l'intérieur de l'unité centrale.
On y trouve : une boîte d'alimentation, une carte mère à laquelle sont reliées par des nappes des périphériques tels que : lecteur de disquettes, lecteur de CD-ROM,...

Fig. 6

II-1-1 Le boîtier : C'est un boîtier métallique ou en plastique contenant les différents composants internes. Les éléments principaux d'un boîtier sont caractérisés par les formes, les dimensions, le nombre d'emplacements pour les lecteurs, l'alimentation, la connectique en façade ainsi que son "design" et ses couleurs.

II-1-2 L'alimentation (Fig.7) : Son rôle est de convertir le courant du secteur de 220 volts en un courant continu de 5 et 12 volts avec une puissance qui dépasse actuellement les 250 watts pour alimenter les différents éléments internes et certains périphériques externes du micro-ordinateur.



Fig. 7

II-1- 3 Les nappes (Fig.8) : Ces nappes relient les disques et les différents lecteurs aux contrôleurs intégrés à la carte mère.



Fig. 8

II-1- 4 Les voyants de la face avant : Ils indiquent la mise sous tension de l'unité centrale et l'activation du disque dur. Le bouton de remise à zéro ("Reset") et le verrouillage du clavier sont reliés par de petits fils à la carte mère.

II-1- 5 La carte mère (en anglais motherboard) (Fig.9) : C'est une carte électronique comportant un circuit imprimé et parcourue de circuits, sur laquelle sont fixés le processeur, la mémoire et d'autres composants électroniques (chipset, bios,...). Elle dispose d'une série de connecteurs où on peut brancher divers périphériques.



Fig. 9

II-2- Les périphériques

Définition : Un périphérique est tout matériel pouvant être raccordé à un ordinateur par l'intermédiaire de l'un de ses ports (série, parallèle, USB, firewire, SCSI, ...). On distingue les catégories de périphériques suivantes :

- Les périphériques d'entrée (clavier, souris, lecteur CD-ROM,...)
- Les périphériques de sortie (écran, imprimante,...)
- Les périphériques d'entrée / sortie (lecteur de disquettes, modem,...)



Fig. 10

III- les composants de la carte mère

III-1- Les critères de choix des cartes mères

Le choix de la carte mère conditionne le fonctionnement de tout l'ordinateur. Cette carte contient des éléments intégrés : le chipset, les bus, les ports, l'horloge, la pile du CMOS, le bios,... Elle permet aussi d'interconnecter tous les autres composants : le processeur, la mémoire, le disque dur, la carte graphique,

Il existe plusieurs façons de caractériser une carte-mère : son facteur d'encombrement, son chipset, son type de support de processeur et le nombre de connecteurs de chaque type.

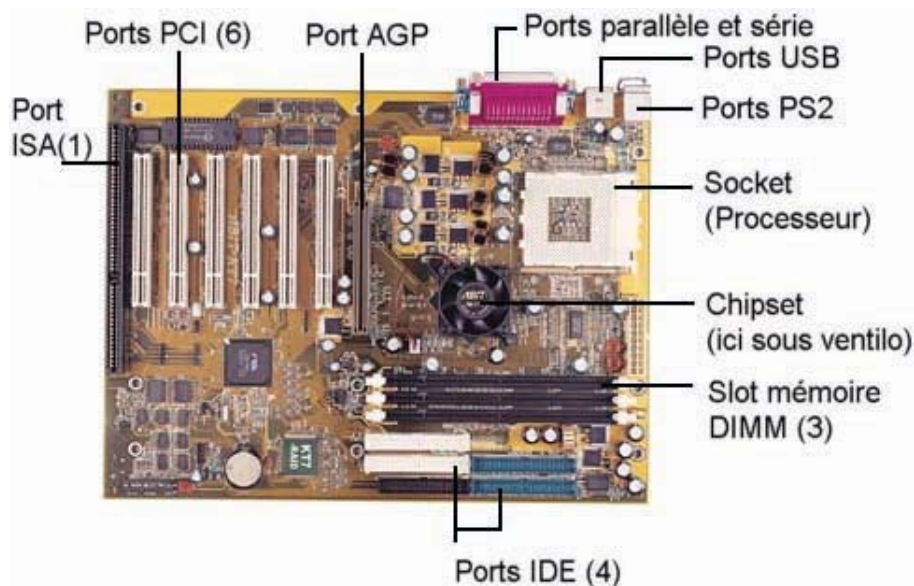


Fig. 11

III-2- Les composants de la carte mère

III-2-1 Chipset (Fig.12) : Le chipset (en français jeu de composants) est un circuit intégré chargé d'orienter les informations entre les différents bus de l'ordinateur afin de permettre à tous les éléments constitutifs de l'ordinateur de communiquer entre eux.



Fig. 12

III-2-2 La Rom Bios (Fig.13) : La rom bios ("Basic Input/Output System" traduisez "Système d'entrées/sorties basique" est utilisé pour permettre le contrôle du matériel. Le BIOS est un composant essentiel de l'ordinateur, il s'agit d'un petit logiciel dont une partie est dans une ROM (mémoire morte, c'est-à-dire une mémoire qui ne peut pas être modifiée), et une autre partie est dans un EEPROM (mémoire modifiable par impulsions électriques, d'où le terme flasher pour désigner l'action de modifier l'EEPROM).



Fig. 13

III-2-3 La mémoire CMOS (Complementary Metal Oxide Semiconductor) : Ce type de mémoire permet notamment de conserver les données nécessaires au démarrage de l'ordinateur (date, heure, mot de passe, ...). Elle est de faible taille et elle est alimentée par une pile intégrée à la carte mère.

III-2-4 Les bus : Un bus est un ensemble de liaisons physiques (câbles, pistes de circuits imprimés, ...) pouvant être exploitées en commun par plusieurs éléments matériels afin de communiquer. Un bus sert à véhiculer les informations entre les différents éléments de l'ordinateur.

Il existe différents types de bus : le bus processeur et le bus d'adresse. Le bus processeur a pour rôle de transférer les données du processeur vers la mémoire centrale et inversement. Les bus d'adresses sont dédiés comme leur nom l'indique à l'échange d'adresses entre les différents éléments de l'ordinateur.

III-2-5 La mémoire Rom : C'est l'acronyme de (Read Only Memory, ou en français mémoire à lecture seule), appelée aussi mémoire morte, parfois mémoire non volatile car elle ne s'efface pas lors de la mise hors tension de l'ordinateur.

III-2-6 Les ports d'entrée-sortie : Un port sert à connecter des périphériques à l'ordinateur. Il existe plusieurs types dont nous allons découvrir quelques uns dans l'activité qui suit :

Activité

Regardez la face arrière de votre ordinateur et vérifiez où peut-on connecter les périphériques suivants : la souris, le clavier, l'écran, le câble réseau, l'imprimante, ... (Fig.14).

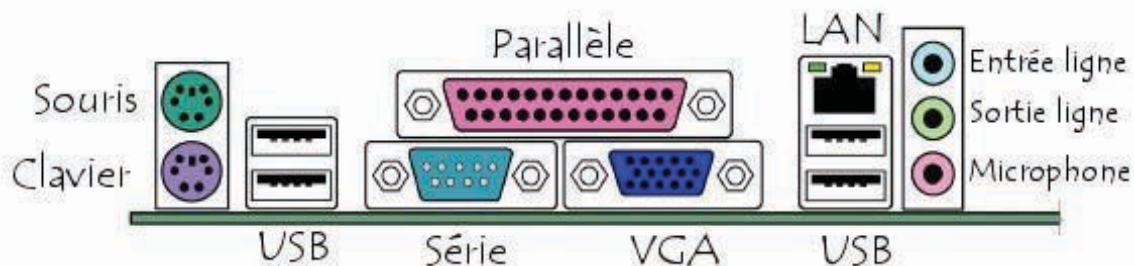


Fig. 14

- 1) **Les ports séries dédiés (PS / 2)** : Ce sont deux ports dédiés pour la souris et le clavier.
- 2) **Les ports séries classiques** (Fig.15 et Fig.16) : Les ports séries (également appelés RS-232) représentent les premières interfaces ayant permis aux ordinateurs d'échanger des informations avec le "monde extérieur". Le terme série désigne un envoi de données via un fil unique : les bits sont envoyés les uns à la suite des autres.

Les ports séries sont généralement intégrés à la carte-mère. Les connecteurs séries possèdent généralement 9 ou 25 broches et se présentent sous les formes suivantes. (connecteurs DB9 et DB25).



Fig. 15. Port série DB 9

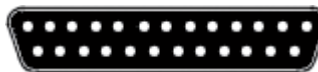


Fig. 16. Port série DB25

- 3) **Les ports USB** : Le bus USB (Universal Serial Bus, en français bus série universel) est basé sur une architecture de type série. Il s'agit toutefois d'une interface entrée-sortie beaucoup plus rapide que les ports séries standards. Il a été élaboré pour la connexion de plusieurs périphériques. Il existe deux types de connecteurs USB :

- Les connecteurs de type A (Fig.17), de forme rectangulaire, utilisés pour des périphériques à faible débit (clavier, souris, webcam, ...).

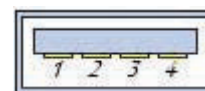


Fig. 17

- Les connecteurs de type B (Fig.18), de forme carrée, utilisés pour des périphériques à haut débit (disques durs externes, scanner,...).

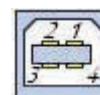


Fig. 18

La vitesse de transmission des données pour la norme USB 1.0 peut atteindre 12 Mbits/s en mode haut débit et 1.5 Mbits/s à faible débit tandis que la norme USB 2.0 peut atteindre jusqu'à 480 Mbits/s. Le grand avantage est la connexion à chaud.

4) **Les ports parallèles** (Fig.19) : Les données sont transmises en parallèle simultanément sur plusieurs fils. Les ports parallèles présents sur les ordinateurs personnels permettent d'envoyer simultanément 8 bits (un octet) par l'intermédiaire de 8 fils. Ils sont généralement intégrés à la carte-mère. Les connecteurs DB25 permettent de connecter un élément extérieur (imprimante par exemple).



Fig. 19. Port parallèle DB25

5) **Le port Firewire (norme : IEEE 1394)** : Il permet de connecter des périphériques (tels que les caméras numériques et autres) et de faire circuler des données à très haut débit (entre 100 et 400 Mbits/s pour la norme IEEE 1394a) et en temps réel. Les nouvelles configurations d'ordinateurs comportent généralement un ou deux ports firewire. Il existe ainsi des cartes d'extension (généralement au format PCI) permettant de doter un ordinateur de connecteurs Firewire. Les connecteurs et câbles Firewire sont repérables grâce à leur forme. La norme IEEE 1394 a défini deux types de connecteurs :

Les connecteurs 1394a-2000 utilisés sur les caméras numériques (Fig.20)



Fig. 20

Les connecteurs 1394a-1995 (Fig.21)



Fig. 21

6) **Le port Bluetooth** (Fig.22) : Le Bluetooth est une technologie de réseau personnel sans fil (noté WPAN pour Wireless Personal Area Network). Il a été mis au point par Ericsson en 1994. Il permet de relier des périphériques (imprimantes, téléphones portables, souris, clavier, etc.) et des ordinateurs entre eux de l'ordre de quelques dizaines de mètres, en utilisant les ondes radio (d'une fréquence de 2.4 GHz). Les périphériques ne doivent pas être obligatoirement en liaison visuelle pour communiquer. La norme IEEE 802.15.1 définit le standard Bluetooth 1.x permettant d'obtenir un débit de 1 Mbit/s.



Fig. 22

7) **Le port infrarouge** : la liaison infrarouge permet une communication sans fils et à une distance de quelques mètres avec une débit pouvant atteindre quelques mégabits par seconde. Cette technologie est largement utilisée pour la domestique (télécommandes) mais souffre toutefois des perturbations dues aux interférences lumineuses.

III-2-7- Le processeur

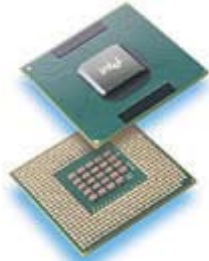


Fig. 23



Fig. 24

1) **Présentation** (Fig.23) : L'unité centrale de traitement (CPU : Central Processing Unit), encore dénommée processeur ou microprocesseur, est l'élément de l'ordinateur qui interprète et exécute les instructions d'un programme. Il est cadencé au rythme d'une horloge fixée sur la carte mère et dont la fréquence dépasse de nos jours les trois Giga-Hertz.

2) **Le premier processeur Intel** (Fig.24) : Le premier microprocesseur (Intel 4004) a été inventé en 1971. Il s'agissait d'une unité de calcul de 4 bits, cadencé à 108 kHz. Depuis, la puissance des microprocesseurs augmente exponentiellement.

III-2-8- La mémoire

1) **Présentation** : La mémoire est un ensemble de circuits intégrés capable de stocker temporairement des données. Elle est caractérisée par sa capacité (exprimée en méga-octets) et son temps d'accès (exprimé en nanosecondes). On distingue plusieurs types de mémoire.

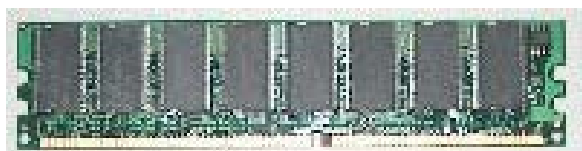
2) **La mémoire RAM** : La mémoire centrale d'un ordinateur est composée de circuits appelés mémoires vives (RAM : Random Acces Memory). C'est une mémoire dans laquelle doit être logé un programme en cours d'exécution ainsi que les variables s'y afférant. Une bonne partie du système d'exploitation y est installée dès le lancement de la machine, elles deviennent à accès aléatoire rapide. Plus la mémoire est rapide et plus vite le processeur est servi. C'est une mémoire volatile; elle devient vide hors tension. Elle se présente sous forme de petites cartes d'extension (appelées "Barrettes de mémoire").

Retenons

- Un micro-ordinateur est une machine qui assure le traitement automatiquement de l'information à l'aide de programmes.
- Chaque ordinateur est composé d'une unité centrale et de périphériques.
- On distingue les périphériques d'entrée, les périphériques de sortie et les périphériques d'entrée/sortie.
- Le boîtier renferme l'unité centrale, l'alimentation et des périphériques internes.
- La carte mère est une carte électronique parcourue de circuits, sur laquelle sont fixés la mémoire, le processeur, la Rombios,..
- Les ports sont des circuits permettant de connecter des périphériques à l'unité centrale.
- Le processeur est le cerveau de l'ordinateur. Il regroupe plusieurs unités tels que l'unité arithmétique et logique.
- Il existe des mémoires volatiles appelées RAM (Fig.25) et des mémoires non volatiles appelées ROM.



Fig. 25 mémoire SDRAM



et mémoire DIMM

Exercices

Exercice 1

Remplissez le tableau suivant en attribuant à chaque élément sa liaison avec la carte mère soit par câble, soit par une nappe et soit directement.

| Éléments | Liaison |
|-----------------------|---------|
| Disque dur | |
| Boîte d'alimentation | |
| Lecteur de disquettes | |
| Haut parleur interne | |

Exercice 2

Tournez le boîtier de votre ordinateur, puis identifiez les ports disponibles.

Exercice 3

Observez l'avant de votre ordinateur puis classez ses périphériques par catégorie dans le tableau si-dessous.

| Périphériques externes | Périphériques internes |
|------------------------|------------------------|
| | |
| | |
| | |

Leçon 2

Les périphériques et les supports de stockage

Objectifs spécifiques

- Identifier les principaux périphériques d'un micro-ordinateur et connaître leurs caractéristiques.
- Identifier les principaux supports de stockage d'un micro-ordinateur et connaître leurs caractéristiques.

Plan de la leçon

I- Les périphériques

1. Présentation
2. Les périphériques d'entrée
3. Les périphériques de sortie
4. Les périphériques d'entrée / sortie

II- Les supports de stockage

1. Présentation
2. Les disquettes
3. Le disque dur
4. Le flash disque
5. Le CD-ROM / CD-R / CD-RW
6. Le DVD
7. La bande magnétique
8. La carte mémoire (la mémoire flash)

Retenons

Exercices

Leçon 2

Les périphériques et les supports de stockage

I- Les périphériques

I-1- Présentation

Un périphérique est tout matériel pouvant être raccordé à un ordinateur par l'intermédiaire de l'un de ses ports (série, parallèle, USB, firewire, SCSI, ...). On distingue les catégories de périphériques suivantes :

- **Les périphériques d'entrée** : Ils sont des périphériques capables d'envoyer des informations à l'unité centrale de l'ordinateur. *Exemple (la souris, le clavier,...).*
- **Les périphériques de sortie** : Ils sont capables d'assurer la sortie des informations à partir de l'unité centrale . *Exemple (l'écran, l'imprimante, ...).*
- **Les périphériques d'entrée-sortie** : Ils sont capables d'assurer la sortie des informations à partir de l'unité centrale et d'en recevoir aussi . *Exemple (le lecteur de CD-ROM, le lecteur de DVD-ROM, Le Modem, ...).*

I-2- Les périphériques d'entrée

I-2-1 Le clavier (Fig.1) : Le clavier (en anglais keyboard) est le périphérique d'entrée principal, Il sert à saisir des caractères (lettres, chiffres, symboles ...). Il est généralement branché à l'arrière de l'unité centrale à travers un port série dédié à la carte mère.



Fig. 1

I-2- 2 La souris (Fig.2) : La souris (en anglais mouse) est un périphérique de pointage (en anglais pointing device) sert à déplacer un curseur sur l'écran et permettant de sélectionner, déplacer, manipuler des objets grâce à des boutons. On appelle ainsi « clic » l'action consistant à appuyer sur un bouton afin d'effectuer une action. il est généralement branché à l'ordinateur à travers un port série avec ou sans fil.



Fig. 2

I-2- 3 Le scanner : Un scanner (ou numériseur) est un périphérique d'entrée permettant de numériser une image ou un document. On distingue généralement trois catégories de scanners :

– Les scanners à plat (Fig.3) permettant de numériser un document en le plaçant à plat contre une vitre. Il s'agit du type de scanner le plus courant.

– Les scanners à main (Fig.4) possédant une taille réduite. Ces scanners doivent être déplacés manuellement (ou semi manuellement) sur le document, par bandes successives, afin de le numériser en entier.

– Les scanners par défilement faisant défiler le document devant une fente lumineuse fixe afin de le numériser.



Fig. 3



Fig. 4

I-2- 4 Le lecteur de CD-ROM (Fig.5) : Il lit les informations sur les disques compacts (CD). Mais comme son nom l'indique, le Compact Disc Read Only Memory (CD-ROM) ne permet pas ni de modifier les informations ni d'en ajouter. Un rayon laser balaie le disque et extrait les informations sous forme de bits (0 ou 1) transmis vers la mémoire centrale. La vitesse avec laquelle le lecteur de CD-ROM lit les données s'exprime par un chiffre suivi d'un X. (Exemple 24 X ou 40X).



Lecteur CD

Fig. 5

I-2-5 Le lecteur DVD (Fig.6) : Les lecteurs DVD (Digital Versatile Disc) fonctionnent comme un lecteur de CD. Mais le DVD-ROM se différencie du CD-ROM par une grande capacité de stockage, obtenue grâce à une technologie de « gravure » plus fine. Sa forte capacité intéresse encore beaucoup le marché de l'informatique.



Fig. 6

I-2- 6 Le microphone (Fig.7) : C'est un périphérique permettant d'introduire des informations sonores. Il doit être connecté à une carte son.

I-2- 7 La manette de jeux (Fig.8) : c'est un pointeur utilisé généralement pour les jeux vidéos. Elle contient plus de boutons qu'une souris.



Fig. 7 : Microphone



Fig. 8 : Manette de jeux

I-2- 8 La webcam (Fig.9) : C'est une petite caméra vidéo branchable sur un PC. Elle peut être connectée généralement à travers le port USB. Elle permet de capturer des images réelles et de les numériser afin d'être traitées par ordinateur.

I-2- 9 Le stylo optique (Fig.10) : Il s'utilise sur un écran ou une table spéciale. Le stylo optique permet de sélectionner des objets ou de dessiner à main levée.



Fig. 9 : Webcam

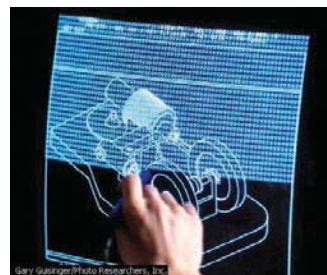


Fig. 10 : Stylo optique

I-2- 10 Appareil photo numérique (Fig.11) : Il est équipé d'une mémoire pour stocker les photos. Pour transférer ces photos vers l'ordinateur, on doit le brancher à ce dernier grâce à un connecteur.



Fig. 11

Activité 1

Aidé par votre professeur; établissez un tableau représentant les périphériques d'entrée et leurs caractéristiques (modèle, raccordement, . . .) d'un micro-ordinateur du laboratoire.

| Périphérique | Caractéristique |
|--------------|-----------------|
| | |
| | |
| | |
| | |

I-3- Les périphériques de sortie

I-3- 1 L'écran (ou moniteur) : C'est le périphérique d'affichage de l'ordinateur permettant de visualiser les données. On distingue deux types d'écran :

- Les écrans à tube cathodique (CRT pour Cathod Ray Tube) (Fig.13), équipant la majorité des ordinateurs de bureau. Il s'agit de moniteurs volumineux et lourds, possédant une consommation électrique importante.
- Les écrans à cristaux liquides (LCD pour Liquid Crystal Display) (Fig.12) équipant la totalité des ordinateurs portables, et actuellement beaucoup d'ordinateurs de bureau.

Un moniteur est caractérisé essentiellement par sa taille: c'est la longueur de la diagonale de l'écran. Elle s'exprime en pouces (un pouce équivaut à 2,54 cm). On trouve des écrans à 15, 17, 19 , 21 pouces et plus.



Fig. 12



Fig. 13

I-3- 2 L'imprimante : L'imprimante (printer) est un périphérique permettant de faire sortir des données de l'ordinateur sur papier. Il existe trois types d'imprimantes :

- L'imprimante matricielle (également appelée imprimante à aiguilles).
- L'imprimante à jet d'encre et imprimante à bulles d'encre (Fig.15).
- L'imprimante laser (Fig.14).

- Ces imprimantes sont généralement caractérisées par :
- la vitesse d'impression : généralement en pages par minute (ppm).
 - la résolution exprimée en points par pouces (ppp ou dpi).

I-3- 3 La table traçante (Fig.16) : Elle permet de tracer des graphiques sur plusieurs types de papier et en plusieurs formats de A0 à A4 à la différence d'une imprimante ordinaire. Ce périphérique est utilisé dans la CAO (Conception Assistée par Ordinateur) et dans le DAO (Dessin Assisté par Ordinateur). La tête d'impression de la table traçante est formée généralement par des stylos à encre.



Fig. 14



Fig. 15



Table traçante

Fig. 16

I-3- 4 Les haut-parleurs (Fig.17) : Ce sont des enceintes qui permettent de reproduire le son. Elles sont caractérisées par la puissance de sortie, mesurée en Watts.



Fig. 17



Fig. 18

I-3- 5 Le vidéo projecteur (Fig.18) : Ce périphérique permet de projeter le contenu de l'écran sur un écran de dimensions plus grandes. Il est utilisé généralement dans l'EAO (Enseignement Assisté par Ordinateur).

Activité 1

Aidé par votre professeur; établissez un tableau comparatif des trois types d'imprimantes, représentant les points suivants :

- les avantages et les inconvénients de l'une par rapport à l'autre.
- les caractéristiques de chacune.

I-4- Les périphériques d'entrée/sortie

I-4- 1 Le lecteur de disquettes : On trouve généralement un lecteur de disquettes dans une machine, il permet de lire et d'écrire sur des disquettes, même si cela peut paraître démodé à l'époque du multimédia. On distingue les lecteurs de disquettes suivants :

- Les lecteurs de disquettes 3"1/2 (Fig.19) (pour disquette 3"1/2).

- Les autres lecteurs de disquettes : Zip, JAZ, Streamer pour des disquettes de capacité importante.



Fig. 19

I-4- 2 Le graveur de CD ou de DVD : Aujourd'hui, pour choisir un graveur de CD ou de DVD, vous devez penser à l'interface (IDE, SCSI ou USB), la vitesse de gravure (32X, 48X, 52X ou plus) en CD-R (CD-ROM normal) et DVD ±R / RW et la possibilité éventuelle de graver les CD-RW (CD 'Rewriteable': CD réinscriptible environ 1000 fois) et le DVD±RW .

I-4- 3 Le modem : On distingue deux types de modem : Un modem interne (Fig.20) se présente sous forme de carte introduite dans l'un des connecteurs d'extension de la carte mère. Le modem externe (Fig.21) est un petit boîtier relié au micro ordinateur. Il sert à transmettre ou recevoir des données entre deux ordinateurs en utilisant les lignes téléphoniques. Avec un modem, on peut envoyer des fax ou des courriers électroniques, ou surfer sur le web. Un modem est caractérisé par le taux de transfert des informations. Un modem de "56 k" a une vitesse de 56 kbits/s (ou baud).



Fig. 20 : Modem interne

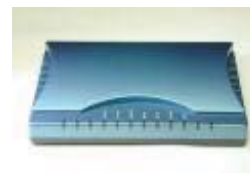


Fig. 21 : Modem externe

II- Les supports de stockage

II-1- Présentation (Fig.22)

Les supports de stockage (mémoires de masse ou mémoires auxiliaires) permettent de stocker les données d'une façon quasi permanente. Ce sont des supports physiques sur lesquels sont conservées des données. Ces supports peuvent être transportés d'un ordinateur à un autre. Un disque dur, une disquette, un CD-ROM,... constituent des supports de stockage.

L'unité de mesure de la capacité de stockage d'une mémoire de masse est l'octet. Un octet est formé de huit bits (binary digit) qui sont les plus petites parties de l'information pouvant être traitées par l'ordinateur. Un bit peut prendre la valeur 1 ou la valeur 0.

Les multiples de l'octet sont :

- le Kilo Octet (1 Ko = 1024 octets),
- le Mega Octet (1 Mo = 1024 Ko) et
- le Giga Octet (1 Go = 1024 Mo)



Fig. 22

II-2- Les disquettes (Fig.23 et Fig.24) : Les disquettes existent sous forme de disque souple, capable de conserver une aimantation en l'absence d'un champ magnétique. Les données sont écrites sur le disque par la tête de lecture / écriture de l'unité de disquette. Notons que pour utiliser une disquette, l'ordinateur devra être équipé du lecteur approprié.



Fig. 23



Fig. 24

II-3- Disque dur (fig.25) : Principale mémoire de stockage permanent de l'ordinateur. Physiquement, c'est un empilement de disques magnétiques. En général, le disque dur est le support sur lequel les systèmes d'exploitation et les logiciels divers que vous utilisez sont installés, ainsi que vos fichiers personnels. Un disque dur se caractérise par :

- sa capacité de stockage : c'est la quantité de données qu'il est capable de stocker . Elle se mesure en Gigaoctets (Go).
- son temps d'accès : Il spécifie le temps qui s'écoule entre le moment où vous demandez une information au disque et celui où il la fournit. Il se mesure en milli-secondes. Une valeur type étant de 10 ms.
- Sa vitesse de rotation est mesurée en tours par minute.



Fig. 25

II-4- Le flash disque (flash disk) (Fig.26) : C'est la nouvelle génération de support de stockage amovible, comme un disque dur mais externe, de faible taille. Il possède les mêmes caractéristiques qu'un disque dur.

Il ne peut être consulté qu'à partir d'un port USB.



Fig. 26

II-5- Le disque laser (Compact Disk)(Fig.27): C'est un support de stockage optique, il est devenu le support indispensable dans le monde de l'informatique et de l'audiovisuel. il se trouve sous forme de : CD-ROM, CD-RW et CD-AUDIO. Sa capacité est de l'ordre de 640 Mo et plus. Il se caractérise par son temps de transfert de données élevé, par son prix très bas et par sa durée de vie assez longue. comme la disquette, le disque laser ne peut être consulté qu'à partir d'un lecteur CD ou un graveur CD.



Fig. 27

II-6- Le DVD (Digital Versatile Disc) : Comme le disque laser mais avec un capacité plus grande qui est de l'ordre de quatre Giga Octets (4 Go), il se trouve sous forme de DVD-ROM et DVDRW. Il ne peut être consulté qu'à partir d'un lecteur DVD ou d'un graveur DVD.

II-7- La bande magnétique (Fig.28): ce genre de support est utilisé pour le stockage de grande quantité d'informations (archives). L'inconvénient de ce type de support est que le mode d'accès est séquentiel.



Fig. 28



Fig. 29

II-8- Le streamer (Fig.29) : Il se trouve sous forme de cassette magnétique, doté d'une capacité de stockage importante. Le streamer est aujourd'hui l'un des supports de stockage le plus fiable. Notons que pour utiliser une cassette streamer, l'ordinateur devra être équipé d'un lecteur approprié.

II-9- Le carte mémoire (Fig.30) : Il existe un grand nombre de formats de cartes mémoires non compatibles entre-eux. La carte mémoire a les mêmes caractéristiques qu'une mémoire vive mais les données ne se perdent pas lors d'une mise hors tension, c'est à dire que c'est une mémoire flash à semi conducteurs, non volatile et réinscriptible.



Fig. 30

En raison de sa durabilité et de sa vitesse élevée, la carte mémoire est idéale pour de nombreuses applications (les appareils photos numériques, les téléphones cellulaires, les imprimantes, les ordinateurs portables, les baladeurs mp3,...). Parmi les formats de cartes mémoire on trouve les cartes Compact Flash, les cartes Memory Stick, les cartes SmartMedia et les cartes MMC (MultimediaCard),...

Retenons

- **Les périphériques sont de trois types** : périphériques d'entrée, périphériques de sortie et périphériques d'entrée/ sortie.
- **Les périphériques d'entrée** : Ils permettent la lecture ou l'entrée de données pour être ensuite traitées. Comme exemple, on cite : le clavier, la souris, les lecteurs de format 5"1/4 (CD-ROM et DVD), le microphone, le scanner, la webcam,...
- **Les périphériques de sortie** : Ils sont utilisés pour l'écriture ou la sortie de données. Comme exemple, on cite : l'écran, l'imprimante, la table traçante, le haut-parleur, le vidéo projecteur,...
- **Les périphériques d'entrée/ sortie** : Ils permettent de transférer des données dans les deux sens, en entrée ou en sortie. On en cite les lecteurs de disquettes format 3"1/2 (lecteurs de disquettes 3"1/2 , Zip et JAZ.), les graveurs CD-ROM et DVD, le modem,...
- **Les mémoires de masse** appelées aussi mémoires auxiliaires sont des supports de stockage non volatiles. On distingue les supports magnétiques (les disquettes, les streamers et les bandes magnétiques) et les supports optiques (disque laser) et les supports électroniques (carte mémoire, flash disk) etc..

Exercices

Exercice 1

Classez les mots suivants dans le tableau ci-dessous où on mettra les périphériques d'entrée dans la première colonne, les périphériques de sortie dans la 2ème et les périphériques d'entrée/sortie dans la 3ème.

| | | | | |
|-------------|--------|------------|----------|---------------------------|
| Lecteur DVD | CD-ROM | Graveur | Moniteur | lecteur de disquettes |
| Imprimante | Souris | Microphone | WebCam | Flash disque Haut-parleur |

| | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |

Exercice 2

Soit un micro-ordinateur contenant un port parallèle, deux ports séries, deux ports USB, une carte son, une carte réseau, une carte graphique avec sortie TV.
Mettez une croix (X) devant le port qui correspond au périphérique.

| | Port parallèle | Port série | Port USB |
|--------------|----------------|------------|----------|
| Imprimante | | | |
| Ecran | | | |
| Haut parleur | | | |
| Souris | | | |
| Disque flash | | | |

Exercice 3

Soient les unités de mesure suivantes : pixel, octet, watt, baud, ppm, cps, kbs, pouce.
Remplissez la 3^{ème} colonne du tableau ci-dessous par l'unité de mesure correspondante.

| Elément | Caractéristique | Unité de mesure |
|------------------------|----------------------|-----------------|
| Imprimante matricielle | Vitesse d'impression | |
| Ecran | Taille | |
| Disque dur | Capacité | |
| Modem | Débit | |
| Haut parleur | Puissance | |
| Imprimante laser | Vitesse d'impression | |

Exercice 4

Quels sont les principaux avantages d'un disque dur par rapport à :

- 1- une disquette 2- une bande magnétique 3- un disque laser

Exercice 5

Remplissez le tableau ci-dessous en attribuant à chaque périphérique les caractéristiques demandées.

| Périphériques | Unité de mesure | Fixe/amovible | Volatil/Permanente |
|---------------|-----------------|---------------|--------------------|
| Disquette | | | |
| Flash disque | | | |
| RAM | | | |
| Disque dur | | | |
| DVD | | | |



Chapitre 3

Systemes d'exploitation

Objectifs :

Définir le rôle et les fonctionnalités d'un système d'exploitation

Utiliser les principales fonctions d'un système d'exploitation

Plan du chapitre :

Leçon 1 :
Présentation et services

Leçon 2 :
Concepts de base

Leçon 3 :
Utilisation pratique du système d'exploitation

Exercices



Le système d'exploitation joue un rôle central dans le fonctionnement de tout ordinateur. Il s'agit d'un ensemble de logiciels qui vont se mettre au service des logiciels applicatifs (un traitement de texte, un navigateur Internet, ...) afin de rendre possible l'exécution simultanée de ses derniers. Ils permettent aussi de faciliter le partage des ressources d'un ordinateur comme la mémoire centrale, le processeur, le disque dur, etc.

Ce chapitre présente d'abord les services rendus par un système d'exploitation. Ensuite, l'accent est mis sur ses concepts fondamentaux. En dernier lieu, certains aspects parmi les plus usuels de l'utilisation des systèmes d'exploitation Microsoft Windows (du monde des logiciels payants) et GNU/Linux (du monde des logiciels libres) sont introduits.

Leçon 1

Présentation et services

Objectifs spécifiques

L'objectif de cette leçon est de définir la notion de système d'exploitation et de présenter les services de base qu'il offre.

Plan de la leçon

- I • Définition d'un système d'exploitation
- II • Services offerts par un système d'exploitation
 1. La gestion et le partage des ressources matérielles
 2. Gestion des ressources logicielles et le service multi-tâches
 3. Gestion des sources de données (fichiers et répertoires)
 4. Gestion du dialogue avec l'utilisateur

Retenons

Leçon 1

Présentation et services

I Définition d'un système d'exploitation

Un système d'exploitation d'un ordinateur peut être défini, de prime abord, comme étant l'ensemble des logiciels qui permettent à un usager d'exploiter les potentialités de cet ordinateur en termes de stockage d'informations, de calcul, de traitements divers ou de communications réseaux.

On peut alors dire qu'en absence du système d'exploitation, un ordinateur s'apparentera à un agrégat de composants électroniques n'ayant aucune utilité.

Un système d'exploitation possède souvent une structure assez complexe ; mais nous pouvons dire d'une manière assez sommaire qu'il est composé d'un noyau (kernel en anglais) et d'un grand nombre de bibliothèques et d'utilitaires cohérents et complémentaires les uns aux autres.

– Le noyau est la partie qui est intimement liée au matériel et qui assure le dialogue avec ce dernier.

– Les bibliothèques et les utilitaires gèrent eux plutôt les services offerts aux applications et aux usagers.

La figure 1 illustre le positionnement et la structure d'un système d'exploitation.

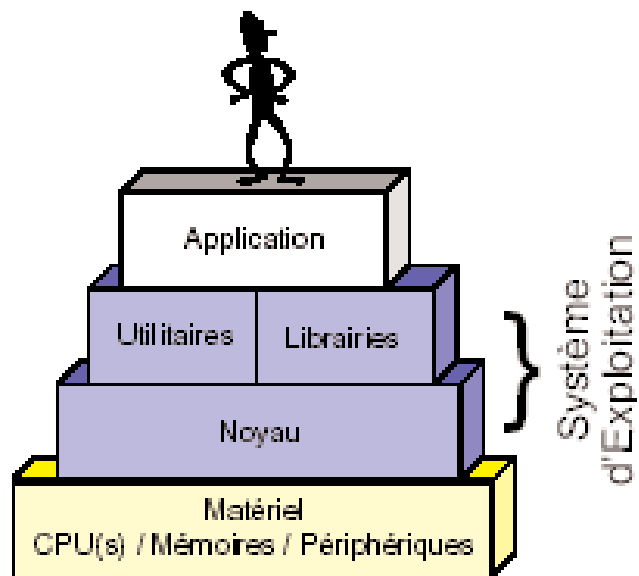


Fig. 1 : Positionnement et structure d'un système d'exploitation

Voici quelques exemples de systèmes d'exploitation :

- La famille Microsoft Windows (98, 2000, Millénium, XP, ...) pour les ordinateurs personnels (PC)
- Les distributions Linux (Fedora, Redhat, Novell/Suse, Mandriva, ...) pour les ordinateurs personnels (PC)
- Le système MacOS pour les micro-ordinateurs du type Macintosh
- Le système Solaris dédié principalement aux stations de travail de la marque Sun Microsystems.
- Le système AIX pour la gamme mini et gros ordinateurs d'IBM.

Notons que, nombreux systèmes d'exploitation partagent la même structure dite UNIX qui, à l'origine, a été développée dans les laboratoires Bell de AT&T au début des années 1970 aux Etats Unis. De part sa notoriété, ce système a été adopté par divers éditeurs. GNU/Linux, un descendant direct de ce système est un exemple en la matière. Son noyau a été créé en 1991 par le jeune étudiant finlandais Linus Torvalds. Il présente la particularité d'être gratuit et de source libre (open-source). Ceci veut dire qu'on peut l'obtenir sans frais et avec les codes sources complets des programmes qui ont servi à le programmer. Cette option permet aux utilisateurs d'apporter des modifications et des améliorations au système.

II Services offerts par un système d'exploitation

Les principaux services offerts par un système d'exploitation sont :

II-1 La gestion et le partage des ressources matérielles

Le système d'exploitation est responsable du dialogue avec toutes les ressources matérielles de l'ordinateur (le processeur central, la mémoire vive, le disque dur, la carte graphique, le modem...). Il permet également de gérer les périphériques d'un ordinateur en facilitant leur installation, leur configuration, leur mise à niveau ou leur suppression. Les périphériques sont souvent gérés à travers des modules logiciels appelés pilotes (drivers en anglais). Par ailleurs, le système d'exploitation assure le partage de ces ressources par les différents programmes applicatifs qui sont en exécution.

Activité 1

Listez les ressources matérielles de votre ordinateur.

Indications :

- Poste de travail puis Propriétés
- Matériel puis Gestionnaire des périphériques

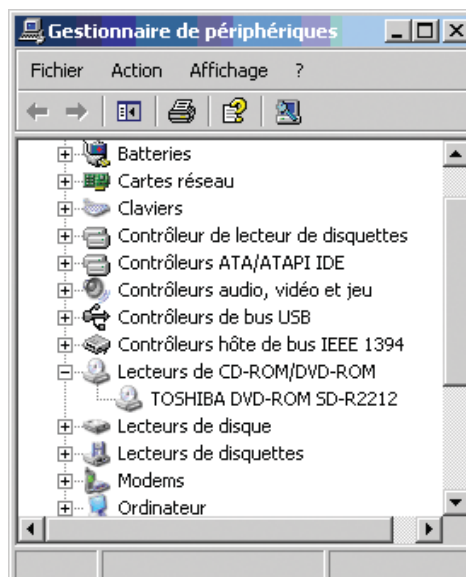


Fig. 2 : Vue du gestionnaire de périphériques

II-2 Gestion des ressources logicielles et le service multi-tâches

Le système d'exploitation facilite la gestion des ressources logicielles en apportant à l'utilisateur l'aide convenable à l'installation, à la configuration, à la mise à jour et à la désinstallation cohérente des logiciels auxquels il a recours sur son ordinateur. Par ailleurs, dans le cas où un utilisateur serait amené à lancer l'exécution simultanée de plusieurs logiciels, le système d'exploitation assurera, via ce qu'on appelle le service multi-tâches, le déroulement cohérent de ces logiciels. Il s'agit en général d'un pseudo-parallélisme puisque le système d'exploitation ne fait que partager le temps d'un unique processeur central entre les différents programmes concurrents.

Activité 2

Listez les programmes installés sur votre ordinateur.

Indications :

- Menu Démarrer, Puis Paramètres.
- Panneau de configuration, puis Ajout/suppression de programmes.



Fig. 3 : Vue sur la configuration des programmes

II-3 Gestion des sources de données (fichiers et répertoires)

Les fichiers sont des structures qui permettent de stocker les données des utilisateurs sur les périphériques de stockage. Ils sont en général organisés en répertoires. Le système d'exploitation offre les services de création, de mise à jour et de suppression de fichiers et de répertoires. Le service responsable de cette gestion est appelé le système de gestion de fichiers.

Le système de gestion de fichiers offre également les services de partitionnement du disque dur (découpage physique du disque en plusieurs espaces) et de recherche de fichiers. Ces services sont utiles surtout dans le cas de gros volumes de données. Les versions récentes de systèmes d'exploitation intègrent les services de gravure sur CD ou sur DVD et de reconnaissance d'arborescences de données à partir de clés USB.

II-4 Gestion du dialogue avec l'utilisateur

Il s'agit du service le plus visible du système d'exploitation. En effet, les systèmes d'exploitation offrent souvent des interfaces graphiques sous la forme d'un bureau de travail permettant d'organiser l'accès aux applications et aux ressources périphériques. L'interface utilisateur permet également de faciliter la configuration de ces ressources.

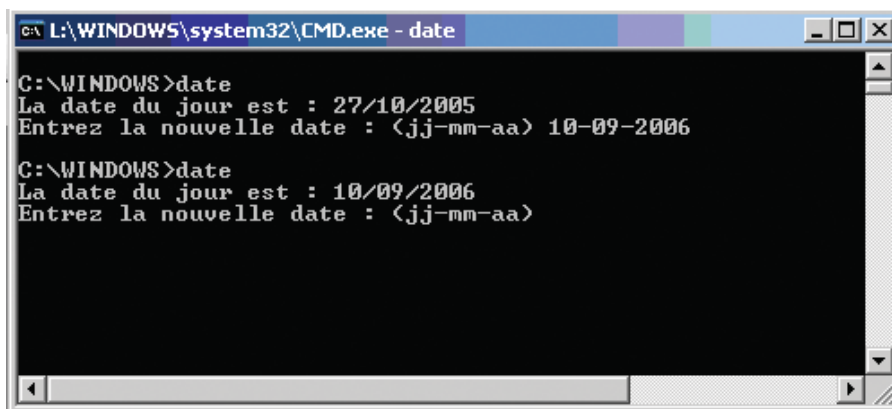
En plus des interfaces graphiques qui amènent un confort d'exploitation incontestable mais qui exigent une carte graphique, les systèmes d'exploitation offrent également des interfaces en mode texte (commandes MS-DOS, commandes Shell Linux, ...) qui peuvent être sollicitées pour certains usages spécifiques. La leçon III vous fournira davantage d'informations relatives à l'utilisation de l'interface graphique.

Activité 3

- Lancez l'interface en mode texte (l'invite de commandes) sous Windows
- Changez la date avec la commande MS-DOS date. Vérifiez que cette nouvelle date est prise en compte.

Indications :

- Menu Démarrer puis Accessoires
- Lancez Invite de commandes



```
C:\WINDOWS\system32\CMD.exe - date
C:\WINDOWS>date
La date du jour est : 27/10/2005
Entrez la nouvelle date : <jj-mm-aa> 10-09-2006

C:\WINDOWS>date
La date du jour est : 10/09/2006
Entrez la nouvelle date : <jj-mm-aa>
```

Fig. 4 : Invite de commandes

Retenons

- Un système d'exploitation est un ensemble de programmes nécessaires au démarrage et à l'utilisation d'un ordinateur.
- Le système d'exploitation offre un ensemble de fonctionnalités qui incluent la gestion et le partage des ressources matérielles, la gestion des ressources logicielles, la gestion des fichiers et la gestion du dialogue avec l'utilisateur.

Leçon 2

Concepts de base

Objectifs spécifiques

L'objectif de cette leçon est de présenter les concepts de base sur lesquels repose un système d'exploitation.

Plan de la leçon

- I• Les fichiers et les répertoires
- II• Les processus
- III• Les utilisateurs

Retenons

Leçon 2

Concepts de base

Le système d'exploitation est un logiciel complexe dont la conception repose sur un ensemble de concepts dont les plus pertinents sont les suivants :

I- Les fichiers et les répertoires

Un fichier est une structure qui permet de stocker des données ou des programmes sur un support physique. Souvent un fichier possède un nom sous la forme **nom.extention**. L'extension sert à identifier les types de fichiers. L'extension **html**, par exemple, désigne des pages web, l'extension **doc** désigne des documents produits avec le logiciel du traitement de textes Microsoft Word.

| Extension | Type du fichier |
|-----------|---|
| html | Page web |
| gif | Fichier image |
| txt | Fichier texte |
| exe | Programme exécutable |
| bat | Fichier de commandes exécutable |
| doc | Document Microsoft Word |
| mdb | Fichier de base de données Microsoft Access |
| pas | Fichier de code source Pascal |

Fig. 1 : Extension de fichiers

Plusieurs fichiers peuvent être stockés dans un répertoire ou encore un dossier. Un répertoire peut contenir également aussi bien des fichiers que d'autres dossiers appelés sous-répertoires. On parle ainsi d'arborescence de répertoires.

Activité 4

Lancez l'Explorateur Windows et explorez l'organisation des fichiers de votre disque dur C :\.

Indications :

- Menu Démarrer, puis Programmes, puis Accessoires
- Lancez l'explorateur Windows

Astuce : Il est également possible de lancer l'explorateur en appuyant simultanément sur la combinaison (touche Windows + le caractère « e »).

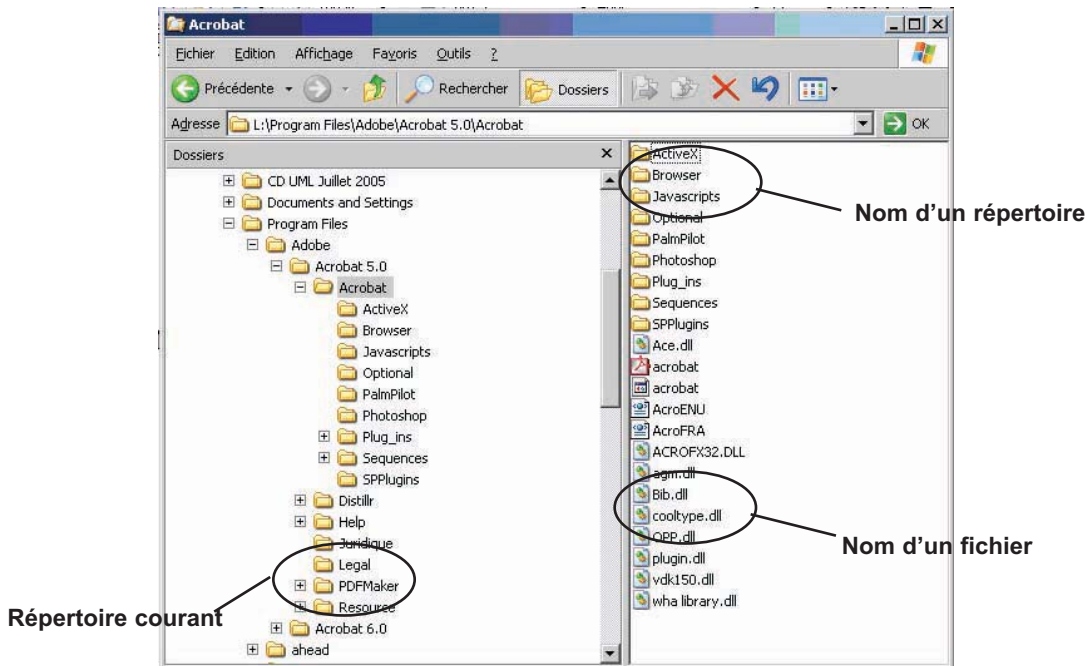


Fig. 2 : Vue de l'explorateur de fichiers

II- Les processus

Un processus est un programme en cours d'exécution. Un système d'exploitation en fonctionnement s'apparente à une « usine » à processus partageant les ressources de la machine (processeur, mémoire, ...). On distingue deux types de processus :

- les processus système qui font partie du système d'exploitation lui-même (par exemple les processus internes chargés de la gestion de la mémoire centrale).
- les processus applicatifs qui correspondent à des applications lancées par les utilisateurs pour accomplir certaines tâches (applications de bureautique telles que Microsoft Word, Adobe Acrobat Reader , ...).

Activité 5

Listez tous les processus actifs du système.

Indications :

- Appuyez sur la combinaison de touches CTRL-ALT, puis choisissez Gestionnaire des tâches, puis appuyez sur l'onglet « Processus ».

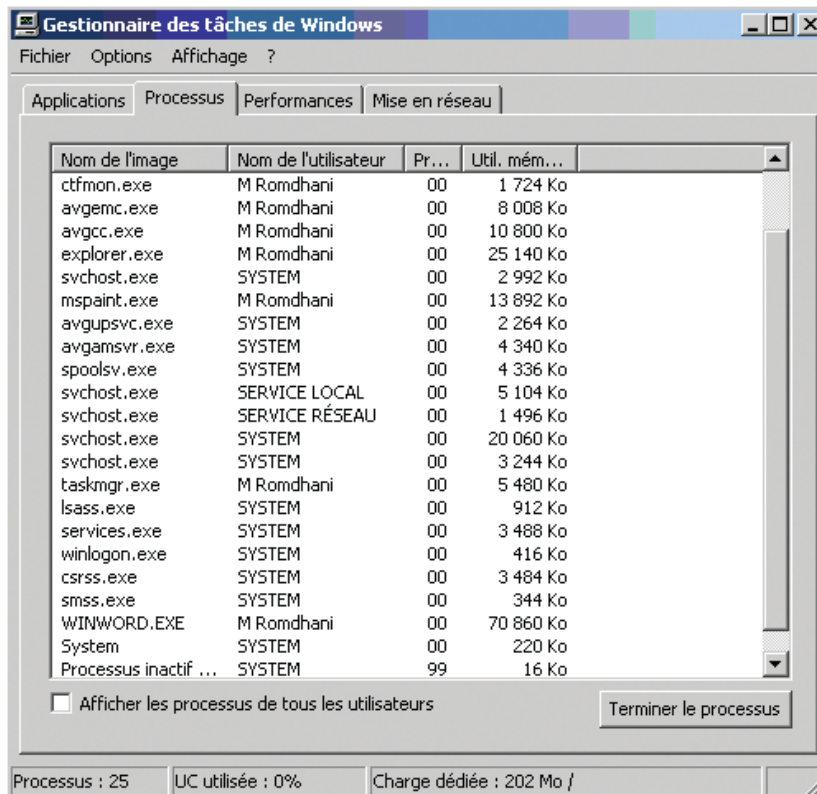


Fig. 3 : Vue du gestionnaire des tâches

On remarque bien dans cette liste le processus WINWORD.EXE qui correspond au logiciel Microsoft Word. Les autres processus correspondent plutôt à des processus système.

III- Les utilisateurs

Le concept d'utilisateur dans un système d'exploitation est très important. On travaille toujours sur ordinateur sous l'identité d'un utilisateur donné. Plusieurs utilisateurs peuvent dans certains cas se connecter simultanément sur la même machine (Ce cas de figure correspond à ce que l'on appelle le mode multi-utilisateurs). Un utilisateur correspond alors à un contexte de travail propre et qui est initié dès l'ouverture d'une session de travail. Les utilisateurs possèdent chacun son identité système, son mot de passe et son répertoire de travail.

Dans tous les systèmes d'exploitation, on distingue un utilisateur particulier, souvent appelé « super-utilisateur » qui possède tous les droits d'administration de l'ordinateur (Gestion d'autres utilisateurs, Installation/désinstallation de matériels, Mise à jour du système, ...). Cet utilisateur s'appelle « Administrateur » sur les plates-formes Windows et « root » sur les environnements UNIX/Linux.

La notion de groupe est utilisée dans les systèmes d'exploitation afin d'organiser les utilisateurs et de faciliter leur gestion. De manière plus précise, un groupe est constitué par des utilisateurs qui sont sensés exploiter les mêmes applications et les mêmes ressources du système.

Activité 6

Créez un nouvel utilisateur.

Indications :

- Demarrer / panneau de configuration
- Outils d'administration / Gestion de l'ordinateur
- Utilisateurs et Groupes locaux puis utilisateur
- menu Action / Nouvel utilisateur.

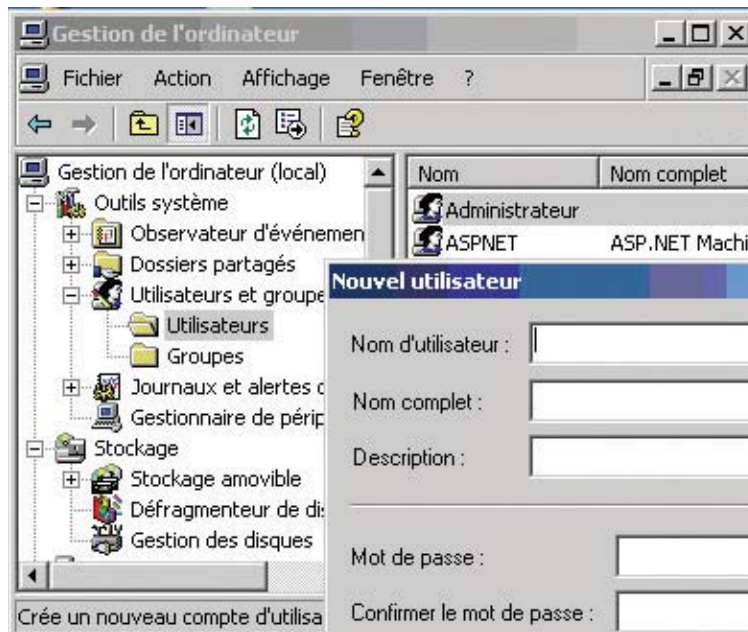


Fig. 4 : Création d'un nouvel utilisateur

- Saisissez le nom de l'utilisateur et assignez lui un mot de passe.
- Quittez la session et connectez-vous avec cette nouvelle identité.

Retenons

- Un fichier est une structure qui permet de stocker des données ou des programmes sur un support physique.
- Un répertoire est une structure qui permet d'organiser de manière hiérarchique des sous-répertoires ou des fichiers.
- Un processus est un programme en cours d'exécution.
- Un utilisateur correspond à une identité sous laquelle le système d'exploitation le reconnaît dès l'ouverture d'une nouvelle session de travail.
- Un groupe correspond à un ensemble d'utilisateurs travaillant ensemble sur les mêmes applications.

Leçon 3

Utilisation pratique du système d'exploitation

Objectifs spécifiques

L'objectif de cette leçon est d'apprendre l'utilisation pratique d'un système d'exploitation et de présenter les services de base qu'il offre.

Plan de la leçon

- I• Le bureau
- II• Le menu Démarrer ou lanceur
- III• La barre des tâches
- IV• Configuration des propriétés d'affichage
- V• Panneau de configuration
- VI. Lancement de programmes
- VII• Fichiers et répertoires
 1. Exploration de l'arborescence des fichiers et répertoires
 2. La gestion des fichiers
 3. La gestion des répertoires
 4. La recherche des fichiers et des dossiers

Retenons

Exercices

Leçon 3 Utilisation pratique du système d'exploitation

Nous introduisons dans cette leçon quelques aspects parmi les plus usuels de l'utilisation quotidienne des deux systèmes d'exploitation Microsoft Windows XP et GNU/Linux.

I. Le bureau

Le bureau est une interface graphique qui facilite l'accès aux applications et aux services du système d'exploitation. Il est présent dans tous les systèmes d'exploitation modernes et offre à peu près les mêmes services. Il est lancé automatiquement au démarrage de la machine. Les composants importants du bureau sont :

- **Poste de travail** : Ce composant permet d'accéder aux divers périphériques de votre machine (lecteurs disques, lecteurs CD-ROM, unité disquette).
- **Voisinage réseau** : Ce composant permet d'accéder aux ordinateurs du voisinage dans le cas d'un fonctionnement en réseau. Il permet également de configurer les paramètres réseau de l'ordinateur.
- **Mes Documents** : Ce répertoire, spécifique à MS Windows, est l'endroit de stockage par défaut de tout nouveau document créé par un utilisateur.
- **Navigateur web** (Internet Explorer dans le cas de Windows ou de FireFox/Mozilla dans le cas de Linux) : Il s'agit d'un raccourci permettant un accès rapide au navigateur.
- **Corbeille** : Ce composant permet de collecter tout fichier effacé permettant ainsi une éventuelle restitution.
- **Menu démarrer ou lanceur** : Ce composant important permet à l'utilisateur de démarrer les programmes et les outils d'administration et de configuration du système.

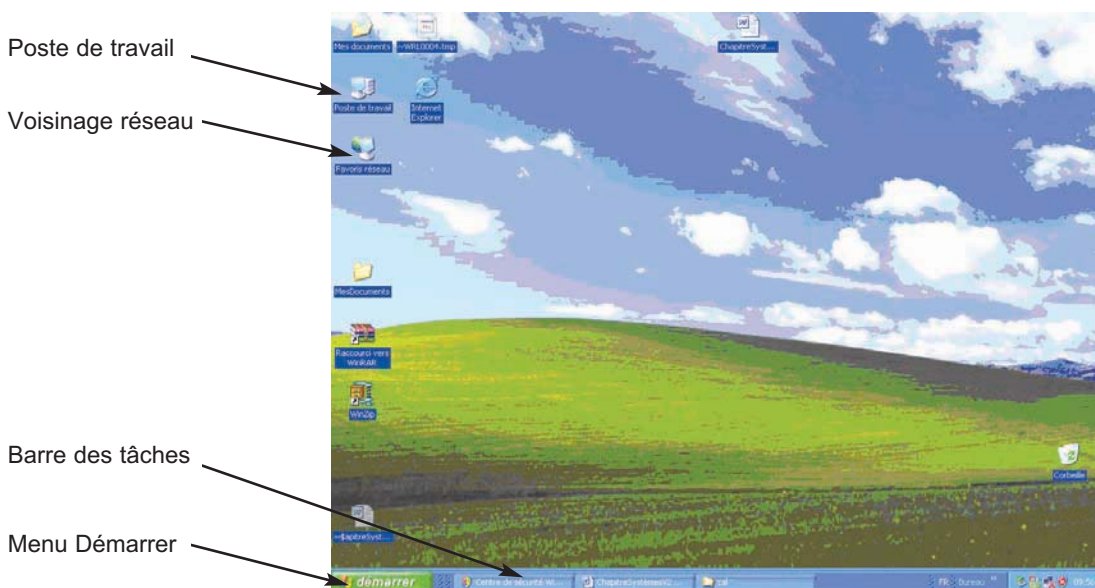


Fig. 1 : Bureau du Windows XP

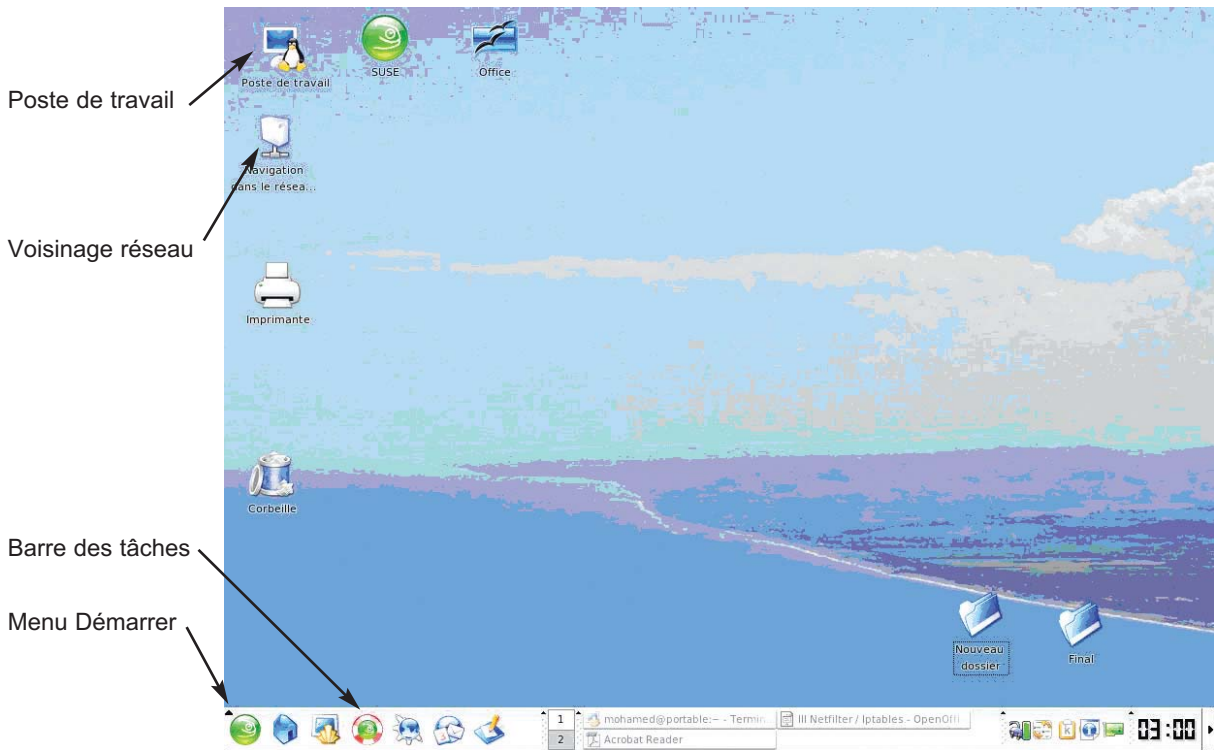


Fig. 2 : Bureau de Linux

II- Le menu démarrer ou lanceur

Ce composant du bureau permet d'accéder à toutes les options, programmes et contrôle du système d'exploitation. Il est très utile et il est important de bien apprendre son utilisation.

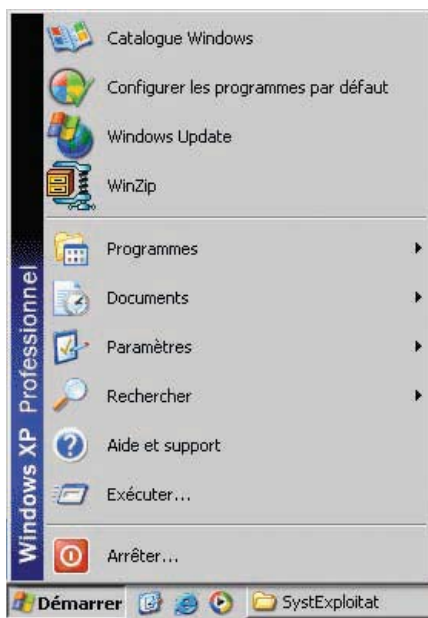


Fig. 3 : Lanceur de Microsoft Windows XP

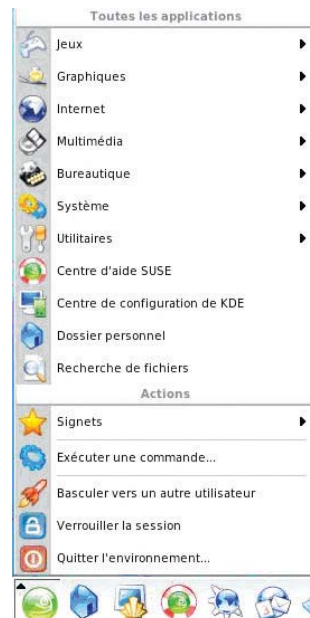


Fig. 3 bis : Lanceur d'une distribution de GNU/Linux

III- La barre des tâches

La barre des tâches permet de montrer les applications en cours d'exécution et de passer facilement de l'une à l'autre (en cliquant dessus). Par exemple, suite au lancement du logiciel Paint, une zone correspondante à cette application est apparue dans la barre des tâches comme le montre la figure ci-dessous.

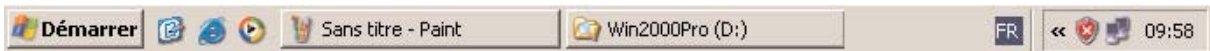


Fig. 4: Barre des tâches

IV- Configuration des propriétés d'affichage

Pour pouvoir accéder aux propriétés de l'affichage, il faut cliquer sur le bouton de droite directement sur le bureau et choisir Propriétés dans le menu défilant.

Les propriétés de l'affichage permettent de :

- Modifier l'image d'arrière-plan de votre bureau (Onglet Bureau)
- Modifier l'écran de veille (Onglet Écran de veille)
- Changer la résolution d'affichage (Onglet Paramètres)



Onglet Bureau

Il est possible de changer l'image de votre bureau. Vous choisissez l'image qui vous convient le mieux. Si aucune image ne vous intéresse, vous pouvez choisir vous-même une image que vous avez dans votre ordinateur. Pour se faire, vous devez appuyer sur Parcourir et aller chercher l'image à l'endroit où elle est située dans votre ordinateur.

L'option position vous permet de positionner votre image. Trois choix sont possibles: Étirer, Centrer et Mosaïque.

Fig. 5 : Configuration de l'arrière plan sous Windows XP

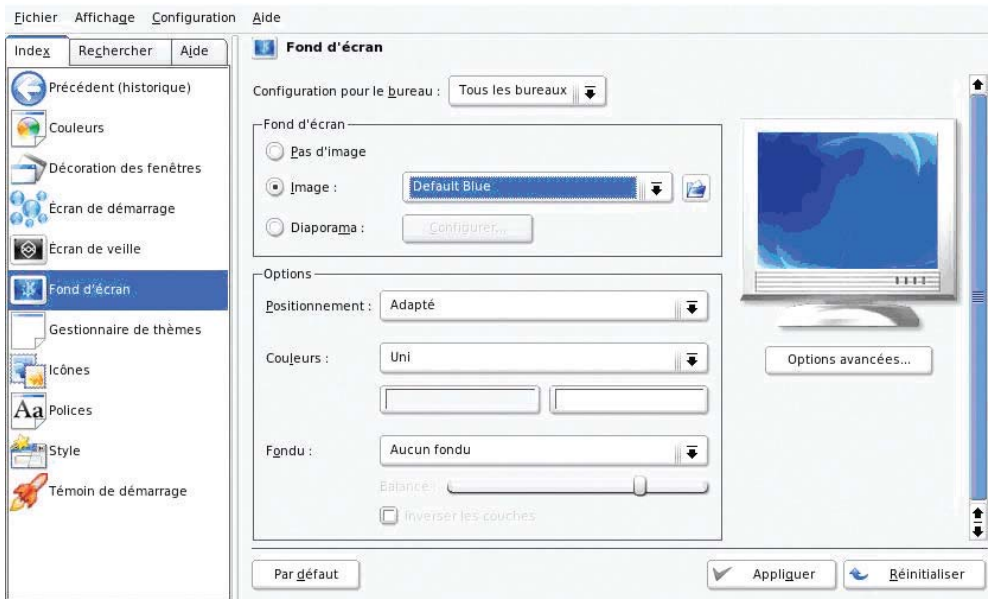


Fig. 5 bis: Configuration de l'arrière plan sous Linux

Activité 7

Modifiez l'écran de fond de votre bureau.



Fig. 6: Configuration de l'écran de veille

Onglet Écran de veille

Vous pouvez changer l'écran de veille par défaut de votre ordinateur par le contenu disponible qui vous convient.

Il est important de noter que vous pouvez définir le temps pendant lequel Windows commencera à afficher l'écran de veille. Par exemple, si vous avez spécifié 10 minutes, alors après 10 minutes d'inactivité, Windows affichera l'écran de veille afin de ne pas altérer votre moniteur.

Activité 8

Définissez un nouvel écran de veille affichant le mot « Informatique » après 5 minutes d'inactivité.



Fig. 7: Configuration de la résolution de l'écran

Onglet Paramètres

L'onglet paramètres est très important, car il permet d'accéder aux options de votre carte graphique. Ainsi, vous pouvez changer la résolution de votre écran.

Conseil pour le paramétrage :

| Écran | Résolution | Couleur |
|-----------|-------------|---------|
| 15 pouces | 800 X 600 | 32 bits |
| 17 pouces | 1024 X 768 | 32 bits |
| 19 pouces | 1600 X 1200 | 32 bits |

V- Panneau de configuration

Le panneau de configuration est une interface qui regroupe les outils d'administration et de la configuration du système.

Sous les systèmes Windows XP, ce panneau de configuration est accessible à partir du menu Démarrer. Un aperçu du panneau de configuration est montré dans la figure 8

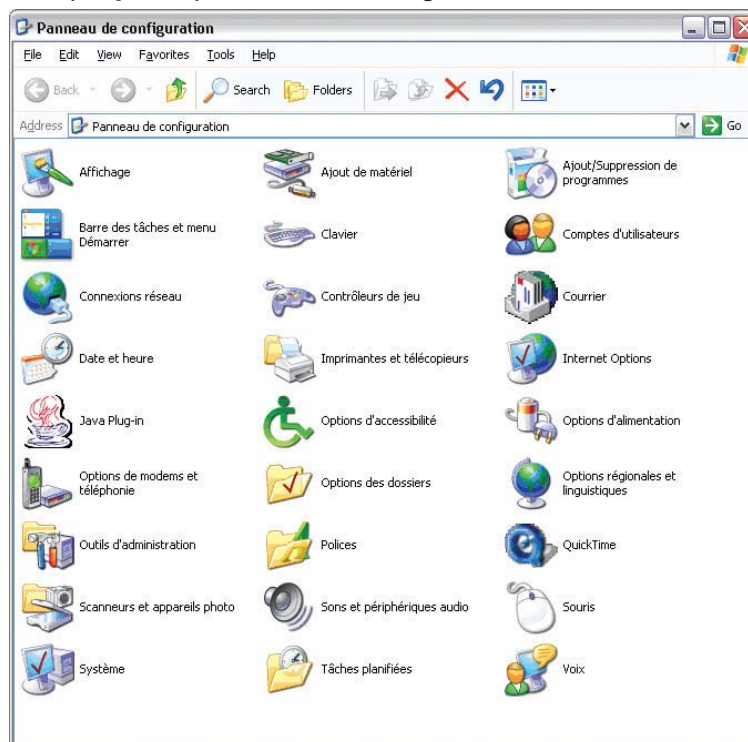


Fig. 8: Panneau de configuration de Windows XP

Ci-après, un résumé des fonctions de certaines de ces icônes est donné.

| | |
|---|--|
|  Comptes d'utilisateurs | Ce bouton vous permet de vous connecter avec votre nom d'utilisateur et un mot de passe. Vous pouvez changer votre mot de passe et le nom d'utilisateur à partir de ce bouton. |
|  Affichage | Ce bouton permet de faire des modifications à l'affichage de votre bureau de travail. |
|  Ajout de matériel | Permet de faire l'installation de nouveaux périphériques tels que l'imprimante, graveur, etc. |
|  Ajout/Suppression de programmes | Permet de faire l'installation (Ajout) de nouvelles applications ou la suppression de programmes existants. |
|  Clavier | Permet de modifier la langue du clavier ainsi que la vitesse du curseur. |
|  Connexions réseau | Permet de faire la configuration pour les connexions réseau. |
|  Date et heure | Permet d'ajuster l'heure et la date de l'ordinateur. |
|  Imprimantes et télécopieur: | Permet de faire la configuration des imprimantes et télécopieurs. |
|  Internet Options | Permet la configuration de l'accès à Internet. |
|  Barre des tâches et menu Démarrer | Permet de personnaliser la barre de tâche et le menu Démarrer. |
|  Options de modems et téléphonie | Permet de modifier les options du modem. |
|  Options régionales et linguistiques | Permet de modifier les options régionales et linguistiques de Windows. Il permet par exemple l'ajout du support de la langue arabe. |
|  Outils d'administration photo | Permet de faire la configuration des scanners ou des appareils photo. |
|  Souris | Permet de faire la configuration de la souris, tel que les boutons gauche/droite. |

Fig. 9 : Fonction des icônes du panneau de configuration

Les fonctions qui sont listées dans le tableau précédent ont leurs équivalents dans la distribution Linux Suse de Novell. La figure 9 bis correspond au centre de configuration KDE permettant d'accéder à ces fonctions.

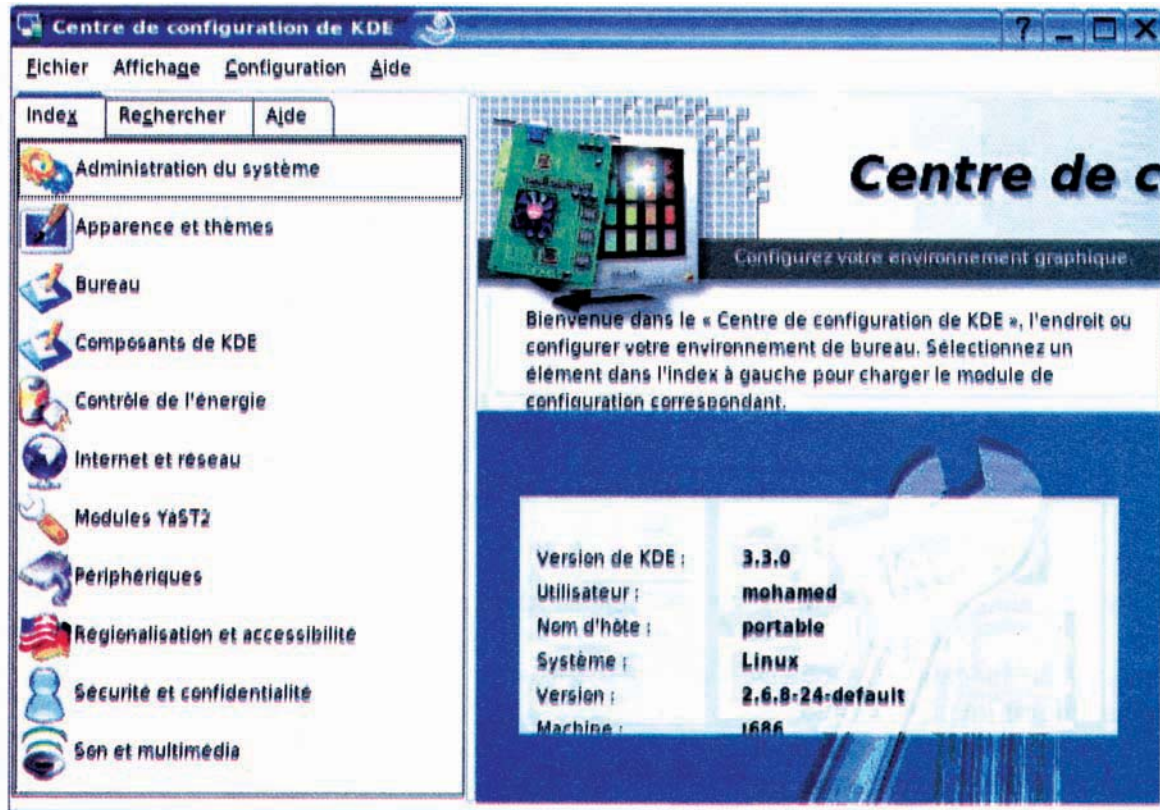


Fig.9 bis : Panneau de configuration de Linux

Activité 9

Changez l'heure à partir du panneau de configuration de Windows XP.

Indications :

- Lancez le panneau de configuration à partir du Menu Démarrer/Option Paramètres.
- Sélectionnez Date et Heure, la boîte de dialogue suivante apparaîtra.
- Modifiez l'heure et cliquez sur OK.

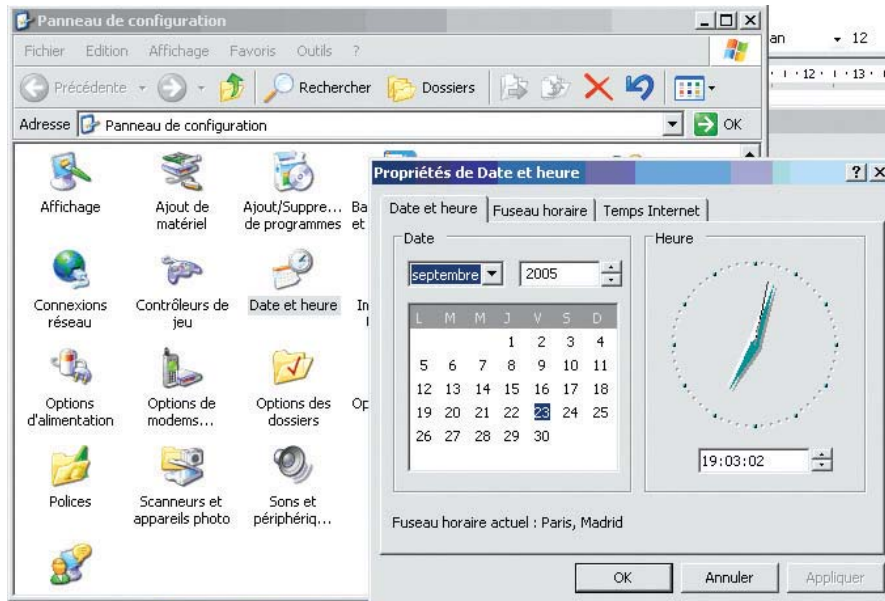


Fig.10: Modification de l'heure

VI- Lancement de programmes

Il y a plusieurs manières de lancer (ou démarrer) un programme. La manière la plus courante consiste à utiliser le menu démarrer (le lanceur). On sélectionne Programmes, puis on choisit le programme à démarrer. Il est toutefois possible de lancer un programme à partir d'un raccourci se trouvant sur le bureau. Il est également possible de déclencher le démarrage d'un programme en cliquant simplement deux fois sur l'un des fichiers de données construit par ce programme. Une fois que vous avez démarré un programme, un bouton le représentant apparaît dans la barre des tâches. Pour passer d'un programme en cours d'exécution à un autre, cliquez sur le bouton correspondant dans la barre des tâches.

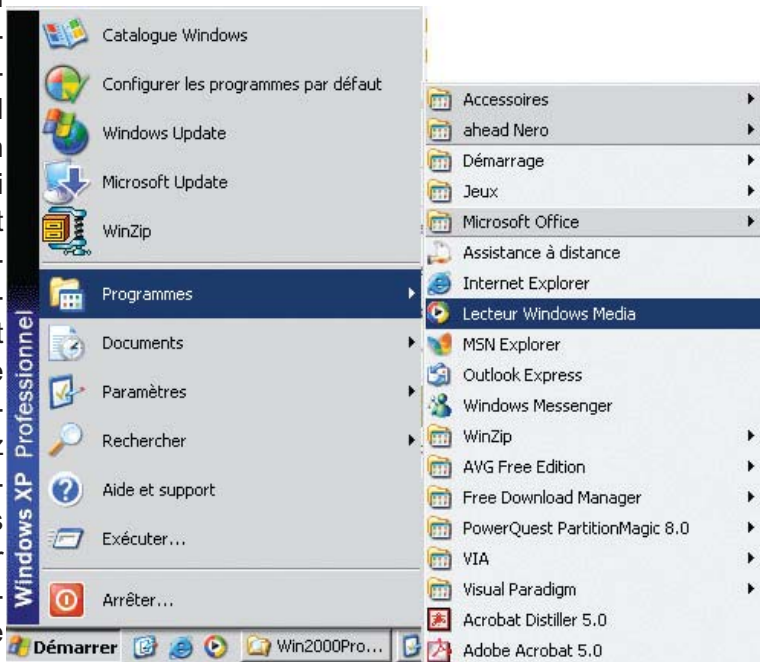






Fig..11: Lancement du programme MediaPlayer sous Windows

Afin de quitter un programme, il suffit de cliquer sur Quitter du menu Fichier de ce programme. Il est également possible de quitter un programme en cliquant sur  se trouvant à l'extrémité droite de la barre de titres.

Par ailleurs, nous disposons dans la barre de titres les boutons suivants :

- Le bouton  permet de réduire la fenêtre à un bouton sur la barre des tâches.
- Le bouton  permet d'afficher la fenêtre en plein écran.
- Le bouton  permet de rétablir la taille précédente de la fenêtre une fois agrandie

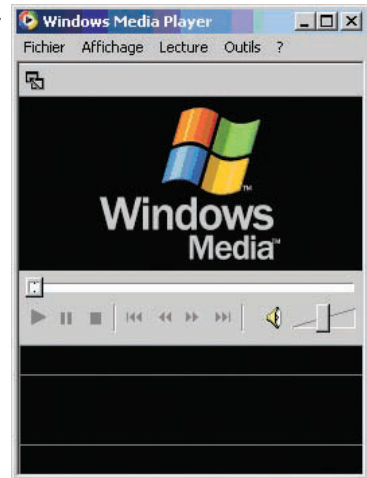


Fig.12: Windows Media Player

VII- Fichiers et répertoires

VII-1 Exploration de l'arborescence des fichiers et répertoires

L'explorateur de Windows est le programme qui permet d'afficher l'organisation arborescente de l'ensemble des fichiers et des répertoires contenus dans un support de stockage. L'**explorateur** de Windows est accessible à partir du menu démarrer en développant les options Programmes, puis Accessoires. Il est également possible de le lancer en pointant sur le bouton **Démarrer** et en cliquant sur le bouton droit de la souris.



Fig.13: Lancer l'explorateur

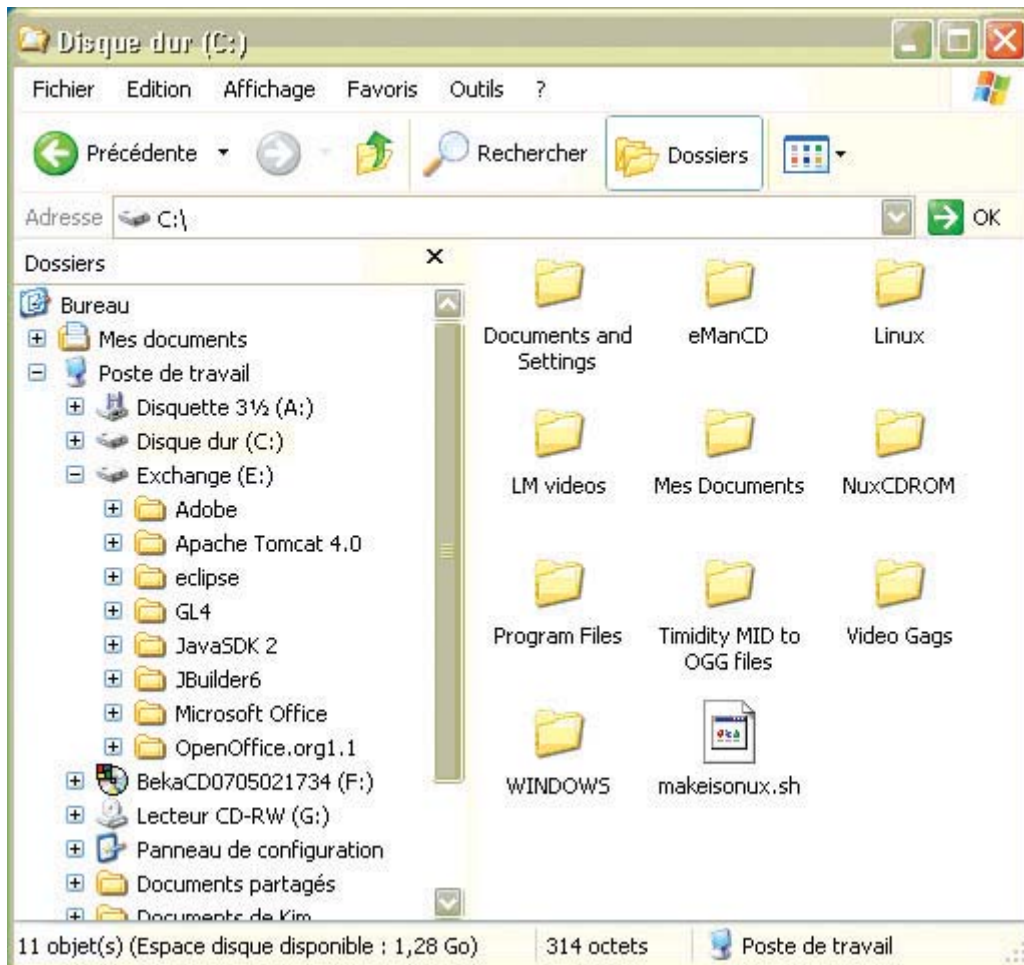
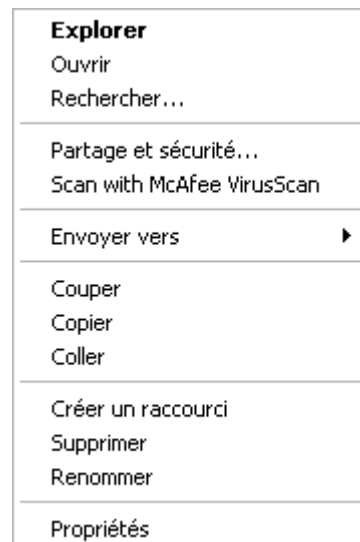


Fig. 14: Explorateur Windows

VII-2 La gestion des fichiers

Plusieurs opérations peuvent être effectuées sur un fichier. On recense parmi elles :

- la duplication (copier/coller)
- le déplacement (couper/coller)
- la suppression (supprimer)
- le renommage (renommer),...



0 Fig.15: Gestion des fichiers

Activité 10

Editez les propriétés d'un fichier. Il s'agit d'un ensemble d'informations sur le fichier (type, taille, emplacement, date de création, de modification et du dernier accès).

Indications :

- Lancez l'explorateur Windows et localisez un fichier de votre choix.
- Cliquez sur le bouton droit et sélectionnez Propriétés.
- La boîte de dialogue devra vous fournir les propriétés de votre fichier.

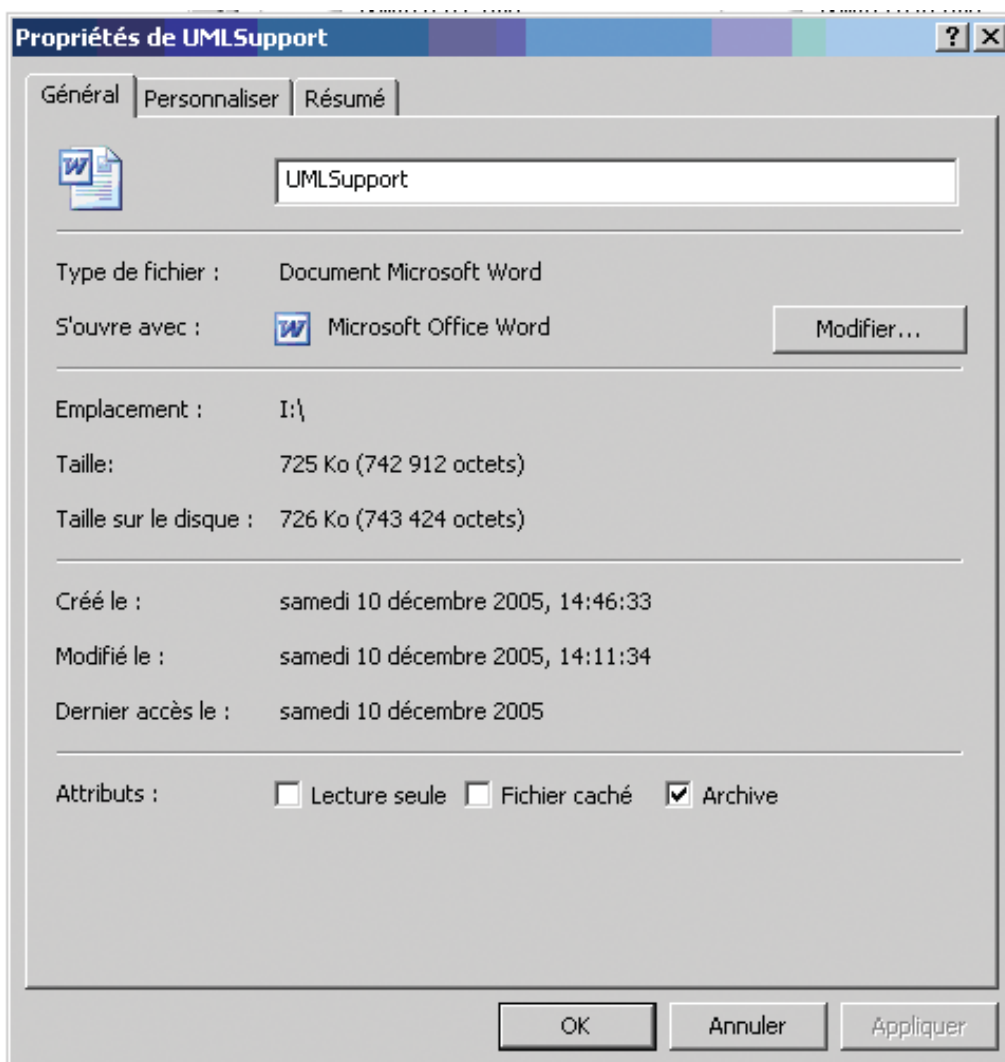


Fig. 16: Propriétés d'un fichier

Activité 11

Toujours à partir de l'explorateur, pointez un fichier de votre choix et réalisez les manipulations suivantes :

- Créez un raccourci Bureau sur le fichier
- Copiez ce fichier vers le répertoire Mes Documents

VII-3 La gestion des répertoires

De manière analogue aux opérations appliquées sur les fichiers, plusieurs opérations peuvent être effectuées sur un répertoire. On recense parmi elles :

- la duplication (copier/coller),
- le déplacement (couper/coller),
- la suppression (supprimer),
- le renommage (renommer),...

Activité 12

Créez un nouveau répertoire appelé MonRep dans le répertoire Mes Documents.

Indications :

- Lancez l'explorateur Windows et se placer dans le répertoire Mes Documents
- Se positionner dans la partie droite de l'explorateur, cliquez sur le bouton droit et sélectionnez Nouveau puis Dossier.

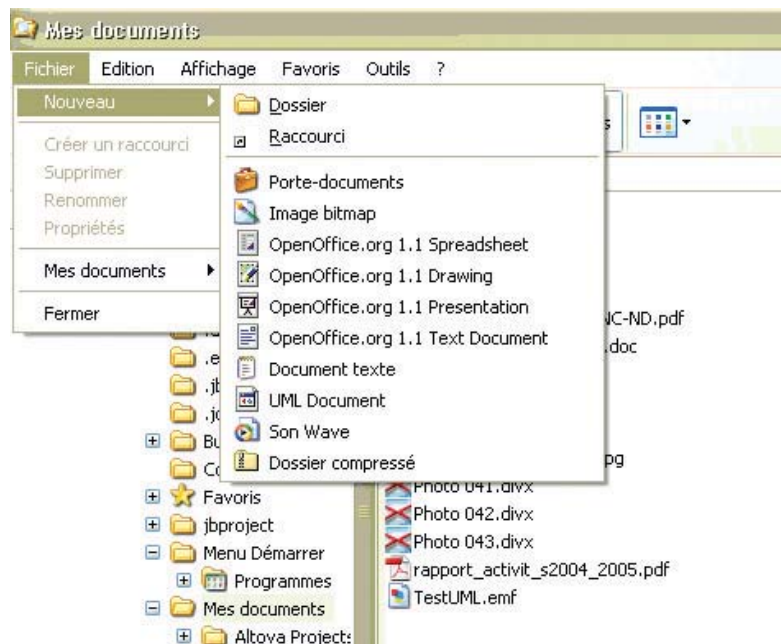


Fig.17: Création d'un nouveau dossier

Activité 13

L'objectif de cette activité est la pratique de quelques opérations liées à la gestion des répertoires

Indications :

- Choisissez le répertoire **MonRep** créé dans l'activité précédente et le copiez sur une disquette.
- Renommez le dossier **MonRep** contenu dans la disquette sous le nom **MonNouveauRep**
- Copiez le dossier **MonRep** sur le Bureau (En utilisant Copier/coller)
- Supprimez le dossier **MonNouveauRep** de la disquette
- Créez successivement les répertoires suivants sous le répertoire **MonRep : Cours, Exercices.**

VII-4 La recherche des fichiers et des dossiers

La recherche de fichiers et de dossiers est une fonctionnalité importante dans la mesure où elle permet de localiser rapidement ces ressources sur les supports de stockage. Sous Windows, la recherche peut s'effectuer en fonction de plusieurs critères dont :

- Le nom du fichier ou du répertoire : Il est possible d'utiliser les caractères «jokers» * et ?. L'astérisque (*) permet de remplacer un ensemble de caractères et le point d'interrogation (?) permet de remplacer un seul caractère.
- La date de modification du fichier.
- La taille.

La procédure de recherche se fait de la manière suivante :

1. A partir du menu Démarrer, pointez sur Rechercher, puis cliquez sur **Des fichiers ou des dossiers.**
2. Répondez à la question «**Que voulez-vous choisir ?**» en cliquant sur **les fichiers et tous les dossiers.**
3. Tapez le nom complet ou une partie du nom dans le formulaire de recherche, puis cliquez sur Rechercher. Les résultats de la recherche s'afficheront dans la partie droite de la fenêtre.

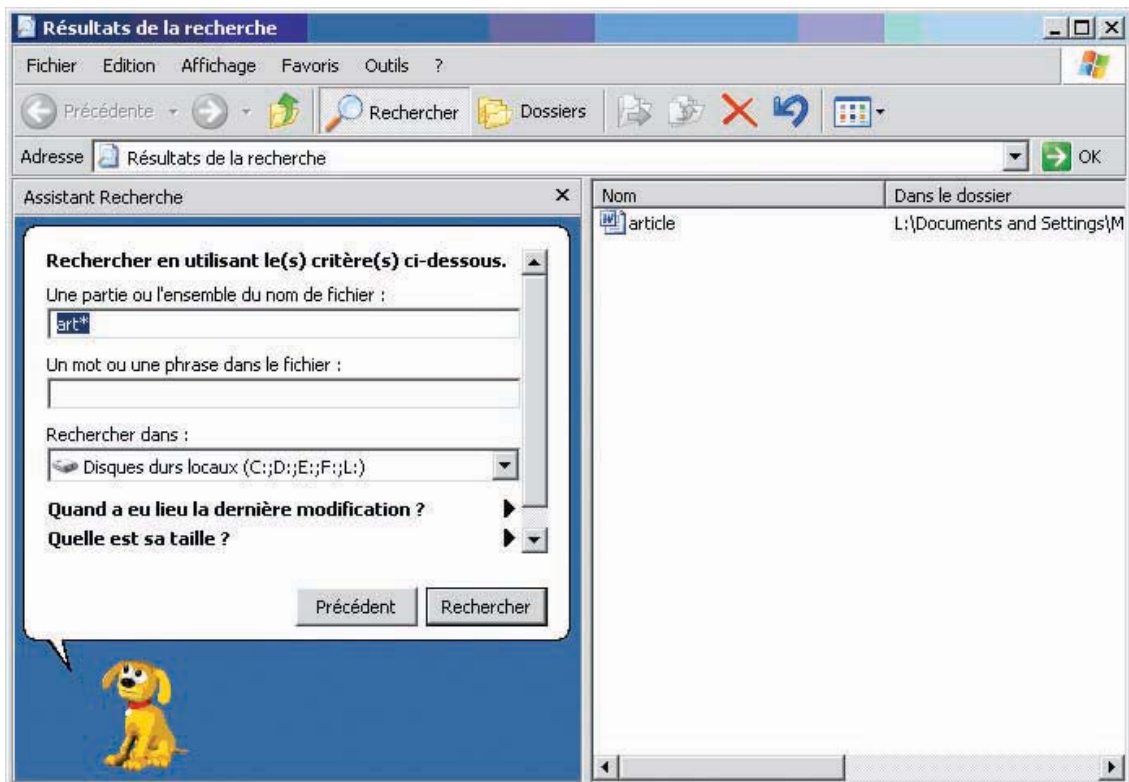


Fig. 18: Recherche des fichiers ou dossiers

Activité 14

L'objectif de cette activité est la pratique de quelques opérations liées à la recherche de fichiers en fonction de divers critères.

Indications :

- Recherchez tous les documents (ou fichiers) Microsoft Word (extension .doc) présents sur votre disque C.
- Recherchez tous les fichiers (quelque soit leurs types) qui ont été modifiés depuis 2 jours.
- Recherchez tous les fichiers (quelque soit leurs types) dont la taille dépasse les 10 Mo.

Retenons

- Le bureau est une interface graphique qui facilite l'accès aux applications et aux services du système d'exploitation. Le bureau offre notamment un lanceur (le menu démarrer) permettant de lancer des programmes.
- Le panneau de configuration est une interface qui regroupe les outils d'administration et de configuration du système.
- L'explorateur est un programme qui permet d'afficher l'organisation arborescente de l'ensemble des fichiers et des répertoires contenus dans un support de stockage. Il permet aussi de faciliter les manipulations sur ces éléments.

Exercices

Exercice 1

Répondre aux questions suivantes :

- Qu'est-ce qu'un système d'exploitation ?
- Qu'est-ce qu'un processus ? Quelle est la différence entre un programme et un processus ?
- Quelle est la différence entre un fichier et un dossier (répertoire) ?
- A quoi sert la barre des tâches ?
- Citez un élément matériel que l'on peut configurer à partir du panneau de configuration.

Exercice 2

Indiquez si les propositions suivantes sont correctes ou non.

| Proposition | Correcte/Incorrecte |
|--|---------------------|
| Avant de démarrer l'ordinateur, le système d'exploitation réside en mémoire centrale RAM. | |
| Le système d'exploitation permet de lancer plusieurs applications en même temps. | |
| Les périphériques sont gérés par le système d'exploitation à travers des modules logiciels appelés pilotes ou drivers. | |
| Le système d'exploitation permet à plusieurs utilisateurs d'ouvrir des sessions de travail simultanées sur le même ordinateur. | |
| Un utilisateur ne peut pas changer son mot de passe. | |
| Un fichier doit obligatoirement posséder une extension. | |
| Il est conseillé d'éteindre toujours son ordinateur à partir du menu démarrer. | |
| Tous les systèmes d'exploitation reposent sur les mêmes concepts et offrent des services similaires. | |
| La famille des systèmes d'exploitation Windows (2000, XP, ...) sont des systèmes d'exploitation libres et gratuits. | |

Exercice 3

L'objectif de cet exercice est la pratique de quelques opérations usuelles sur les fichiers et les répertoires.

1. Créez un dossier appelé «**Rapports**» directement sous le disque dur **C:**
2. Lancez l'application Word et créez un nouveau document appelé «**MonExemple1.doc**» et l'enregistrez sous le dossier «**Rapports**».
3. Créez un nouveau document, à partir de l'application Word, appelé «**MonExemple2.doc**» et l'enregistrez également sous le dossier «**Rapports**».
4. Renommez le fichier «**MonExemple2.doc**» sous le nouveau nom «**MonExemple.doc**».
5. Déplacez le fichier «**MonExemple1.doc**» sous le Bureau.
6. Pratiquez la recherche de tous les fichiers commençant par «**Mon**» sur le disque dur **C:**
7. Supprimez le fichier «**MonExemple1.doc**» puis la totalité du dossier «**Rapports**».

Exercice 4

Complétez le texte ci-dessous en plaçant au bon endroit les termes suivants :

***Bureau, périphériques, démarrage, disque dur, arrêter,
l'icône, système d'exploitation, exécution, enregistre.***

Ahmed met en marche d'abord son ordinateur en appuyant sur le bouton de démarrage électrique. L'ordinateur commence alors son processus de en testant rapidement les puis il charge le système d'exploitation dans la mémoire centrale RAM à partir du A la fin du démarrage, le système d'exploitation offre une interface graphique (le) à Ahmed à partir de laquelle ce dernier peut commencer son travail.

Ahmed veut aujourd'hui faire quelques corrections sur un document Word qu'il avait créé auparavant. Ce document se trouve dans le dossier MesDocuments visible sur le Bureau. Ahmed ouvre ce dossier et clique sur représentant le document. L'application Word est alors lancée, et le document Word est ouvert. C'est qui a assuré ce travail. Ainsi, l'application Word a été chargée à partir du disque dur et elle réside alors en mémoire centrale RAM au moment de son exécution.

Ahmed effectue ses modifications etson document puis quitte l'application Word. Le système d'exploitation assure l'écriture de la nouvelle version du document sur le disque dur, puis se charge de l'arrêt de l'application Word.

Pour éteindre l'ordinateur, Ahmed choisit l'option du menu démarrer, puis confirme l'arrêt. Le système mettra fin à l' de tous les programmes, de manière propre, puis éteindra l'ordinateur.



Chapitre 4

Les réseaux informatiques

Objectifs :

- .Connaitre les différents types de réseaux
- .Travailler dans un environnement réseaux
- .Identifier et utiliser les services Internet

Plan du chapitre :

Leçon 1:
Architecture générale d'un réseau informatique

Leçon 2:
Exploitation d'un réseau

Leçon 3:
Le réseau Internet





Leçon 1

Architecture générale d'un réseau informatique

Objectif spécifique

Connaître les différents types de réseaux.

Plan de la leçon

I- Introduction

II- Les topologies

1. Topologie en étoile
2. Topologie en bus
3. Topologie en anneaux

III- Les équipements d'un réseau

1. Equipements matériels
2. Equipements logiciels

Exercices

Leçon 1

Architecture générale d'un réseau informatique

I- Introduction

Nous avons vu dans les chapitres précédents comment exploiter un micro-ordinateur et ses périphériques tels que l'imprimante, le scanner, le lecteur de CDROM ou de DVDROM, etc. Cette exploitation monoposte ou individuelle ne favorise pas le travail en commun ou en groupe, l'échange d'idées, le partage de ressources, etc. En contre partie, il est très onéreux d'équiper chaque poste par tous les périphériques dont on a besoin comme les imprimantes laser, les scanners, les tables traçantes, etc. Il arrive parfois qu'un cours tourne autour d'un document donné et il est impératif d'en faire des copies pour tous les postes ou faire circuler le support qui le contient à tous les postes. Ces opérations répétitives sont coûteuses et font perdre beaucoup de temps. L'idée de mettre ses machines

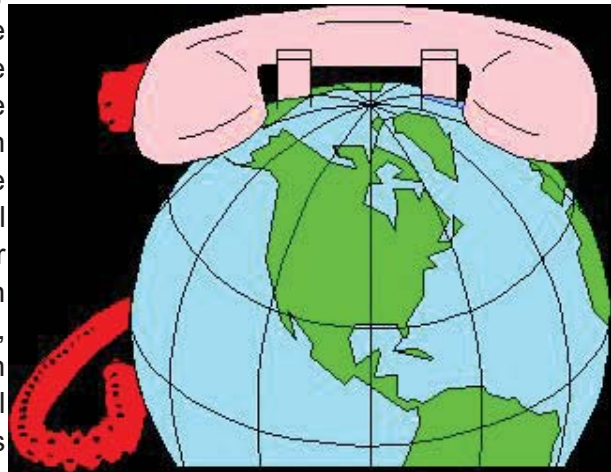


Fig. 1 : Image du réseau mondial

en réseau permet de résoudre plein de problèmes. Nous verrons au fur et à mesure qu'on progresse dans le chapitre, le grand intérêt des réseaux dans l'enseignement et dans la vie quotidienne des citoyens. Pensez aux gains de temps et d'énergie dans le travail d'une entreprise, au courrier électronique, à la téléphonie mobile, etc. Ceci devrait vous donner une première idée sur les avantages des réseaux. Un réseau peut être réduit à quelques machines et couvrant une zone limitée ou assez étendue en nombre de machines connectées. En voici les deux définitions s'y rapportant.

Réseau local ou LAN (Local Area Network)

C'est un ensemble d'ordinateurs et de périphériques connectés entre eux sur un site géographique réduit.

Exemple : Le réseau local de votre laboratoire ou de votre lycée, le réseau local d'une petite entreprise.

Réseau étendu ou WAN (Wide Area Network)

C'est un réseau d'ordinateurs et de périphériques situés sur lieux géographiquement éloignés.

Exemple : Réseau d'une entreprise multinationale ayant des filiales reliées entre elles par ce réseau.

Nous citons aussi Internet comme étant le plus grand réseau mondial étendu. Nous lui consacrerons une leçon à la fin de ce chapitre ?

Activité 1

1. Explorez la disposition des différents composants informatiques dans votre laboratoire. Questionnez votre enseignant pour connaître les noms et les rôles de chacun.
2. Reconnaissez les éléments suivants :
 - les câbles de connexion,
 - les fiches de connexion,
 - l'emplacement de connexion à votre ordinateur,
 - le concentrateur ou hub vers où vont tous les câbles.

II- Les topologies

Discutez avec votre professeur et imaginez différentes manières de connecter des ordinateurs et des périphériques entre eux pour en faire un réseau ?

La disposition des machines en réseau et la manière de communiquer entre elles, est appelée topologie.

II-1 Topologie en étoile

Dans une topologie en étoile, tous les éléments sont connectés entre eux par des câbles en paires torsadées appelées communément 10BaseT. Les fiches de connexion au bout de ces câbles ressemblent à celles du téléphone ; elles sont appelées des fiches de type RJ45. Ces câbles convergent vers un concentrateur ou un hub. La longueur d'un tel câble de connexion ne doit pas dépasser les 100 mètres. L'avantage d'une telle configuration est sa non altération quand une machine du réseau est non fonctionnelle ou tombe en panne. Le débit dans un tel réseau est de l'ordre de 10 mégabits par seconde (10 MB/s).

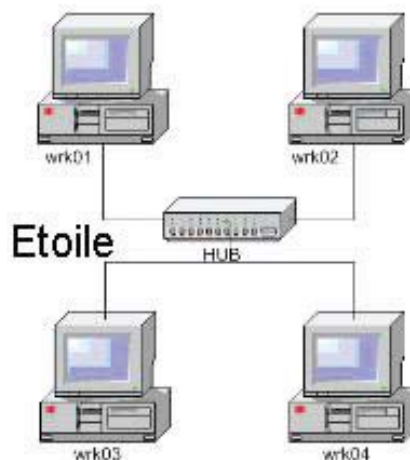


Fig.2 : Schéma d'un réseau en étoile

II-2 Topologie en bus

Dans une topologie en bus, les éléments sont disposés formellement sur une ligne par l'intermédiaire d'un câble coaxial connu sous le nom 10Base2. Les connexions se font avec des fiches BNC. Ce réseau est caractérisé par une longueur réduite d'un segment du réseau. En effet, la longueur d'un segment ne doit pas dépasser 185 mètres faute de quoi le signal deviendrait trop faible pour être convenablement reçu. Au besoin et pour remédier à cette défaillance, il faudra intercaler des répéteurs pour amplifier le signal et le renvoyer dans la ligne. Deux inconvénients altèrent la simplicité d'un tel réseau : les collisions fréquentes et l'arrêt total du réseau quand l'une des machines tombe en panne.

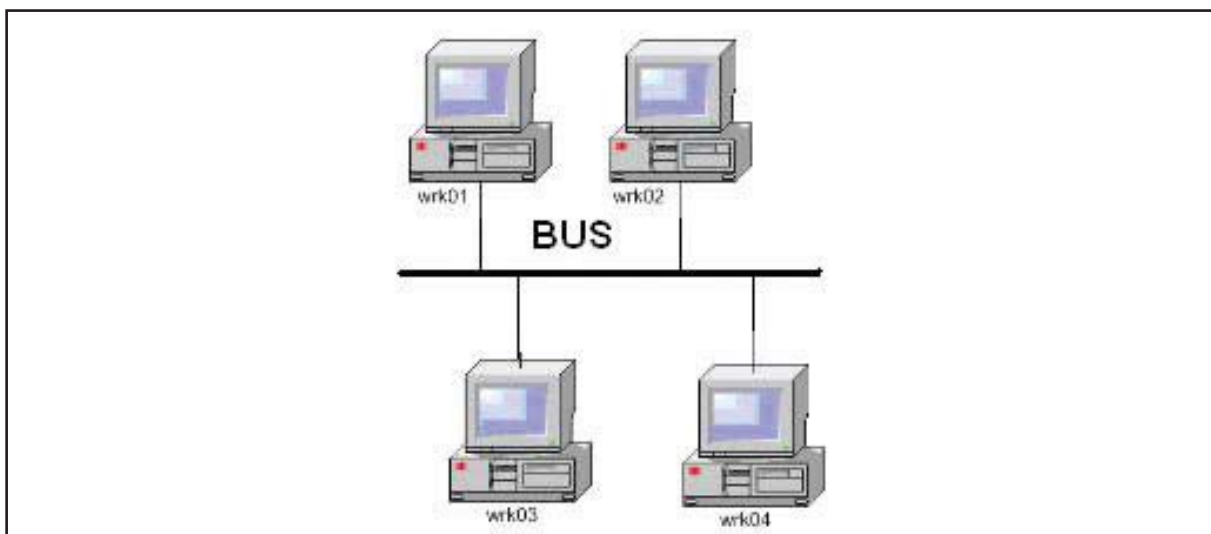


Fig.3 : Schéma d'un réseau en bus

II-3 Topologie en anneaux

Dans une topologie en anneau, les machines sont connectées entre elles et disposées physiquement autour d'un anneau. La transmission se fait à l'aide d'un jeton permettant de reconnaître la machine "destinataire". Le câblage est le même que pour la réseau en étoile. On y utilise la norme 10BaseT ou câble à paires torsadées.

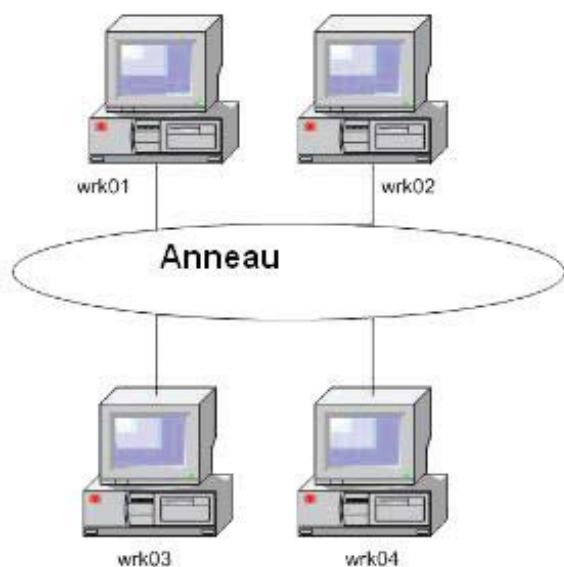


Fig. 4 : Schéma d'un réseau en anneau

III- Les équipements d'un réseau

III-1 Equipements matériels

Pour installer un réseau entre plusieurs machines, il faudra avoir d'abord le matériel de connexion nécessaire. Commençons par regarder le réseau de la salle.

Activité 2

1. Regardez attentivement les diverses connexions entre les machines de votre laboratoire.
2. Repérez les éléments suivants :
 - le câble et la fiche permettant de relier votre poste aux autres machines.
 - la carte ou l'interface à laquelle est branchée le câble réseau de votre ordinateur.
 - vers où vont tous les câbles de la salle ?
 - l'emplacement de certains périphériques comme l'imprimante et le scanner et sur quelle machine sont-ils branchés ?

Essayez de reconnaître les éléments précédents parmi les objets des figures suivantes:



Fig.5 : Une carte réseau à différentes connexions RJ45, BNC et série



Fig. 6 : un câble en paires torsadées terminées par une fiche RJ45



Fig. 7 : Une armoire de brassage

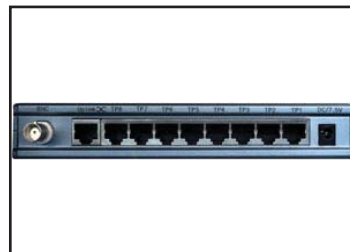


Fig.8 : un concentrateur ou hub

III-2 Equipements logiciels

Après avoir câblé toutes les machines, il faudra ensuite installer dans chacune de celles-ci les protocoles de communication qui leur permettront de dialoguer ou de communiquer. Ces protocoles sont des programmes informatiques qui permettent à deux ou plusieurs ordinateurs mis en réseau de communiquer et d'échanger des données. Il en existe plusieurs, nous en citons :

- NetBIOS (Net Bios Input Output System) protocole réseau créé par IBM
- NetBEUI (Net Bios Extended User Interface) une extension du protocole NetBIOS
- IPX/SPX protocole NetWare de Novell
- TCP/IP (Transmission Control Protocol / Internet Protocol) une suite de protocoles sur lesquels sont basés les réseaux intranet et le réseau mondial Internet. Le réseau Internet fera l'objet de la dernière leçon de ce chapitre.

Attention

Retenez que quelque soit le protocole utilisé, chaque machine du réseau doit avoir un identifiant unique lui permettant d'être connu sans ambiguïté dans le réseau. Cet identifiant peut être un nom ou un nombre ou un numéro. L'essentiel est l'unicité de cet identifiant pour pouvoir repérer sans confusion n'importe quelle machine branchée au réseau.

Activité 3

- Ouvrez le panneau de configuration de votre machine (Voir Fig.9 ci-contre)
- Lancez la rubrique relative au réseau local de votre machine.
- Sélectionnez la carte réseau existante et explorez ses propriétés.
- Quelle est l'adresse de votre machine dans le réseau ?

Attention

Ne changez rien aux propriétés et aux paramètres du réseau sans aviser votre professeur.

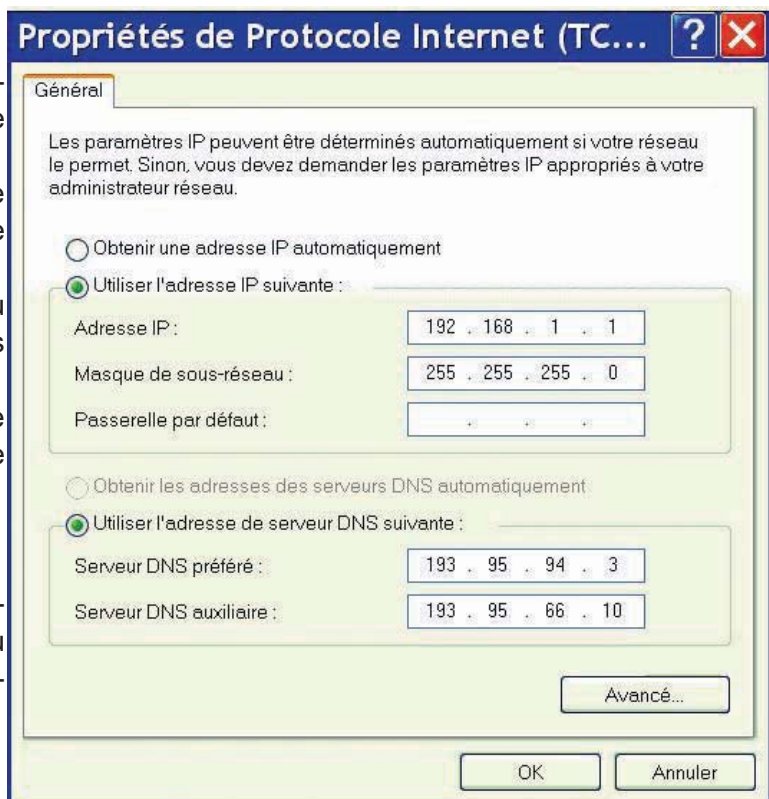


Fig 9: Adresse IP



Exercices

Exercice 1

Qu'appelle-t-on topologie ?

Quels sont les avantages d'une topologie en étoile ?

Quelle est la topologie de votre réseau local au laboratoire d'informatique ?

Exercice 2

Quelle est la logistique nécessaire en matière de logiciels pour que les machines d'un réseau puissent communiquer entre elles ?

Exercice 3

Citez tous les composants matériels qui ont servi à réaliser un réseau local dans votre laboratoire.



Leçon 2

Exploitation d'un réseau

Objectif spécifique

Travailler dans un environnement réseau.

Plan de la leçon

- I- Partage de documents
- II- Partages d'imprimantes et de scanners
- III- Envoi de messages
- IV- Exploitation d'applications réseau

Exercices

Leçon 2

Exploitation d'un réseau

I- Partage de documents

Un des grands avantages de travailler en réseau est de faciliter et d'encourager le travail en commun, de communiquer à distance, d'exploiter des ressources partagées et donc de faire des économies. Dans cette leçon, nous allons découvrir et pratiquer le partage de fichiers et de dossiers, l'exploitation en commun d'imprimantes et de scanner, l'exploitation d'applications sur le réseau. Nous établirons des communications par des messages. Ces avantages sont offerts par les réseaux locaux et les réseaux étendus.

Activité 1

1. Créez un dossier intitulé DOCS_PARTAGES à la racine de la partition D ou C.
2. Mettez dans ce dossier un jeu de fichiers comportant des images et des sons choisis selon votre goût.
3. Partagez ce dossier en lecture et en écriture avec libre accès et invitez vos camarades à y accéder pour en prendre des copies dans leurs machines.

Voici les étapes à suivre pour réaliser l'activité précédente.

- 1) Commencez par créer un répertoire tel qu'indiqué précédemment. Il est préférable d'utiliser une partition autre que celle contenant le système d'exploitation. Demander à votre professeur d'en créer une s'il n'y a que la partition C.
- 2) Cherchez dans les images et les sons de votre ordinateur, ceux qui vous plaisent et que vous souhaitez partager avec les autres élèves de la classe. Prenez-en des copies dans le répertoire que vous avez créé.
- 3) Affichez les propriétés du dossier en question et changez les paramètres de partage de telle façon qu'il devient partagé avec un accès total. (voir Fig.1)
Vous pouvez vous assurer de ce partage en explorant dans le voisinage réseau le contenu de votre poste.

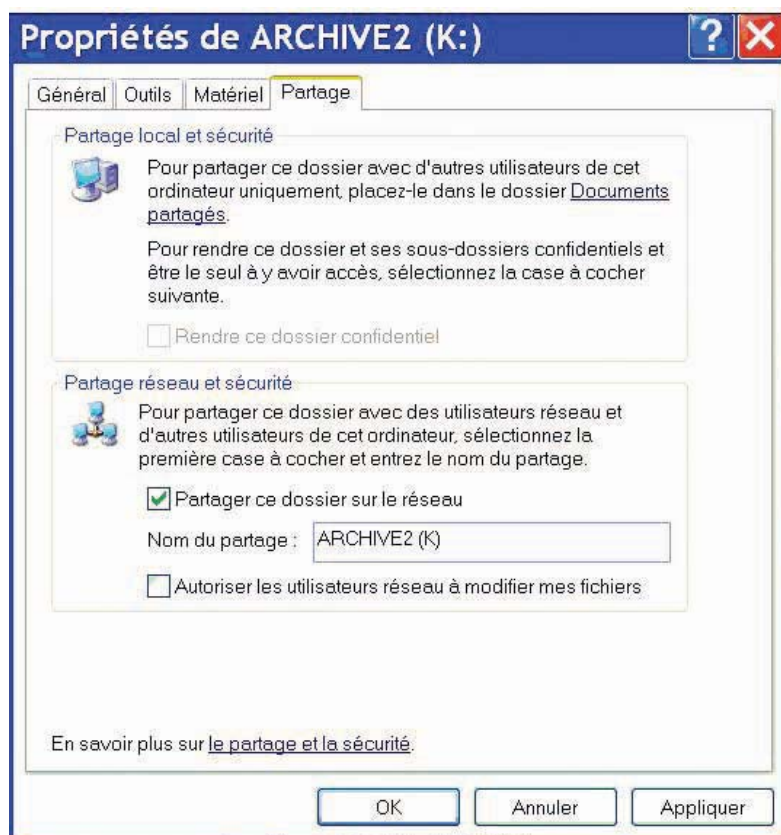


Fig. 1 : Partage d'une partition numérotée K et appelée ARCHIVE2

II- Partage d'imprimantes et de scanners

Pour partager une imprimante ou un scanner sur un réseau, il faut d'abord les installer sur l'un des postes en tant que périphériques fonctionnant sur celui-ci. Ensuite, il est impératif d'installer les pilotes de cette même imprimante ou scanner sur les autres machines en indiquant que ce périphérique est branché sur une autre machine du réseau local.

Activité 2

1. Sur quel port est branchée l'imprimante ?
2. Lancez le logiciel de traitement de textes et produisez un document contenant votre nom, votre prénom ainsi que votre classe.
3. Lancez la commande Imprimer du menu Fichier et reconnaissez l'imprimante destination.
4. Lancez l'impression de cette page.
5. Allez récupérer votre feuille.

Activité 3

Si vous disposez d'un scanner exploitable sur un réseau, demandez à votre enseignant de le brancher sur l'une des machines du réseau et de le faire partager.

- 1) A partir de votre poste, retrouvez le périphérique en question.
- 2) Toujours à partir de votre poste, scannez une image de votre choix.
- 3) Affichez l'image obtenue avec une application de traitement d'images.



Fig. 2 : Scanner

III- Envoi de messages

Le réseau permet aussi le dialogue avec les autres personnes qui y sont branchées. Imaginez que vous êtes dans une entreprise ou une institution relativement spacieuse et qu'un employé a besoin d'une information chez l'un de ses collègues. En l'absence de réseau, il est obligé de quitter son poste pour acquérir cette information. Cette perte de temps n'aura pas lieu si ces employés disposent d'un réseau local qu'ils exploitent à bon escient. Pour se faire, il existe plusieurs applications permettant l'échange de messages. Nous en citons celles qui sont souvent jointes aux systèmes d'exploitation. Par exemple, les versions Windows ont les applications PopUp et NetMeeting. La deuxième est beaucoup plus exigeante sur le plan installation mais elle est assez variée du point de vue possibilités d'échange. Les figures ci-dessous montrent les diverses options de communications offertes par NetMeeting.

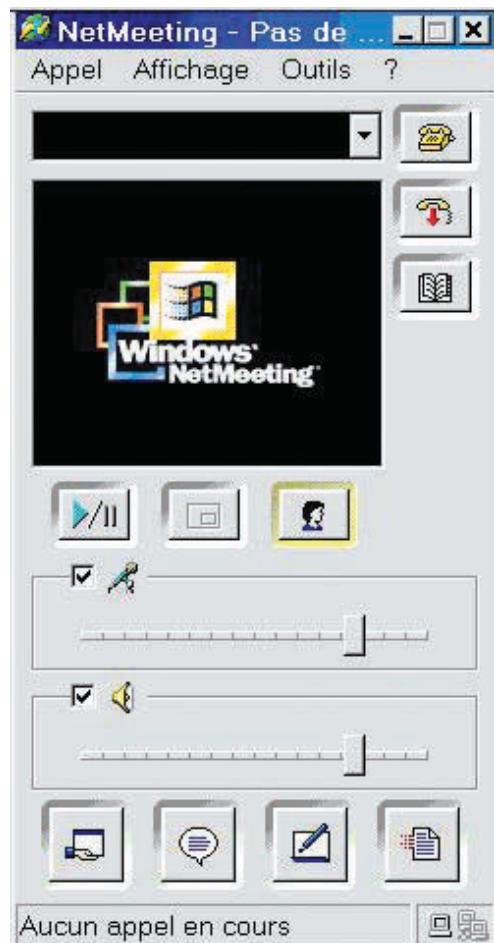


Fig. 3 : Fenêtre du lancement de NetMeeting

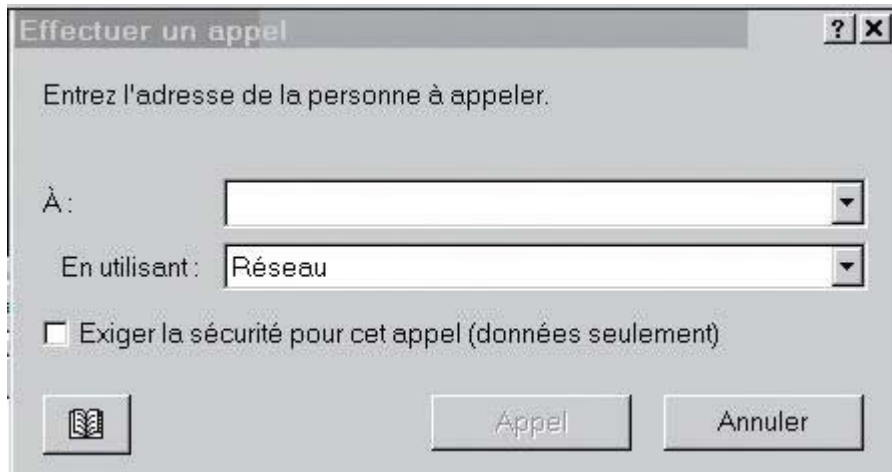


Fig. 4 : Fenêtre du lancement de NetMeeting

Activité 4

1. Demandez à votre enseignant de configurer l'application de messagerie et utilisez-la pour échanger des messages et des documents avec vos camarades de la classe.
2. Sur un papier, énumérez les avantages et les inconvénients de cette opération ?
3. Discutez-en avec votre enseignant.

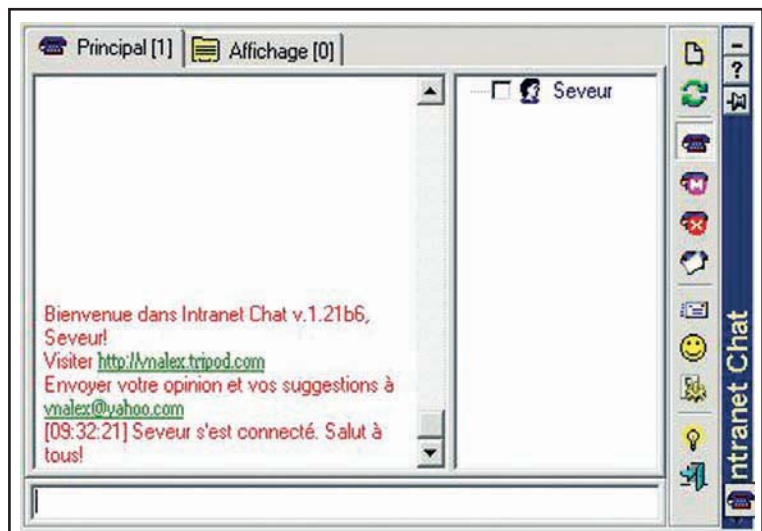


Fig.5 : Fenêtre d'envoi d'un message

IV- Exploitation d'applications réseau

Il existe plusieurs applications permettant d'exploiter le réseau et d'aider les utilisateurs à communiquer entre eux. Certaines sont dédiées spécialement à l'enseignement et plus particulièrement à l'apprentissage. Elles permettent parfois des opérations d'évaluation. Dans la plupart des cas, ces applications fonctionnent en mode client / serveur. Une des machines du laboratoire est choisie comme serveur et toutes les autres sont des clients. La personne travaillant sur le serveur a une vue générale sur tous les autres postes. Cet utilisateur est appelé l'administrateur. C'est le cas de votre enseignant qui administre la gestion du réseau et la machine sur laquelle il travaille fait fonction de serveur. En effet, à partir de son poste serveur, l'enseignant peut réaliser plusieurs tâches, nous en citons quelques unes :

- Avoir un accès visuel à tous les écrans
- Envoyer à tous les postes ou à quelques uns qu'il choisit, les documents ou l'écran qu'il veut. Cette opération permet de faciliter la communication avec les élèves. Elle a tendance à remplacer d'autres supports assez classiques par exemple le tableau ou le papier.
- Mettre les notes de cours à portée de l'apprenant pour une éventuelle copie sur le cahier.
- Accéder directement au document sur lequel l'élève est en train de travailler et pouvoir intervenir pour corriger et apporter une aide.
- Synchroniser l'apprentissage avec une vue globale de tous les écrans.
- Assurer des tests d'évaluation.
- Gérer la progression du cours et l'utilisation du matériel informatique disponible.

Activité 5

1. Demandez à votre enseignant de lancer l'application réseau existante et de vous montrer la gestion des postes clients.
2. Expérimentez avec lui les différentes possibilités et les grands profits de l'utilisation d'une telle application.

Activité 6

1. Demandez à votre enseignant de vous donner un test et de le limiter dans le temps.
2. Expérimentez l'opération d'évaluation si possible.

Exercices

Exercice 1 (Pratique)

Créez à la racine de la partition de votre machine réservée aux travaux, un dossier MES_IMAGES et mettez y vos images préférées.

Dites comment faire pour partager ce dossier et réalisez cette opération ?

Exercice 2 (Pratique)

Comment faire pour :

- Envoyer un "bon jour" à tous les camarades de votre classe ? Réalisez cette tâche ?
- Recensez les dossiers partagés dans une machine donnée du réseau ? Réalisez cette tâche avec une machine voisine ?

Exercice 3

Pourquoi fait-on le partage d'une imprimante ?

Qu'est ce qui se passe quand tous les utilisateurs du laboratoire envoient des documents pour la même imprimante partagée ?

Exercice 4

Quels sont les avantages d'une application réseau dédiée à l'apprentissage ?



Leçon 3

Le réseau Internet

Objectifs spécifiques

Identifier et utiliser les services Internet

Plan de la leçon

I- Partage de documents

1. Une présentation brève
2. Les protocoles d'Internet
3. Les adresses IP
4. Les adresses significatives URL
5. La logistique nécessaire pour se connecter à Internet

II- La recherche d'informations sur le web

III- La messagerie

IV- Les forums de discussions

Exercices

Leçon 3

Le réseau Internet

Nous avons vu dans les leçons précédentes, l'utilité d'un réseau local. Imaginons alors l'intérêt que peut apporter un réseau étendu où des millions de personnes autour du globe viennent participer avec leurs idées, leurs cultures, leurs savoirs, leurs savoir-faire et tout ce qui pourrait exister sur terre. Un tel réseau existe bel et bien et devient une passerelle incontournable vers le développement des pays. Ce réseau s'appelle Internet.

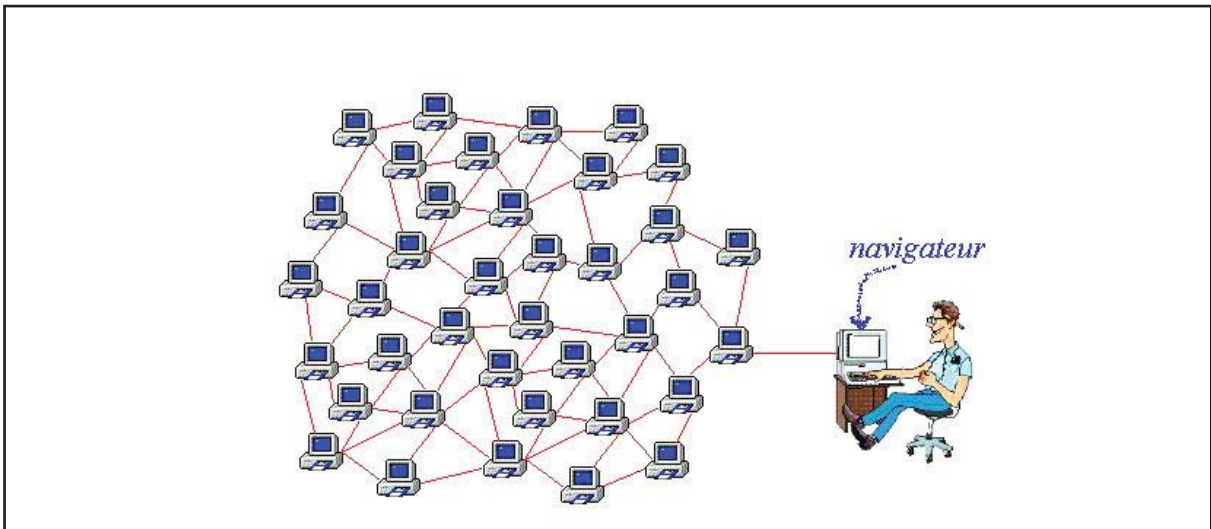


Fig. 1 : Schéma du réseau mondial Internet

I- Partage de documents

I-1 Une présentation brève

Internet est un réseau mondial assurant la connexion et la communication entre un grand nombre de réseaux. Il permet donc d'interconnecter des millions de machines dans le monde. Son histoire a commencé aux Etats Unis d'Amérique en 1970 et plus précisément à l'agence de développement militaire l'ARPA. L'idée est de créer un réseau qui ne sera pas altéré si l'un de ses nœuds serait détruit. Cette idée a donnée naissance au fameux protocole d'Internet nommé TCP/IP. Les universitaires l'ont ensuite exploité pour connecter plusieurs universités et c'est le début de la propagation de ce réseau. Vers les années 1980, Internet est désormais connu et opérationnel dans plusieurs pays. L'exploitation de ce réseau devient possible aux particuliers. En même temps, le tas d'informations mis à disposition devient colossal et ne cesse d'augmenter de jour en jour. En plus, Internet a offert plusieurs services dont quelques-uns sont devenus des appropriations essentielles pour les particuliers, les entreprises et les diverses institutions. Ils ont même changé et pour beaucoup notre mode de vie. Nous en citons le web, la messagerie, les forums de discussion ainsi que divers moyens de communications.

I-2 Les protocoles d'Internet

Avant de connecter une machine au réseau, il faudra d'abord la faire équiper d'un ensemble de règles de communications lui permettant de s'identifier sur le réseau et de pouvoir converser avec les autres éléments du réseau. Ces règles constituent un protocole de communications en réseau. Le protocole utilisé par Internet s'appelle TCP/IP. Il est réellement un ensemble de protocoles dont chacun est chargé d'une activité bien déterminée comme la navigation sur le web, le téléchargement, la messagerie entrante et sortante, etc.

I-3 Les adresses IP

Il est tout à fait évident que pour pouvoir repérer une machine sur un réseau, il faut lui attribuer un identifiant ou une adresse. Cette adresse doit être unique pour que cette machine soit identifiée et connue sans ambiguïté sur le réseau. Pour un réseau utilisant le protocole TCT/IP comme Internet, il a été convenu de former une telle adresse par une suite de quatre octets. Comme chaque octet peut générer un entier compris entre 0 et 255, vous reconnaissez alors les adresses de la forme : 193.95.126.12 ou 205.45.76.87 ou encore 127.0.0.1 que vous avez dû voir dans les livres, sur les journaux ou ailleurs.

Ces adresses ne sont pas livrées au hasard, elles sont rangées par classes de réseaux suivant la valeur de l'octet le plus à gauche. Ces classes sont nommées A, B, C, D ... Chacune regroupe une plage d'adresses et a ses propres caractéristiques.

Remarquons que cette convention d'adressage est arrivée à une situation de saturation et elle ne peut résorber les nouveaux réseaux et machines qui se connectent chaque jour à Internet. L'idée est alors d'augmenter le nombre d'octets dans l'adresse d'une machine connectée à Internet. La version sur 6 octets est déjà entrée en application pour décongestionner cette saturation. IP6 ou version 6 est déjà à l'œuvre.

I-4 Les adresses significatives URL

Nous savons que ces machines connectées au réseau Internet comportent des serveurs web, de messageries ou autres. A priori, il faudra se rappeler des adresses numériques de toutes machines auxquelles nous voulons accéder. Cette pratique est fastidieuse car il n'est pas facile de se rappeler de toutes ces suites de nombres entiers pour retrouver les serveurs ou les machines à contacter. L'idée est d'utiliser des noms ou des appellations significatives faciles à retenir puis de trouver un moyen de chercher par ordinateur et d'une façon automatique les adresses numériques correspondantes. Cette tâche a été confiée à des serveurs dits DNS (Domain Name Server). Bien entendu, ces serveurs sont très fréquemment mis à jour pour pouvoir suivre l'évolution de la toile.

Exemples :

| | |
|---------------|--|
| www.edunet.tn | correspond à l'adresse IP : 193.95.94.35 |
| www.yahoo.fr | correspond à l'adresse IP : 217.12.0.253 |
| www.google.fr | correspond à l'adresse IP : 66.62.9.104 |

Vous avez dû remarquer que les adresses URL se terminent souvent par une extension telles que .tn, .fr, .com, .edu, .net, .mil, .org, etc. Cette extension permet de définir la nature du domaine ou le pays d'origine.

Exemples :

.tn signifie que le site est tunisien comme www.edunet.tn

.fr signifie que le site est français comme www.yahoo.fr

.com signifie que le site est commercial comme www.ibm.com

.org signifie que le site est celui d'une organisation comme www.un.org

I-5 La logistique nécessaire pour se connecter à Internet

La connexion à Internet s'est simplifiée et vulgarisée à tel point qu'un ordinateur et un moyen de communication deviennent suffisants pour le faire. Il suffit de se procurer une carte d'accès à Internet que les fournisseurs mettent en vente dans les kiosques et les points de vente les plus proches de chez vous. Disons que pour ne pas trop simplifier les choses, une bonne machine multimédia, un abonnement à Internet fait auprès d'un fournisseur de services Internet, une ligne de télécommunication RTC, LS ou ADSL et dans certains cas satellitaires permettront de réaliser une telle connexion.

Activité 1

1. Demandez à votre enseignant de vous donner un test et de le limiter dans le temps.
2. Expérimentez l'opération d'évaluation si possible.

II- La recherche d'informations sur le web

Le web, mot anglais signifiant la toile d'araignée, est une description judicieuse des innombrables interconnexions de réseau Internet. Cette toile offre comme nous l'avons mentionné précédemment plusieurs services. L'un des plus sollicités est la recherche d'informations dans les divers sites que les particuliers et les organismes mettent à la disposition des utilisateurs du réseau. Ces informations sont publiées pour diverses intentions et objectifs. Vous y trouvez les informations de caractère informel, documentaire, aide, publicitaire, commercial, utilitaire, etc. Pratiquement, tous les domaines sans exception sont abordés suffisamment et différemment sur le réseau Internet à l'aide de sites web. Un site web est un ensemble de pages comportant des liens dits liens hypertextes permettant une lecture non linéaire de l'ensemble du document du site et éventuellement des sources extérieures sur d'autres sites. La distance physique n'a plus de sens dans cette lecture. Une telle lecture et exploration de sites web s'appelle navigation ou surf en anglais.

Pour surfer ou naviguer sur Internet et dans les sites web, il vous faut un navigateur. Ce dernier est un programme à installer dans votre machine et à configurer à cet effet pour pouvoir se connecter au réseau Internet. Il existe plusieurs logiciels navigateurs sur Internet. Nous en citons les plus connus qu'ils soient commerciaux ou à licences libres : MS Internet Explorer, NetScape Navigator, Opera, FireFox, ...

Si vous avez équipé votre machine du système d'exploitation MS Windows, Internet Explorer est installé automatiquement avec un assistant vous aidant à paramétrer la connexion à Internet ainsi qu'une application pour gérer votre boîte de messagerie. Les autres applications peuvent être installées aussi sous le système d'exploitation Windows.

Nous travaillerons avec le navigateur disponible. Et de toutes les façons, quand vous lancez un navigateur, vous allez avoir une interface qui ressemble à l'une de ces images.

Vous connaissez la source de l'information à chercher. En général, vous avez l'adresse



Fig. 2 : Fenêtre de l'explorateur Internet Explorer



Fig. 3 : Fenêtre de l'explorateur FireFox

URL. Elle est sous la forme `www.serveur_offrant_le_service/chemin/nom_fichier.htm`. Il suffit de l'écrire dans la zone d'adresse juste après l'instruction indiquant le protocole à utiliser. `http:\\` http signifie que la navigation utilise le protocole hyperText Transmission Protocol.

Dans le cas où vous n'avez pas d'idée sur les adresses où vous allez puiser l'information, vous pouvez utiliser des sites qui la chercheront pour vous. Ces sites contiennent en fait des applications web appelées moteurs de recherche. Ces sites répertorient pratiquement tout ce qu'il y a sur la toile de telle façon qu'ils peuvent répondre dans les plus fins détails à n'importe quelle question ou requête.

Il existe plusieurs moteurs de recherche sur le net. Les plus connus sont : Google, Altavista, msn, lycos, yahoo, voila, ... Ils font indexer pratiquement toutes les pages du web. Certains moteurs, compte tenu de la façon avec laquelle ils indexent les sites, s'appellent plutôt annuaire comme Yahoo, par exemple. En fait, ils indexent plutôt les sites et les classent par catégories.

La recherche se fait via un formulaire à remplir. Plus précisément vous remplissez ce formulaire, meilleurs seront les résultats retournés.



Fig. 4 : La fenêtre Options dans le navigateur Internet Explorer

Activité 2

1. Utilisez un moteur de recherche pour retrouver de l'information relative à la division euclidienne. Explorez les 20 premiers sites retournés comme résultats. Enregistrez les pages dont vous trouvez le contenu intéressant.
2. Faites le même travail pour chercher des documents sur le malouf en Tunisie.

Remarques :

- Si le texte, objet de votre recherche, comporte plusieurs mots, le résultat renvoyé par un moteur commence généralement par vous donner les sites ou les pages comportant la totalité de ce texte jusqu'aux pages comportant un seul mot de cette suite.
- Si vous tenez à chercher les pages comportant la totalité du texte, il faudra le mettre entre guillemets.
- Les opérateurs AND assurent une recherche des sites où figurent conjointement les mots opérands entourant cet opérateur.
- L'opérateur OR assure la recherche des sites ou des pages comportant l'un ou l'autre ou les deux à la fois. Le résultat commence généralement par les pages comportant les deux mots.

III- La messagerie

L'autre service qui s'est bien imposé au point où il est devenu une nécessité, est la messagerie ou encore le courrier électronique. De nos jours, pratiquement chaque internaute dispose d'au moins une adresse électronique de courrier ou encore d'un compte email. D'ailleurs, tous les fournisseurs de services Internet (FSI) délivrent d'office avec l'abonnement à Internet, un tel compte et une adresse de courrier électronique. Aussi, plusieurs serveurs prestataires de services Internet offrent gratuitement aux internautes qui le veulent des comptes électroniques gratuits avec des capacités de stockage avoisinant ou dépassant parfois un gigaoctet.

Voyons comment créer et gérer un compte email.

Sachez d'abord qu'un compte email est personnel. Il est semblable à la boîte postale de votre maison. Quand vous créez un compte chez un FSI ou gratuitement dans un serveur de services Internet, vous précisez alors votre adresse sur ce serveur. Cette adresse est de la forme :

Foulen.EIFoulani@seueur.

Exemples :

Mohamed.benAli@edunet.tn

samira_bentMhamed@yahoo.fr

On vous donne ensuite votre login ou nom de l'utilisateur, votre mot de passe ainsi que les noms des deux serveurs qui se chargeront de la réception et de l'envoi de votre courrier. Le serveur se chargeant de l'envoi s'appelle serveur SMTP, celui qui se charge de la réception s'appelle le serveur POP. POP (Post Office Protocol) et SMTP (Simple Mail Transfert Protocol) sont les deux protocoles permettant respectivement la réception et l'envoi des messages. Les données qui vous ont été livrées, serviront à configurer et paramétrer l'application de gestion de courrier que vous allez utiliser. Parmi ces applications, nous en avons cité l'application qui est installée d'office avec le système d'exploitation MS Windows. Il s'agit d'Outlook Express. Une version plus élaborée de cette application nommée Outlook est jointe à la suite MS Office.

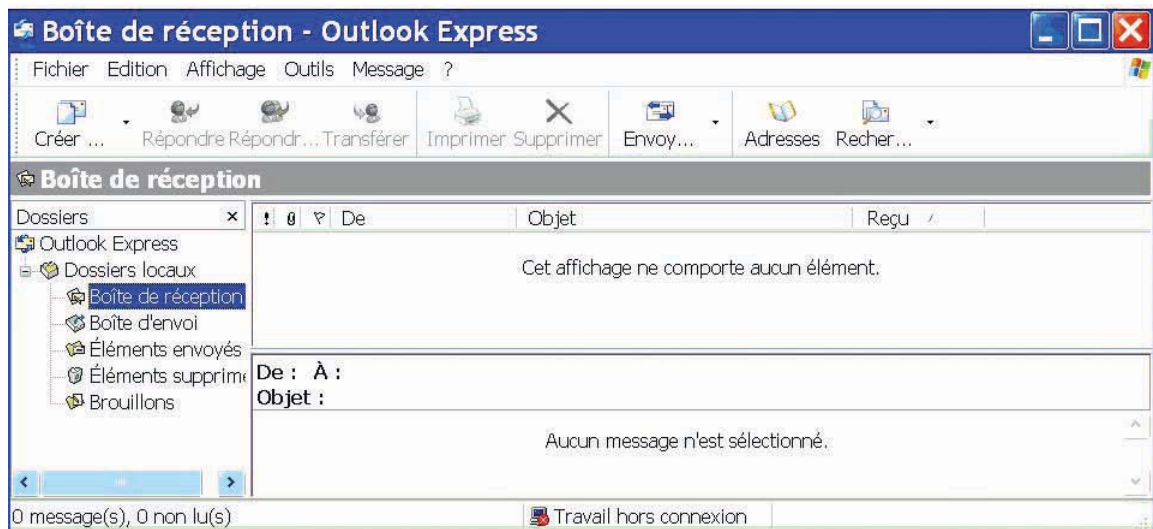


Fig. 5 : Fenêtre de MS Outlook Express

Les comptes offerts gratuitement par des serveurs de prestataires de services sur le net ne donnent pas de noms pour les serveurs de courrier rentrant et sortant. L'application de gestion lancée à partir du serveur se charge de la totalité de l'opération.

Notons un avantage très important d'avoir un compte de messagerie électronique, le fait que vous pouvez accéder à votre courrier de tout point de la terre, pourvu qu'il y ait une connexion à Internet. Ce mode de communication facilite énormément la communication entre les gens.

Activité 3

1. Lancez l'application de gestion de messagerie électronique et explorez son menu et sa configuration.
2. Regardez ensuite cette figure et identifiez le rôle des icônes encerclées.

Activité 4

1. Si vous avez une adresse électronique, utilisez la pour envoyer un message à vos camarades.
2. Mettez les adresses électroniques de ceux qui vont écrire dans votre carnet d'adresses.

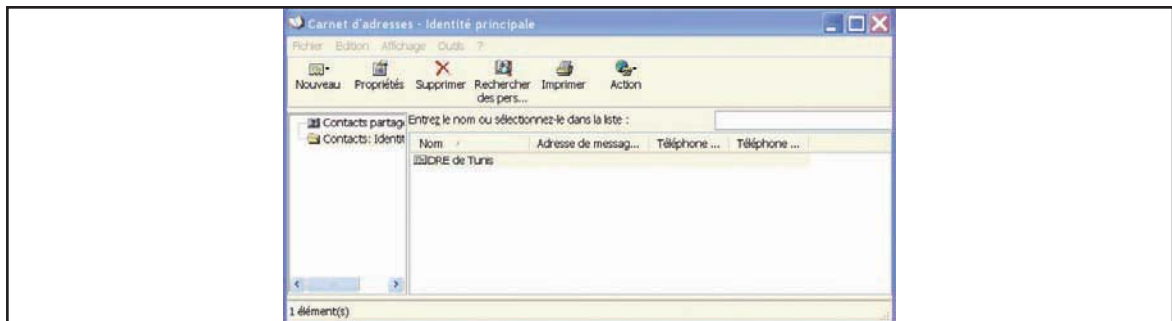


Fig.6

Activité 5

Il vous arrive de trouver dans votre boîte postale des brochures publicitaires. La même opération se fait dans les boîtes électroniques. Discuter avec votre professeur de cette distribution, de son impact et de la notion de spam.

IV- Les forums de discussion

Il arrive fréquemment que quelqu'un bute devant un problème et a besoin d'aide. Internet est la bonne destination. En effet, vous pouvez trouver sur Internet et suivant le thème dans lequel vous souhaitez avoir de l'aide, le coin utile. Il vous suffit d'intégrer le groupe en question et de poser votre question ou votre requête, les réponses ne tardent pas de venir. L'esprit de coopération est très développé sur le Net. Ces cercles d'échange et de discussion s'appellent les forums. Sachez qu'il y a des forums dans pratiquement tous les domaines de la vie. Généralement, ils sont classés par thème. Certains sont modérés pour ne diffuser que les questions et les réponses relatives au sujet de discussion. C'est le modérateur du forum qui décide de la publication des différentes interventions. Vous pouvez y accéder et vous abonner puis participer à la discussion, donner votre point de vue et faire savoir votre opinion.

Pour accéder à un forum, il faudra connaître son adresse électronique ou le "serveur de news" qui l'héberge. Vous pouvez donc y accéder à l'aide d'un navigateur ou d'une application de gestion de messagerie.

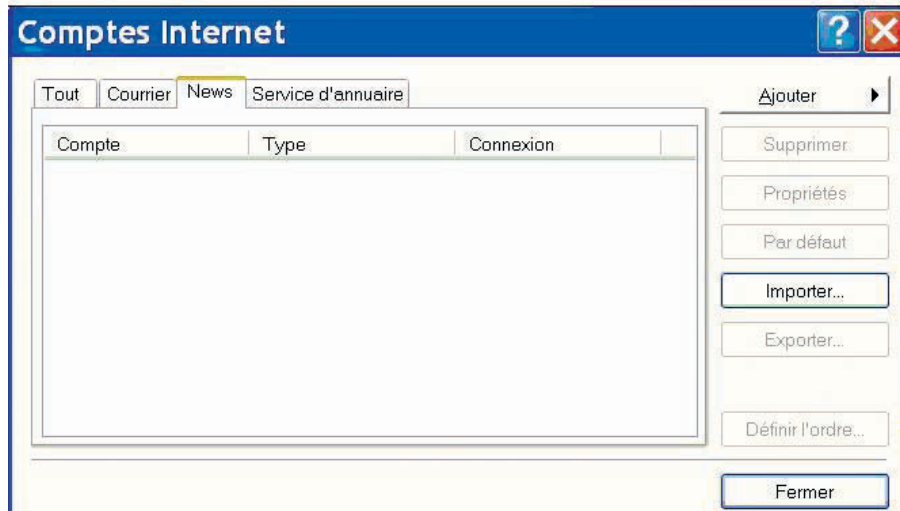


Fig. 6 : Le coin du service News dans l'application Outlook Express

Exercices

Exercice 1 (Pratique)

Cherchez sur Internet des documents simples parlant d'Internet lui-même.

Enregistrez les documents qui vous ont intéressé dans un dossier que vous créez pour cette fin.

Expliquez les raisons de votre choix.

Exercice 2 (Pratique)

Même question que dans l'exercice 1 à propos du thème : Travaux des égyptiens sur le nombre π .

Exercice 3 (Pratique)

Même question que dans l'exercice 1 à propos du thème : histoires des mathématiques au maghreb.

Exercice 4 (Pratique)

Comment faire pour envoyer tout le dossier par mail à un ou plusieurs de vos amis ?

Réalisez cette opération.

Exercice 5

Comment appelle-t-on le protocole utilisé par le réseau Internet ?

Comment appelle-t-on le protocole qui permet :

- l'envoi d'un courrier,
- la réception d'un courrier
- la lecture de pages web par un navigateur

Exercice 6 (Pratique)

Réalisez avec l'aide de votre enseignant un forum et invitez vos camarades en en prendre part pour s'y inscrire et participer aux discussions



Chapitre 5

Démarche de résolution de problèmes

Objectifs :

- Résoudre un problème en utilisant une approche d'analyse.
- Connaître les différentes étapes de résolution.

Plan du chapitre :

Leçon :
Les étapes de résolution
d'un problème



Leçon

Démarche de résolution de problèmes

Objectifs spécifiques

- Résoudre un problème en utilisant une approche d'analyse..
- Connaître les différentes étapes de résolution.

Plan

I- Introduction

II- Première étape : Position du problème

III- Deuxième étape : spécification et Analyse du problème

IV- Troisième étape : Ecriture de l'algorithme

V- Quatrième étape : Ecriture du programme

VI- Cinquième étape : Exécution et tests du programme

Retenons

Exercices

Leçon

Les étapes de résolution de problèmes

I. Introduction

Vous avez dû remarquer tout le long des chapitres précédents que l'utilisation de l'ordinateur se fait à travers des logiciels dédiés à des applications particulières. Ce sont des programmes ou ensemble de programmes cohérents, compatibles entre eux pour assurer un bon fonctionnement et résoudre un problème bien déterminé. Parmi ces programmes nous citons les systèmes d'exploitation permettant comme leur nom l'indique, l'exploitation de l'ordinateur, ses divers périphériques ainsi que les différents programmes à faire exécuter. Certains modules des systèmes d'exploitation permettent la communication entre deux ou plusieurs ordinateurs, ils assurent la mise en marche d'un réseau informatique comme nous l'avons vu au chapitre précédent. Dans ces mêmes systèmes, vous avez découvert aussi plusieurs autres applications permettant de répondre à divers besoins tels que le traitement de texte, le dessin, le son, la téléphonie et même les jeux. Dans notre quotidien, l'informatisation évolue d'une façon vertigineuse, en effet, tous les domaines de la vie courante ou presque, sont devenus informatisés. En partant du récepteur numérique de votre téléviseur et de votre téléphone portable jusqu'aux différentes factures qui vous parviennent, toutes ces opérations deviennent gérées par des programmes informatiques que les ordinateurs exécutent avec les plus grandes précision et rapidité.

Activité 1

Citez au moins sept exemples d'applications de l'informatique dans le quotidien que vous vivez. Faites une description de cette utilisation.

Pour vous aider, nous vous donnons quelques exemples :

- Dans mon téléphone portable, il y a des programmes de jeux. De temps à autres, quand je m'ennuie, je lance mon jeu préféré. Mon portable devient un petit ordinateur !
- Dans la pharmacie du quartier, les agents saisissent les noms des médicaments ; l'ordinateur se charge du reste et à l'aide de l'imprimante, il inscrit le total à payer directement sur l'ordonnance.
- Dans mon lycée, il y a un agent de l'administration qui gère avec un ordinateur les notes des élèves et fait sortir leurs bulletins pour chaque trimestre.

Toutes les pratiques informatisées et les exemples que vous avez évoqués ou constatés sont gérés par des programmes que l'ordinateur exécute pour résoudre un problème préalablement posé. Nous allons voir dans la suite et, étape par étape, la démarche à suivre pour résoudre un problème posé. Nous ne prétendons pas résoudre des problèmes similaires à ce que vous veniez d'évoquer mais nous allons apprendre les bases d'une telle démarche.

Notons à la fin que pour trouver une solution informatique à un problème donné, il faudra passer par une série d'étapes. Nous allons les évoquer une à une.

II. Première étape : Position du problème

Le problème est souvent posé par un demandeur de solution informatique. C'est l'exemple du pharmacien qui veut gérer toutes les opérations de sa pharmacie, ou une société de transport qui veut superviser les réservations de ses clients ou encore votre professeur qui compte informatiser son carnet de notes. Souvent les énoncés formulés par le poseur du problème manquent de précision, il vous est donc impératif de reformuler en détail avec l'intéressé, le problème posé.

Exemples :

- Ecrire un programme qui calcule et affiche les moyennes des élèves d'une classe. Comme vous le constatez, cet énoncé manque de précision. En effet, il manque le nombre de notes, leurs coefficients respectifs, etc.
- On compte écrire un programme qui résout des équations du premier degré. On pourra se poser la question à propos du domaine des coefficients ainsi que celui de la résolution.
- Un pharmacien se propose d'informatiser les différentes opérations de sa pharmacie.

Bien entendu, ce cas est plus complexe et suscite davantage de questions. Quelles sont les actions concernées par l'informatisation : la facturation, les commandes, la paye du personnel, etc. Cet exemple gère aussi des bases de données, donc il est important de savoir comment les concevoir et les construire.

Activité 2

Donner une formulation détaillée pour chacun des problèmes suivants :

- Ecrire un programme qui résout une inéquation.
- Ecrire un programme qui calcule le bénéfice fait suite à la vente d'un produit.
- Ecrire un programme qui calcule le quotient de deux nombres.
- Ecrire un programme permettant d'informatiser l'édition d'une facture comme celle de la STEG ou de la SONEDE.

Bien entendu, si l'énoncé du problème est clair et précis, cette étape est gagnée.

Notons qu'à l'issue de cette étape, le problème devient suffisamment bien posé et le résultat voulu devrait être décrit avec la plus grande précision. Nous passons maintenant à l'étape de résolution la plus importante

III. Deuxième étape : Spécification et analyse du problème

Cette étape consiste d'abord à spécifier la tâche du programme que nous comptons écrire. Ensuite, on procède à l'analyse du problème pour lui trouver une solution. Il existe plusieurs approches pour analyser un problème.

Nous en citons les plus connues à savoir l'approche ascendante et l'approche descendante. La première approche consiste à partir des données et commencer à développer les traitements nécessaires jusqu'à arriver au résultat escompté.

La seconde approche part du résultat du problème posé et au fur et à mesure, on définit chaque nouvel élément qui apparaît dans l'analyse. Pour trancher, il n'y a pas une approche meilleure que l'autre, chacune a ses avantages et ses inconvénients. D'ailleurs, on adopte souvent les deux approches selon les passages de l'analyse.

Exemple de résolution avec les deux approches.

Problème :

Ecrire un programme qui calcule et affiche la surface d'un rectangle dont la longueur et la largeur sont données.

Approche ascendante

Les données sont : Longueur et Largeur

Le traitement :

Calcul de la surface :

Surface = Longueur x Largeur

La sortie

Afficher Surface

Fin du programme

Approche descendante

Résultat : Afficher (Surface)

Surface = Longueur x Largeur

Longueur = DONNEE

Largeur = DONNEE

Fin du programme

Nous remarquons que l'approche descendante dicte un certain comportement systématique tout le long du développement de la solution. En effet, tout nouvel élément apparu dans l'analyse est placé à son tour à gauche pour être défini. De cette manière, on ne risque pas d'oubli d'un élément quelconque nécessaire à la résolution du problème en question. Notons que cette méthode impose à ce qu'on ait en tête une solution claire et précise au problème posé. Pour systématiser cette opération et en vue d'une organisation plus claire permettant un développement rapide et organisé de cette étape, nous allons utiliser la grille suivante :

| NOM = nom_du_programme | | |
|-------------------------------|--|----------------------------------|
| S (Séquences) | L.D.E. (Liste des Définitions Explicites) | O.U. (Objets Utilisés) |
| | <p>Résultat =</p> <p style="text-align: center;">↑</p> <p style="text-align: center;">Les éléments de l'analyse</p> <p style="text-align: center;">↓</p> <p>Fin nom_du_programme</p> | |

La première ligne de la grille comporte le nom que vous auriez choisi pour votre programme. La colonne centrale comporte explicitement toutes les définitions des objets constituant la solution du problème en suivant une analyse descendante comme indiquée dans l'exemple ci-dessus.

Au fur et mesure que nous avançons dans l'analyse du problème, nous inscrivons les nouveaux objets dans la 3^{ème} colonne pour ne pas oublier de les définir.

La première colonne à gauche intitulée "Séquences" sert à ordonner les actions évoquées dans l'analyse par ordre chronologique d'exécution. Cette opération est très importante pour l'écriture de l'algorithme.

Reprenons l'exemple précédent du calcul de la surface d'un rectangle. L'analyse qui a été faite précédemment pourra être placée dans la grille comme suit :

| NOM = Surface_ Rectangle | | |
|---------------------------------|---|----------------------------------|
| S (Séquences) | L.D.E. (Liste des Définitions Explicites) | O.U. (Objets Utilisés) |
| 4 | Résultat = Afficher (Surface) | |
| 3 | Surface ← Longueur x Largeur | Surface |
| 1 | Longueur = Donnée | Longueur |
| 2 | Largeur = Donnée | Largeur |
| 5 | Fin Surface_ Rectangle | |

La colonne des séquences comporte un ordre logique d'exécution des différentes actions qui ont constitué ce programme.

Remarque : un problème peut avoir plusieurs solutions.

IV. Troisième étape : Écriture de l'algorithme

L'analyse d'un problème telle qu'elle est faite avec la grille devra permettre à la fin de dresser dans l'ordre croissant d'exécution les différentes instructions ou actions qui ont formé la solution du problème. L'écriture des instructions dans cet ordre croissant ou chronologique d'exécution donne un algorithme solution au problème posé. Toutefois, il est nécessaire de faire des conventions d'écriture et de se mettre d'accord sur un même langage pour que cette écriture soit comprise de la même façon par différentes personnes. Un certain vocabulaire muni d'une certaine syntaxe formeront ce qu'on appelle **le langage algorithmique**.

Definition

Un algorithme est une suite ordonnée et finie d'actions ou d'instructions dont l'exécution servira à résoudre un problème donné.

N.B. Le mot algorithme provient du nom du mathématicien arabe Muhammad Ibn Musa AL KHAWARIZMI (740 - 810) né à Khawarezm, un village perse. On lui doit aussi le mot algèbre pour l'un de ses ouvrages « Al Mukhtassar Fi Al-Jaber Wal Mukabala ».

Écrivons l'algorithme déduit de l'analyse du problème relatif au calcul de la surface d'un rectangle :

- 0) Début SurfaceRectangle
- 1) Lire(Longueur)
- 2) Lire(Largeur)
- 3) Surface ← Longueur * Largeur
- 4) Ecrire(Surface)
- 5) Fin SurfaceRectangle

Nous attirons votre attention sur les conventions suivantes :

- l'utilisation du verbe « **Lire** » pour la saisie d'une donnée et du verbe « **Ecrire** » pour l'affichage ou la sortie du résultat.
- l'utilisation du signe « ← » pour traduire l'affutation d'une variable par une valeur.

Ce vocabulaire et sa syntaxe d'utilisation seront développés en détail dans les prochains chapitres.

Activité 3

Problème : On se propose d'écrire un programme qui permet de calculer et d'afficher à l'écran, la moyenne annuelle Moy d'un élève. On donne la formule de calcul suivante :

$$\text{Moy} = (\text{Note_trim1} + 2 * \text{Note_trim2} + 2 * \text{Note_trim3})/5$$

Analyse du problème.

| NOM : Moyenne_annuelle | | |
|------------------------|--|------------|
| S | L.D.E. | O.U. |
| 5 | Résultat = Écrire(Moy) | |
| 4 | $Moy \leftarrow (Note_trim1 + 2 * Note_trim2 + 2 * Note_trim3) / 5$ | Moy |
| 1 | Note_trim1 = Donnée | Note_trim1 |
| 2 | Note_trim2 = Donnée | Note_trim2 |
| 3 | Note_trim3 = Donnée | Note_trim3 |
| 6 | Fin Moyenne_annuelle | |

Nous déduisons l'algorithme correspondant :

- 0) Début Moyenne_annuelle
- 1) Lire(Note_trim1)
- 2) Lire(Note_trim2)
- 3) Lire(Note_trim3)
- 4) $Moy \leftarrow (Note_trim1 + 2 * Note_trim2 + 2 * Note_trim3) / 5$
- 5) Ecrire(Moy)
- 6) Fin Moyenne_annuelle

Activité 4 (non corrigée)

Problème : Ecrire un algorithme qui permet de calculer et d'afficher pour une voiture, la consommation de carburant en litres par 100 km sachant qu'elle a consommé x litres d'essence pour parcourir d km. d et x sont des données.

V. Quatrième étape : Ecriture du programme

Une fois l'algorithme solution d'un problème est établi ou écrit, l'étape suivante sera de penser à la machine et comment le traduire pour qu'il soit compris et exécuté par cette dernière. Les langages avec lesquels nous allons traduire cet algorithme sont appelés langages de programmation. Ils sont dits évolués relativement au langage binaire et au langage assembleur qui sont plus proches de la machine. Il existe plusieurs langages évolués. Ces derniers ont été souvent développés selon des besoins spécifiques. Avec l'évolution de l'informatique et en vue d'étendre l'activité de programmation au plus large public, plusieurs langages ont vu le jour.

Caractéristiques d'un langage de programmation

Un langage de programmation sert à traduire un algorithme en un programme source compréhensible directement ou indirectement par l'ordinateur pour que ce dernier puisse l'exécuter.

Il est composé comme tout langage, d'un vocabulaire obéissant à une sémantique et à une syntaxe. La sémantique donne un sens à chaque mot de ce langage par contre la syntaxe détermine les règles d'utilisation.

Ces langages sont très stricts du côté vocabulaire et syntaxe. Ils sont souvent munis d'un analyseur permettant de détecter les erreurs faites sur le vocabulaire et la syntaxe.

Ces langages sont souvent munis d'un éditeur de texte permettant au programmeur de saisir son programme. Le fichier obtenu est appelé programme source. Ce dernier est compréhensible par un lecteur qui connaît bien le langage.

L'exécution de ce programme dépend du langage utilisé. Si le langage est **interprété**, le programme source sera exécuté instruction par instruction. Par contre, si le langage est **compilé**, il faudra faire la compilation du programme source et corriger les éventuelles erreurs pour produire un autre programme dit programme exécutable. C'est ce dernier, écrit en langage machine par le langage lui-même, qui, sera exécuté. Les éventuelles erreurs qui se produiront seront dites erreurs d'exécution.

Un langage **compilé** donne des programmes exécutables dont l'exécution est très rapide relativement à l'exécution d'un programme interprété.

En voici un tableau des langages de programmation les plus connus.

| Nom | Origine | Année | Caractéristiques |
|--|---------------------------------------|-----------|--|
| FORTRAN : (Formula translation) | J. BACKUS, IBM | 1954-1957 | Compilé, Applications scientifiques (mathématiques). |
| LISP : LISt Processing | J. BACKUS, IBM | 1956-1961 | Langage fonctionnel (interprété ou compilé). Applications pour intelligence artificielle. |
| Algol : (Algorithmic Language) | J. Backus, P. Naur | 1958-1960 | Compilé, Langage à structure de blocs. |
| APL : (A Programming Langage) | IBM | 1960 | Interprété, Applications mathématiques |
| COBOL : (Commun Business Oriented Language) | US DOD GPO et CODASYL Committee | 1960 | Compilé, Applications de gestion et de commerce. |

| Nom | Origine | Année | Caractéristiques |
|--|----------------------------------|-----------|--|
| BASIC : (The Beginners All-purpose Symbolic Instruction Code) | J. KEMENY, T. KURTZ (à Darmouth) | 1963-1964 | Interprété, Syntaxe simple, larges applications. Très répandu avec les micros. |
| PASCAL : (hommage au savant Blaise PASCAL) | Niklaus WIRTH | 1968-1971 | Compilé, Applications générales. |
| C | Richie et Thomson | 1972-1973 | Compilé, Langage à structuration de blocs |
| PROLOG : (PROgramming in LOGic) | D.H. WARREN et R.KOWALSKI | 1979 | Souvent interprété, Applications mathématiques et logiques. |
| C++ | AT&T Bell Labs | 1982-1985 | Compilé, Langage orienté objets. |
| PERL : Practical Extraction and Reporting Language | Larry WALL | 1986 | Interprétation de script. Applications générales. |
| JAVA | SUN Microsystems | 1994-1995 | Langage orienté objet. Applications diverses. |

Nous allons utiliser le Pascal. C'est un langage compilé et s'adapte bien à l'approche d'analyse adoptée.

Rappelons qu'un programme manipule des données que ce soit en entrée, au cours du traitement ou en sortie. Ces données nécessitent un emplacement en mémoire. C'est pour cette raison que la plupart des langages exigent au programmeur de préciser le type des données à utiliser pour que le programme prépare les emplacements adéquats en mémoire. Nous précéderons donc chaque programme par un tableau indiquant le nom, le type ou la nature ainsi que le rôle des objets utilisés dans le programme.

Le tableau est de cette forme :

| Nom | Type/Nature | Rôle |
|-----|-------------|------|
| | | |

Activité 5

Reprenez l'algorithme intitulé SurfaceRectangle établi précédemment et en déduisez le tableau de type des objets utilisés.

| Nom | Type/Nature | Rôle |
|----------|-------------|---|
| Longueur | Réel | Pour saisir la longueur du rectangle |
| Largeur | Réel | Pour saisir la largeur du rectangle |
| Surface | Réel | Contenir la valeur calculée de la surface |

Nous vous proposons ci-dessous une traduction en Pascal de l'algorithme SurfaceRectangle. Nous vous invitons à le saisir et l'exécuter avec différentes valeurs pour la longueur et la largeur.

```
PROGRAM SurfaceRectangle ;
  USES WINCRT;
  VAR  Longueur, Largeur : REAL ;
        Surface :REAL ;

  BEGIN
    READLN(Longueur);
    READLN(Largeur);
    Surface := Longueur * Largeur;
    WRITELN(Surface) ;
  END.
```

VI. Cinquième étape : Exécution et tests du programme

Bien entendu, un programme est écrit pour être exécuté et résoudre un problème donné. Souvent et à cause d'une erreur quelconque, le programme s'exécute normalement mais ne répond pas favorablement à certains cas. Il est donc impératif de faire subir au programme obtenu, une série de tests bien choisis pour s'assurer que ce dernier fait bien ce qu'on lui demande de faire dans tous les cas possibles.

Activité 6

Ci-dessous le programme en Pascal qui saisit deux entiers m et n, calculez et affichez leur somme et leur produit. Saisissez ce programme dans l'éditeur du langage Pascal, corrigez les éventuelles erreurs de frappe et exécutez ce programme pour plusieurs valeurs de m et n. Déduire.

```
PROGRAM SommeProduit ;
  USES WINCRT;
  VAR  m, n : INTEGER ;
        somme, produit : INTEGER ;
  BEGIN
```

```
BEGIN
  WRITE ('m = '); READLN (m);
  WRITE ('n = '); READLN (n);
  Somme := m + n;
  Produit := m * n;
  WRITELN ('Somme = ',somme) ;
  WRITELN ('Produit = ',produit) ;
END.
```

Retenons

- Pour résoudre un problème, il faudra expliciter en détail son énoncé et bien définir le résultat demandé.
- L'approche descendante part du résultat pour arriver aux actions élémentaires et aux données nécessaires.
- L'écriture dans l'ordre croissant d'exécution des différentes actions de l'analyse donne l'algorithme solution du problème posé.
- Un programme compilé est plus rapide qu'un programme interprété.
- Pour s'assurer du bon fonctionnement d'un programme, on doit lui faire subir beaucoup de tests variés qui englobent tous les différents cas possibles.

Exercices

Exercice 1

Formulez au moins deux problèmes à résoudre avec l'ordinateur dans chacun des domaines suivants :

- Les mathématiques
- La physique
- Les langues

Exercice 2

Donnez les différentes étapes à suivre pour résoudre un problème, depuis la position du problème jusqu'aux tests du programme.

Exercice 3

Faites une analyse et en déduire un algorithme qui conjugue un verbe du 1er groupe au passé composé.

(Définir le résultat puis chaque constituant de ce résultat).

Exercice 4

Ecrivez un algorithme qui saisit un verbe du 1er groupe et le conjugue à l'imparfait.

(Définissez le résultat puis chaque constituant de ce résultat).

Exercice 5

On se propose de résoudre une équation du premier degré dans l'ensemble des réels de la forme $a.x + b = 0$. Effectuez une analyse descendante de ce problème et en déduisez l'algorithme correspondant.

Exercice 6

On veut déterminer le plus grand de deux réels donnés a et b . Trouvez une expression pour traduire ce résultat.

Trouvez une formule mathématique donnant directement ce résultat.

Exercice 7

Mêmes questions que l'exercice 6 pour la détermination du plus petit élément de deux réels donnés a et b .

Exercice 8

Donnez au moins deux caractéristiques d'un langage de programmation.

Exercice 9

Citez deux langages interprétés et deux autres compilés.

Exercice 10

Quelles sont les différences entre un langage interprété et un langage compilé.

Chapitre 6

Les structures de données

Objectifs :

- Identifier et manipuler les différents types de données

Plan du chapitre :

Leçon 1 :
Les constantes et les variables

Leçon 2 :
Les types de données et
les expressions

Leçon 3 :
Les tableaux

Leçon 1

Les constantes et les variables

Objectifs spécifiques :

Identifier et manipuler les constantes et les variables.

Plan de la leçon

- I- Les constantes et les variables
- II- Les variables

Retenons

Exercices

Leçon 1

Les constantes et les variables

Un algorithme est une suite finie d'actions qui manipulent des entités que nous appellerons dans la suite des objets. Dans ce chapitre, nous allons nous intéresser à ces objets qui peuvent être des constantes ou des variables de type simple ou de type structuré.

I- Les constantes

Définition : On appelle constante un objet ayant une valeur fixe tout le long d'un algorithme.

Une constante est caractérisée par :

- son nom (un identificateur unique).
- sa valeur.

N.B. : Dans toute la suite, l'identificateur d'une constante ou tout autre objet est un nom commençant par une lettre et dépourvu de caractères spéciaux comme les caractères de ponctuation ou les opérateurs.

Exemples de constantes :

$P = 3.1415$

Durée = 24

Fréquence = 50

Exemples d'utilisation d'une constante :

Dans le calcul numérique de la surface d'un cercle, on pourra utiliser la constante P et la formule :

$$\text{Surface} = P \times \text{Rayon} \times \text{Rayon}$$

Au niveau de l'analyse, la déclaration de la constante P se fera comme suit :

Tableau de déclaration des objets

| Objet | Type/Nature | Rôle |
|-------|--------------------|------------------|
| ... | | ... |
| P | Constante = 3.1415 | Constante connue |
| ... | | ... |

Remarque :

La valeur que prendra la constante, dans notre exemple $P = 3.1415$ nous renseigne sur le type de cette constante. Dans le cas présent, la constante P est réelle donc les opérateurs valides sur le type réel peuvent lui être appliqués.

Dans le langage Pascal, la déclaration d'une constante se fait comme suit :

```
CONST <nom_constante> = valeur_constante ;
```

Exemple

```
CONST P = 3.14;
```

Activité 1

Identifier quelques constantes utilisées dans votre cours de physique dans l'intention d'écrire des algorithmes les utilisant.

Réponse : La constante de gravité g , connue aussi sous le nom de constante de Newton, utilisée dans le calcul du poids, vaut 9.81 (N/Kg). Cette constante sera déclarée dans le langage PASCAL de la manière suivante :

```
CONST g = 9.81;
```

Nous supposons que les calculs sont à deux chiffres après la virgule.

II. Les variables

Définition : On appelle variable un objet pouvant prendre différentes valeurs tout le long d'un algorithme.

Une variable est caractérisée par :

- Son nom (un identificateur unique)
- Son type
- Son contenu

Remarques :

- 1- La notion de variable est une notion fondamentale en algorithmique car elle constitue le moyen de stocker les données.
- 2- L'opération permettant de changer le contenu d'une variable est appelée affectation. On la désignera dans ce manuel par le symbole algorithmique \leftarrow . En langage Pascal, elle est représentée par $:=$.
- 3- L'opération d'affectation d'une variable ne concerne que cette dernière, autrement dit s'il y a d'autres variables, leurs contenus resteront inchangés.

Activité 2

Soit la séquence d'affectations suivante :

1. m ← 10
2. n ← 20
3. p ← m
4. m ← n
5. n ← p

1) Donnez le résultat d'exécution de cette séquence.

Réponse

Nous pouvons suivre l'évolution des valeurs de nos variables m, n et p en les regroupant dans un tableau constituant la trace de la séquence :

| Trace de la séquence | | | |
|----------------------|----|----|----|
| N° de l'instruction | m | n | p |
| 1 | 10 | - | - |
| 2 | 10 | 20 | - |
| 3 | 10 | 20 | 10 |
| 4 | 20 | 20 | 10 |
| 5 | 20 | 10 | 10 |

A l'instruction N°1, seule la variable m prend la valeur 10.

A l'instruction N°2, la variable n prend la valeur 20 mais la variable m est restée inchangée.

A l'instruction N°3, p a pris la valeur de m c'est à dire 10, m et n sont restées inchangées.

A l'instruction N°4, m a changé et a pris la valeur de n c'est à dire 20 et n et p sont restées inchangées.

A l'instruction N°5, n a pris la valeur de p c'est à dire 10, m et p sont restées inchangées.

2) Quelles sont les valeurs finales de m et de n ?

Réponse

Les valeurs finales de m et de n sont respectivement 20 et 10.

3) Quel est le rôle de cette séquence ?

Réponse

Cette séquence a permis la permutation des contenus des deux variables m et n.

4) Quelle est l'utilité de la variable p ?

Réponse

La variable p a assuré la sauvegarde du contenu de m avant de lui affecter n à l'instruction N°4. Elle est appelée variable intermédiaire, temporaire ou auxiliaire.

En effet, si on affecte à m le contenu de n sans le sauver dans p, on le perdra.

Exemples d'utilisation d'une variable :

Dans le calcul numérique à 3 chiffres après la virgule de la surface d'un cercle, on pourra utiliser la constante et les variables Surface et Rayon.

$$\text{Surface} = P \times \text{Rayon} \times \text{Rayon}$$

Au niveau de l'analyse, la déclaration de la constante P et des variables Surface et Rayon se fera comme suit :

Tableau de déclaration des objets

| Objet | Type / Nature | Rôle |
|---------|-------------------|------------------------------|
| ... | ... | ... |
| P | Constante = 3.145 | Constante connue |
| Surface | Réel | Surface calculée d'un cercle |
| Rayon | Réel | Rayon du cercle |
| ... | ... | ... |

Dans le langage Pascal, la déclaration d'une variable se fait comme suit :

VAR <nom_variable> : type_variable ;

Exemple

```
VAR      n : INTEGER;
         x : REAL ;
         Surface : REAL ;
         Rayon : REAL ;
```

Retenons

- Les structures de données vues dans cette leçon sont :
 - les constantes (de valeurs inchangées tout le long de l'algorithme).
 - les variables (pouvant changer de contenu tout le long de l'algorithme).
- Les identificateurs de ces structures sont des chaînes de caractères commençant par une lettre.

Exercices

Exercice 1

Évaluez le contenu des variables a, b et c suite à l'exécution de chacune des séquences suivantes :

| | | |
|---------------|--------------|--------------|
| 1) a ← 10 | 1) a ← 2 | 1) a ← 7 |
| 2) b ← a + 10 | 2) b ← a * a | 2) b ← 3 |
| 3) c ← a + b | 3) c ← a * a | 3) c ← a + b |
| 4) a ← b - c | 4) a ← a * a | 4) a ← a - b |
| | | 5) a ← a - b |

Exercice 2

Évaluez le contenu des variables p, q et r suite à l'exécution de chacune des séquences suivantes :

| | |
|----------|----------|
| 1) p ← 5 | 1) p ← 5 |
| 2) q ← 8 | 2) q ← 8 |
| 3) p ← q | 3) p ← q |
| 4) q ← p | 4) q ← p |
| | 5) q ← r |

Exercice 3

On désire calculer la surface d'un triangle isocèle. Établissez un tableau de déclaration des objets nécessaires.

Leçon 2

Les types de données et les expressions

Objectifs spécifiques :

- Manipuler des types standards et définir des types utilisateurs.
- Utiliser des opérateurs et des fonctions prédéfinis et les évaluer.

Plan de la leçon

- I- Les types de données
 1. Les types numériques
 2. Le type booléen
 3. Le type caractère
 4. Autres types
- II- Les expressions
 1. Les opérandes
 2. Les opérateurs
 3. Evaluation d'une expression

Retenons

Exercices

Leçon 2

Les types de données et les expressions

I. Les types de données

Les exemples précédents ont montré que chaque déclaration de variable définit son type.

Le type permet:

- de déterminer l'ensemble des valeurs qui peuvent être affectées à la variable en question.
- de fixer implicitement l'ensemble des opérateurs valides sur ces variables.
- De choisir la représentation ou l'occupation en octets dans la mémoire RAM.

Un type est désigné par un identificateur (ou un nom). Les types les plus connus sont :

- Le type Entier, désignant les valeurs des nombres entiers relatifs.
- Le type Réel, désignant les valeurs des nombres réels.
- Le type Caractère, désignant les "valeurs" des caractères.
- Le type Booléen, désignant les valeurs logiques.
- Le type Chaîne de caractères, désignant les "valeurs" des chaînes de caractères.

I.1. Les types numériques

I.1.1. Le type entier

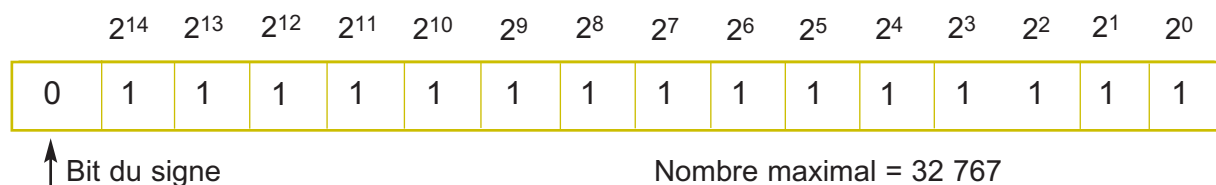
a) Domaine des valeurs du type entier

Les valeurs du type entier forment un sous-ensemble de l'ensemble des nombres entiers (un sous-ensemble de \mathbb{Z}). Il s'agit de la représentation en machine des entiers relatifs.

Un entier est représenté avec un nombre fixe d'octets ou de bits. C'est pour cette raison que l'ensemble des entiers dans un ordinateur est un intervalle borné de \mathbb{Z} donc il a un minimum et un maximum.

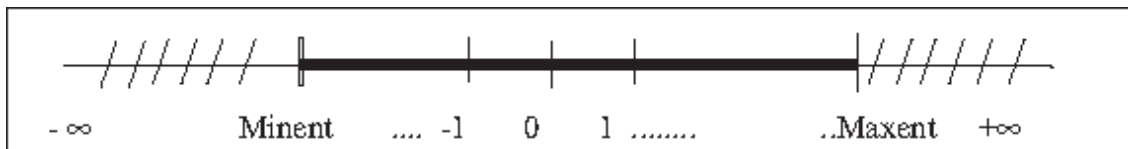
Exemple :

Supposons que les entiers dans une machine sont représentés par deux octets ou par 16 bits avec le bit le plus à gauche pour représenter le signe, on obtient la configuration suivante :



bit du signe = 0 si nombre ≥ 0
 = 1 si nombre < 0

Ce nombre maximal est décrit par la constante prédéfinie Maxent qui définit la valeur absolue du plus grand entier représentable. Le sous-ensemble désigné par entier est donc le suivant :



Exemples de nombres entiers :

-5 1 0 7 -42 673 -3220

b) Les opérateurs arithmétiques sur les entiers

On ajoutera aux opérateurs arithmétiques usuels (+, - et *), deux autres opérateurs **DIV** et **MOD** définis ci dessous : (On convient de noter l'opération de multiplication par le signe *)

DIV : Donne le quotient dans la division entière

Exemple : 17 DIV 3 vaut 5

MOD : Donne le reste de la division entière

Exemple : 17 MOD 3 vaut 2

| | | |
|----------|----------|------------------------------|
| a | b | q vaut a DIV b |
| r | q | r vaut a MOD b |

Remarques :

- Quand on manipule une variable de type entier, il faut faire attention au risque de débordement c'est-à-dire avoir des entiers sortant de l'intervalle [Minent,Maxent].
- Quand il y a débordement au-delà des valeurs Minent et Maxent, les calculs deviennent erronés ou provoquent des erreurs d'exécution.

Dans le langage Pascal, la déclaration d'une variable entière se fait comme suit :

VAR <nom_variable> : INTEGER ;

Exemple : VAR m, i, j, k : INTEGER;

Activité 1**1- Est-ce qu'on pourra utiliser le type entier pour représenter les quantités suivantes**

- a- Note obtenue lors d'un devoir surveillé
- b- Nombre d'élèves d'une classe
- c- Nombre de matières étudiées pendant un trimestre
- d- Moyenne générale d'un élève

2- Évaluez les expressions arithmétiques suivantes :

- a- $((46 \text{ DIV } 3) \text{ MOD } 4)+1$
- b- $(25 \text{ MOD } 7) \text{ DIV } (2 * 3)$

Réponse

- 1-a- Une note est du type réel car on peut avoir par exemple une note de 16.5.
- b- Le nombre d'élèves est un entier.
- c- Le nombre de matières étudiées est un entier.
- d- La moyenne générale est un nombre réel.

- 2-a-Le résultat de cette expression est : 4
- b-Le résultat de cette expression est : 0

I.1.2. Le type réel**a) Domaine des valeurs du type réel**

Les valeurs du type réel forment un sous-ensemble de l'ensemble des nombres réels IR. La définition de ce sous-ensemble est liée à la représentation en binaire des réels dans la machine.

Exemples de nombres réels

| | | | | | | |
|---|-------|------|--------------|------|-------|------------------|
| 0 | -7.25 | 3.14 | 30.10^{-5} | 0.38 | -26.0 | $3.1 \cdot 10^6$ |
|---|-------|------|--------------|------|-------|------------------|

$3.1 \cdot 10^6$ c'est à dire 3 100 000 s'écrira aussi $3.1E+6$. La lettre E se lit : «dix puissance»

b) Les opérateurs arithmétiques sur les réels

Ces opérateurs sont :

- + pour l'addition,
- pour la soustraction,
- * pour la multiplication,
- / pour la division réelle.

Dans le langage Pascal, la déclaration d'une variable réelle se fait comme suit :

```
VAR <nom_variable> : REAL ;
```

Exemple : VAR x, y : REAL;

I.1.3. Les fonctions arithmétiques standards

Le type du paramètre (noté x ou n dans les exemples suivants) des fonctions arithmétiques est soit entier, soit réel ; le type du résultat de la fonction est toujours réel sauf pour les fonctions Abs et Carré qui délivrent un résultat du même type que le paramètre et les fonctions Tronc et Arrondi qui délivrent toujours un entier.

| Nom algorithmique | Code en Pascal | Rôle | Exemples |
|------------------------|------------------|---|--|
| Tronc (x) | TRUNC(x) | supprime la partie décimale pour ne laisser que la composante entière de x. | Tronc (-5.125) vaut -5 Tronc (3.14) vaut 3 Tronc (123.5) vaut 123 |
| Arrondi (x) | ROUND(x) | donne un entier qui est le plus proche du réel x | Arrondi (7.499) vaut 7 Arrondi (7.50) vaut 8 Arrondi (7.99) vaut 8 |
| Abs (x) | ABS(x) | donne la valeur absolue de x. | Abs (-3) vaut 3 |
| Carré (x) | SQR(x) | donne le carré de x. | Carré (3) vaut 9 |
| RacineCarré (x) | SQRT(x) | donne la racine carrée de x si x n'est pas négatif et provoque une erreur, sinon. | RacineCarré (2) vaut 1.414 ... |
| Sin (x) | SIN(x) | donne le sinus de x (x en radians). | Sin ($\pi/2$) vaut environ 1 |
| Cos (x) | COS(x) | donne le cosinus de x (x en radians). | Cos ($\pi/2$) vaut environ 0 |
| ENT(x) | INT(x) | donne la partie entière d'un réel | INT(2.5) vaut 2 INT(-3.5) vaut -4 |
| Aléa | RANDOM | donne un réel au hasard compris entre 0 et 1 exclus | Aléa vaut 0.45 par exemple |
| Aléa(n) | RANDOM(n) | donne un entier au hasard entre 0 et n-1 | Aléa(7) vaut 6 par exemple |

Activité 2

Traduire en langage Pascal les affectations suivantes

```
Delta ← Carré (b) -4*a*c
x1 ← (- b - RacineCarré (Delta) ) / (2 * a)
x2 ← (- b + RacineCarré (Delta) ) / (2 * a)
```

Réponse

```
Delta := SQR (b) -4*a*c ;
x1 := (- b - SQRT (Delta) ) / (2 * a) ;
x2 := (- b + SQRT (Delta) ) / (2 * a) ;
```

I.2. Le type booléen

Dans certains cas, il sera nécessaire d'évaluer des propositions pour déduire si elles sont vraies ou fausses. C'est par exemple le cas lors de la comparaison de deux entiers. Les variables censées prendre une valeur parmi VRAI ou FAUX sont dites variables booléennes.

I.2.1. Domaine des valeurs du type booléen

Le type contient les deux valeurs logiques VRAI et FAUX.

I.2.2. Opérateurs logiques sur les booléens

Ces opérateurs sont les suivants :

| | | | |
|-------------|---|-------------|-------------------------|
| NON | : | Négation | (opérateur unaire) |
| ET | : | Conjonction | } (opérateurs binaires) |
| OU | : | Disjonction | |
| OUex | : | Ou exclusif | |

Ci-dessous la table de vérité de ces opérateurs où P et Q sont des constantes, des variables ou des propositions logiques :

| P | Q | NON(P) | P ET Q | P OU Q | P OU ex Q |
|------|------|--------|--------|--------|-----------|
| FAUX | FAUX | VRAI | FAUX | FAUX | FAUX |
| FAUX | VRAI | VRAI | FAUX | VRAI | VRAI |
| VRAI | VRAI | FAUX | VRAI | VRAI | FAUX |
| VRAI | FAUX | FAUX | FAUX | VRAI | VRAI |

Exemple : La proposition $(7 > 5) \text{ ET } (2 < 3)$ a la valeur VRAI
 La proposition $(7 > 5) \text{ ET } (5 < 3)$ a la valeur FAUX

Remarques :

Il existe un ordre de priorité entre les opérateurs logiques :

- La négation **NON** est prioritaire par rapport à la conjonction ET.
- La conjonction **ET** est prioritaire par rapport à la disjonction OU.
- La disjonction **OU** a la même priorité que l'opérateur OUex.

Si deux opérateurs ont la même priorité, le calcul de l'expression logique se fera de gauche à droite.

Dans tous les cas, les opérations mises entre parenthèses sont prioritaires. Les parenthèses les plus internes sont les plus prioritaires.

Exemple : Soient p, q et r trois variables booléennes

- 1- Dans l'expression $p \text{ OU } q \text{ ET } r$, on évalue d'abord la conjonction $q \text{ ET } r$ puis la disjonction $p \text{ OU } (q \text{ ET } r)$ comme s'il y avait des parenthèses.
- 2- Si on veut d'abord évaluer la disjonction $p \text{ OU } q$, il suffit d'ajouter des parenthèses $(p \text{ OU } q) \text{ ET } r$

Dans le langage Pascal, la déclaration d'une variable booléenne se fait comme suit :
 VAR <nom_variable> : BOOLEAN ;

Exemple : VAR p, q, existe, drapeau : BOOLEAN ;

Activité 3

Evaluer les expressions logiques suivantes

1. $(x \leq 7) \text{ ET } (x \geq 0)$ avec $x = 6$
2. $(x \leq 7) \text{ ET } (x \geq 0)$ avec $x = -1$
3. $\text{NON } (x \leq 7) \text{ OU } (x \geq 0)$ avec $x = 10$

Réponse

1. VRAI
2. FAUX
3. VRAI

I.3. Le type caractère

Un caractère est représenté par le caractère lui-même placé entre apostrophes.

Exemples : "A", "a", "+", "1", "(", etc.

I.3.1. Valeurs du type caractère

Tous les caractères alphanumériques imprimables de l'alphabet latin sont admis. Il s'agira en l'occurrence des lettres (minuscule et majuscule), des chiffres, des symboles spéciaux (de ponctuation, signes et autres).

Outre ces caractères imprimables, on admettra tous les caractères spéciaux non imprimables ayant des significations particulières tels que le retour chariot, l'Echappe (Escape), le Bip sonore, etc.

Tous ces caractères sont ordonnés selon leur code ASCII (voir Annexe à la fin du livre).

Les chiffres, les lettres en majuscule, les lettres en minuscule ont des codes ordonnés et contigus ; ainsi on peut les comparer.

"0" < "1" ... <"9" < ... "A" < "B" ... < "Z" ... "a" < "b" <"z" ...

I.3.2. Opérateurs sur les caractères

Comme les valeurs de type caractère sont ordonnées, les opérateurs relationnels = , ≠ , < , ≤ , > et ≥ y sont définis.

Exemples : "C" < "D" est une proposition **VRAIE**.

"z" > "a" est une proposition **VRAIE**.

Dans le langage Pascal, la déclaration d'une variable de type caractère se fait comme suit

VAR <nom_variable> : **CHAR** ;

- Une variable de type caractère contient impérativement un caractère et un seul.
- L'espace est un caractère "blanc".

Exemples : VAR c1, c2, car : CHAR ;

Activité 4

Évaluez les expressions suivantes :

- 1- "a" > "A"
- 2- "1" < "B"

Réponse

- 1- Vraie
- 2- Vraie

I.3.3 Les fonctions prédéfinies

Nous citons quelques fonctions prédéfinies à grand usage telles que : ORD, CHR, SUCC et PRED.

Supposons que *c* est une variable de type caractère et que *n* est un entier compris entre 0 et 255.

| Nom | Code en Pascal | Rôle | Exemples |
|--------------------|--------------------|---|--|
| ORD(<i>c</i>) | ORD(<i>c</i>) | renvoie le code ASCII du caractère <i>c</i> . Le résultat est un entier positif compris entre 0 et 255. | ORD ("A") vaut 65 ORD ("a") vaut 97 |
| CHR (<i>n</i>) | CHR(<i>n</i>) | renvoie le caractère dont le code ASCII est <i>n</i> . <i>n</i> doit être entre 0 et 255. | CHR (65) vaut "A" CHR (97) vaut "a" |
| SUCC (<i>c</i>) | SUCC(<i>c</i>) | renvoie le caractère successeur de <i>c</i> s'il existe | SUCC ("F") vaut "G" SUCC ("5") vaut "6" |
| PRED (<i>c</i>) | PRED(<i>c</i>) | renvoie le caractère prédécesseur de <i>c</i> s'il existe. | PRED ("Z") vaut "Y" PRED ("9") vaut "8" |
| MAJUS (<i>c</i>) | UPCASE(<i>c</i>) | Convertit le caractère <i>c</i> en majuscule s'il est possible, sinon elle renvoie le même caractère <i>c</i> . | MAJUS("c ") vaut "C" |

NB : *x* : *x* ainsi écrit, représente un objet identifié par l'identificateur *x* ; c'est par exemple une variable.

"*x*" : *x* entre guillemets désigne le caractère *x*.

2 : désigne l'entier 2.

"2" désigne le caractère 2.

CHR (ORD (*c*)) = *c*

ORD (CHR (*n*)) = *n*

I.4. Autres types

Outre les types standards présentés ci-dessus, nous pouvons définir de nouveaux types appelés souvent types utilisateur.

I.4.1. Le type scalaire énuméré

En général, un type scalaire définit une donnée de base ne représentant qu'une seule et unique valeur. C'est le cas du type entier. Par ailleurs, le type scalaire par énumération définit un ensemble ordonné et fini de valeurs désignées par des identificateurs.

Exemple : Nous allons définir un type que nous appelons SAISON contenant les quatre saisons de l'année.

Tableau de déclaration des nouveaux types

| Types |
|--|
| SAISON = (automne, hiver, printemps, été). |

automne, hiver, printemps, et été sont les éléments du type SAISON.

Une variable *s* de type SAISON peut prendre comme valeur : automne, hiver, printemps, et été. L'affectation ci-dessous est correcte.

s ← printemps

L'ordre sur ces valeurs est le suivant : automne < hiver < printemps < été. Par ailleurs, on peut appliquer la fonction Ord sur ces valeurs pour déterminer leurs numéros d'ordre, ainsi Ord (automne) vaut 0 et Ord (hiver) vaut 1 et ainsi de suite.

Les opérateurs applicables à ces valeurs sont :

- Les opérateurs de relation
- Les opérateurs PRED et SUCC. PRED représente le prédécesseur (le précédent) et SUCC représente le successeur (le suivant).

On a par exemple : SUCC (automne) est hiver
 PRED (été) est printemps
 PRED (automne) n'existe pas et SUCC (été) n'existe pas non plus
 et peuvent provoquer des erreurs.

Remarques :

La déclaration du type énuméré IMPAIR = (1, 3, 5) n'est pas possible. Cette déclaration est interdite car 1, 3 et 5 sont des valeurs qui appartiennent au type prédéfini ENTIER.

Dans le langage Pascal, la déclaration d'un type scalaire énuméré et celle d'une variable de ce type se font comme suit :

TYPE <nom_type> = (constante_1, constante_2, ... , constante_n);

VAR <nom_variable> : nom_type ;

Exemple : **TYPE** mois_a_31 = (janvier, mars, mai, juillet, aout, octobre, decembre) ;
VAR mois_long : mois_a_31 ;

Activité 5

Définir en langage Pascal un type énuméré représentant les trois couleurs de base

Réponse

```
TYPE Couleurs_De_Base = (Rouge, Jaune, Bleu) ;
```

I.4.2. Le type intervalle

Le type intervalle possède les propriétés d'un type scalaire discret ordonné (entier, caractère et scalaire énuméré).

La définition d'un intervalle est décrite par la donnée de deux constantes Borne Inférieure et Borne Supérieure appartenant à un type scalaire discret ordonné et telle que Borne Inférieure < Borne Supérieure.

Exemple :**Tableau de déclaration des nouveaux types**

| Types |
|--|
| Indice = 1 .. 10 |
| SAISON = (automne, hivers, printemps, été) |
| Année_Scolaire = automne .. printemps |

Les bornes de Indice sont 1 et 10 prises comme valeurs entières. Une variable de type Indice peut prendre ses valeurs entre 1 et 10.

Les bornes de Année_Scolaire sont automne et printemps. Une variable de type Année_Scolaire peut prendre comme valeur : automne, hivers, et printemps.

Une variable d'un type intervalle possède toutes les propriétés du type de base dont l'intervalle est issu. Toutefois, sa valeur doit être comprise au sens large entre les bornes de l'intervalle.

L'intérêt de ce type réside dans le fait qu'il permet une meilleure lisibilité de l'algorithme.

Exemple : mois : 1 .. 12 est beaucoup plus précis que mois : entier

Dans le langage Pascal, la déclaration d'un type intervalle et celle d'une variable de ce type se font comme suit :

```
TYPE <nom_type> = borne_inf .. borne_sup ;
VAR <nom_variable> : nom_type ;
```

Exemple : TYPE mois = 1 .. 12;
VAR mois_actuel : mois ;

Activité 6

Définissez en langage Pascal un type intervalle représentant les heures possibles d'une journée.

Réponse :

```
TYPE Heures = 0 .. 23 ;
```

Définition : Une chaîne de caractères est une succession de n caractères avec n compris entre 0 et 255.
Si n = 0, on dit que la chaîne est vide.

1.4.3. Le type chaîne de caractères

En algorithmique, les valeurs chaîne de caractères sont définies entre guillemets. En Pascal, les valeurs chaîne de caractères sont définies entre apostrophes.

Exemples :

| Algorithmme | Pascal |
|-----------------------|-----------------------|
| "ceci est un exemple" | 'ceci est un exemple' |
| "l'informatique" | 'l'informatique' |

Les variables chaînes de caractères sont définies par une déclaration pouvant indiquer le nombre maximum de ses caractères.

Exemples :

| Objet | Code | Type / Nature | Objet |
|--------|--------|---------------|---|
| Nom | Nom | Chaîne [10] | Nom d'un individu sur 10 caractères au maximum |
| Snom | Snom | Chaîne | Chaîne pouvant contenir jusqu'à 255 caractères. |
| Invite | invite | Chaîne [18] | Chaîne pouvant contenir jusqu'à 18 caractères. |

VAR

```
Nom : String [10] ;
Snom : String ;
invite : String [18] ;
```

A la variable Nom, on peut affecter une chaîne de caractères ayant au maximum dix caractères, à l'inverse de Snom pour laquelle le nombre de caractères est

visiblement indéfini (ce nombre maximal est de 255 caractères pour certains compilateurs).

Exemples : Nom ← "" {vide}
 invite ← " " {un espace}
 Nom ← "Kamel"
 Snom ← "Nom "
 Snom ← "astucieusement"

Remarques :

On pourra accéder en lecture et en écriture au i ème caractère d'une chaîne CH en utilisant la notation CH [i] où $1 \leq i \leq \text{Long}(\text{CH})$ où Long(CH) désigne la longueur de la chaîne CH.

Nom ← "Rached"
 Nom [1] donne "R"
 Nom [6] donne "d"
 Nom [5] ← "i" Nom devient "Rachid"

Les fonctions et les procédures standards sur les chaînes

La plupart des langages de programmation offrent un jeu de fonctions et de procédures prédéfinies relatives aux chaînes de caractères. Nous citons ci-dessous les plus utilisées

Concat (ch1, ch2, ... , chN) en Pascal CONCAT (ch1, ch2, ... chN)

Fonction qui retourne la concaténation des chaînes ch1, ch2, ..., chN
 C'est l'équivalent de ch1 + ch2 + ch3 + ... chN

Exemples : jj ← "20"
 mm ← "03"
 aaaa ← "2007"
 Date ← Concat (jj, "/", mm, "/", aaaa) donne "20/03/2007"

Long (ch) en Pascal LENGTH (ch)

Fonction qui retourne un entier représentant la longueur en caractères de la chaîne ch. Ce nombre se trouve aussi dans le caractère ch[0].

Exemples : L ← Long ("Bonjour") donne 7
 L ← Long (" Bonjour") donne 8

Sous_chaîne (ch, p, nbc) en Pascal COPY (ch, p, nbc)

Fonction qui retourne une sous-chaîne d'une longueur nbc à partir de la position p dans ch.

Exemples : mot ← "Baccalauréat"
 abrégé ← Sous-chaîne (mot, 1, 3) donne "Bac"

| | | |
|-----------------------|------------------|-----------------------|
| Pos (ch1, ch2) | en Pascal | POS (ch1, ch2) |
|-----------------------|------------------|-----------------------|

Fonction qui retourne la première position de la chaîne ch1 dans la chaîne ch2. Si ch1 n'est pas dans ch2, elle retourne 0.

Exemples : mot1 ← "Baccalauréat"
 mot2 ← "Bac"
 x ← POS (mot2, mot1) donne à x la valeur 1
 y ← POS ("lauréat", mot1) donne à y la valeur 6
 z ← POS ("Lauréat", mot1) donne à z la valeur 0

| | | |
|--------------------------|------------------|--------------------------|
| Efface (ch, p, n) | en Pascal | DELETE (ch, p, n) |
|--------------------------|------------------|--------------------------|

Procédure qui enlève n caractères de ch à partir de la position p.

Exemples : mot ← "CD ROM"
 Efface (mot, 3, 4) mot devient "CD "

| | | |
|-----------------------------|------------------|----------------------------|
| Insère (ch1, ch2, p) | en Pascal | INSERT(ch1, ch2, p) |
|-----------------------------|------------------|----------------------------|

Procédure qui insère la chaîne ch1 dans la chaîne ch2 à partir de la position p. Le caractère numéro p et les suivants sont décalés vers la droite.

Exemples : ch1 ← "DA"
 ch2 ← "DIC"
 Insère (ch1, ch2, 3) ch2 devient "DIDAC"

| | | |
|------------------------|------------------|---------------------|
| Convch (d, ch1) | en Pascal | STR (d, ch1) |
|------------------------|------------------|---------------------|

Procédure qui convertit un nombre décimal d en chaîne de caractères et l'affecte à la variable ch1.

Exemples : Convch (1999, ch) ch contient la chaîne "1999"
 Convch (14.52, ch) ch contient la chaîne "1.4520000000E+01"
 Convch (14.52:8:3, ch) ch contient la chaîne "14.520"

:8:3 n'est qu'un formatage de l'écriture du nombre réel 14.52, on verra ceci en détail dans le chapitre VII.

Valeur (ch, d, erreur)

En Pascal VAL (ch, d, erreur)

Procédure qui convertit une chaîne ch en une valeur numérique décimale et l'affecte à la variable d. Le paramètre erreur est une variable de type entier qui contiendra 0 si la conversion s'est déroulée sans erreur, sinon elle contiendra le numéro (la position) du caractère qui a déclenché l'erreur.

Exemples : ch ← "10.50"
 Valeur (ch, d, e) d contient le nombre 10.5
 e contient 0 (aucune erreur)

ch ← "12/10/1999"
 Valeur (ch, d, e) d contient le nombre 0
 e contient 3 (erreur à la position 3 due au caractère "/" qui n'est pas un chiffre)

Activité 7

Ecrivez une instruction PASCAL permettant d'extraire la sous chaîne "jour" de la chaîne "aujourd'hui" et de l'affecter dans une variable Resultat de type chaîne de caractères.

Réponse : Resultat := COPY ("aujourd'hui", 3, 4) ;

1.5. Opérateurs relationnels

Les opérateurs relationnels les plus utilisés sont :

| Algorithme | Pascal |
|------------------|---------------------|
| <, >, ≤, ≥, ≠, = | <, >, <=, >=, <>, = |

Tous les types que nous avons vus sont des types ordonnés. Toute comparaison entre deux éléments de même type ou de types compatibles est possible. Le résultat de la comparaison est booléen (VRAI ou FAUX).

Exemples : Pour x=32 et y=36
 x < y vaut **VRAI**

II. Les expressions

Définition : On appelle expression toute composition d'opérandes et d'opérateurs réalisant un calcul déterminé.

Nous distinguons deux types d'expression :

- les expressions arithmétiques donnant une valeur numérique.
- les expressions logiques donnant une valeur booléenne.

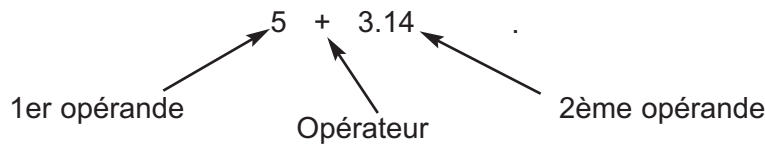
Exemples : $(6 <= 2)$ est une expression logique, sa valeur est FAUX
 $(7+3) \text{ DIV } 2$ est une expression arithmétique, sa valeur est 5

Une expression simple est constituée d'opérandes reliés par des opérateurs.

II.1. Les opérandes

L'opérande est l'élément sur lequel on applique l'opération.

Dans une expression, les opérandes utilisés doivent être de même type ou de types compatibles



Les opérandes peuvent être des constantes, des variables, des valeurs ou des résultats envoyés par des fonctions.

Exemples : $3.14 * D1$ { valeur de D1 connue }
 $(3.14 * D1) + (3.14 * D2)$ { valeurs de D1 et D2 connues }
 $3.14 * \text{CARRE}(R1)$ { valeur de R1 connue }

II.2. Les opérateurs

II.2.1. Les opérateurs arithmétiques

Les opérateurs unaires

Un opérateur est dit unaire s'il est appliqué à un seul opérande. On dit aussi qu'il est monadique.

L'opérateur unaire usuel

| Opérateur | Type opérande | Type résultat |
|-----------|----------------|----------------|
| - | Entier Réal | Entier Réal |

Exemples : $-(5)$ - est l'opérateur et 5 est l'opérande.

Les opérateurs binaires

Un opérateur est dit binaire s'il est appliqué à deux opérandes. On dit aussi qu'il est dyadique

Exemples : $3 + 5$ 3 et 5 sont les opérandes et + est l'opérateur.
 $5 \text{ MOD } 3$ 5 et 3 sont les opérandes et MOD est l'opérateur.

Les opérateurs binaires multiplicatifs

| Opérateur | type opérande 1 | type opérande 2 | type résultat |
|------------|-----------------|-----------------|---------------|
| * | Entier | Entier | Entier |
| | Réel | Réel | Réel |
| | Réel | Entier | Réel |
| | Entier | Réel | Réel |
| / | Entier | Entier | Réel |
| | Entier | Réel | Réel |
| | Réel | Entier | Réel |
| | Réel | Réel | Réel |
| DIV | Entier | Entier | Entier |
| MOD | Entier | Entier | Entier |

Exemples : $100/4.0$ vaut 25.0

Les opérateurs binaires additifs

| Opérateur | type opérande 1 | type opérande 2 | type résultat |
|--------------|-----------------|-----------------|---------------|
| + , - | Entier | Entier | Entier |
| | Réel | Réel | Réel |
| | Réel | Entier | Réel |
| | Entier | Réel | Réel |

Exemples : $5 + 6.2$ vaut 11.2

II.2.2. Les opérateurs relationnels

| Opérateur | Code Pascal | type opérande1 | type opérande2 | type résultat |
|---------------|---------------------|-------------------|-------------------|---------------|
| <, >, =, ≤, ≥ | <, >, =, <>, <=, >= | Tout type ordonné | Tout type ordonné | Logique |

Exemples : Non ($3 < 5$) Non est l'opérateur et $3 < 5$ est l'opérande

Les opérateurs binaires

| Opérateur | type opérande 1 | type opérande 2 | type résultat |
|-----------|-----------------|-----------------|---------------|
| ET | Booléen | Booléen | Booléen |
| OU | Booléen | Booléen | Booléen |
| OUex | Booléen | Booléen | Booléen |

NB : Souvent, les opérateurs logiques sont notés multiplicativement pour le ET et additivement pour le OU.

II.2.3. Les opérateurs logiques

L'opérateur unaire usuel

| Opérateur | type opérande 1 | type résultat |
|-----------|-----------------|---------------|
| NON | Booléen | Booléen |

II.3. Evaluation d'une expression

Comme nous l'avons déjà évoqué dans les expressions logiques, un ordre de priorité est aussi défini dans toute autre expression de calcul.

Cet ordre est le suivant :

- 1) Les parenthèses
- 2) Les opérateurs unaires
- 3) Les opérateurs multiplicatifs
- 4) Les opérateurs additifs
- 5) Les opérateurs relationnels

Exemples : Ordre des opérations de calcul de $3+5*2$

1ère opération $5 * 2 = 10$

2ème opération $3 + 10 = 13$

Remarques :

1- Pour les opérateurs de même priorité, on commence par celui qui est le plus à gauche.

Exemples : $5 + 6 - 2$

Le calcul de $5 + 6 - 2$ commence par $5 + 6$ donc 11 puis $11 - 2$ pour avoir 9.

2- Si on veut imposer un autre ordre, on doit utiliser des parenthèses.

Exemples : $3 + 5 * 2$ vaut 13 mais si on écrit $(3 + 5) * 2$ le résultat est 16

Activité 8

Évaluez les expressions suivantes :

1- $(5 * x + y)$ avec $x = 3$ et $y = 2$

2- $(y + 5 * x)$ avec $x = 3$ et $y = 2$

3- $((y + 5) * x)$ avec $x = 3$ et $y = 2$

Réponse

1- 17

2- 17

3- 21

Retenons:

- Chaque type définit un ensemble de valeurs et un ensemble d'opérations.
- Les types standards sont :
 - les entiers,
 - les réels,
 - les booléens,
 - les caractères,
 - les chaînes de caractères.

Exercices

Exercice 1

Dites si l'on pourra utiliser le type entier pour représenter les quantités suivantes :

- a- Le salaire d'un employé
- b- Le nombre de jours travaillés pendant un mois
- c- Le montant de la prime de transport
- d- Le matricule d'un employé

Exercice 2

Évaluez les expressions suivantes :

a) Pour $a = 3$, $b = 5$ et $c = 8$

$a * (b+c)$, $(a*b) + c$, $a*b + c$, $a+b * c$, $a*b / c+a$

b) $10 + 3 * 5 \text{ DIV } 4$,

$16 + 5*3 - 3*(8 \text{ MOD } 3 + 7)$, Carré (Round (1.3 - 4.5)),

Exercice 3

Évaluez les expressions logiques en A, B, C., et D pour chacune des combinaisons (p,q,r,s) données.

- A. $(p < q) \text{ OU } (r > s)$
- B. $(p < q) \text{ ET NON } (r > s)$
- C. $(p > q) \text{ OU } (r > p)$
- D. $(p + q < r) \text{ ET } (p + q > r)$

Combinaisons données :

- 1) Pour $(p,q,r,s) = (-3, 5, 4, 9)$
- 2) pour $(p,q,r,s) = (3, 7, 4, 9)$
- 3) Pour $(p,q,r,s) = (5, 13, 7, 3)$

Exercice 4

Écrivez les instructions PASCAL permettant de réaliser les objectifs suivants

- 1) Obtenir la longueur de la chaîne «Montagne»
- 2) Récupérer la sous chaîne de 3 caractères à partir de la position 4 de la chaîne «Montagne»
- 3) Effacer les 3 premiers caractères de la chaîne «Bonjour»

Leçon 3

Les tableaux

Objectifs spécifiques

- Déclarer et manipuler des tableaux unidirectionnels

Plan de la leçon

- I- Déclaration d'un tableau
- II- Le type tableau

Retenons
Exercices

Leçon 3

Les tableaux

I. Déclaration d'un tableau

Activité 1

Nous voulons calculer puis trier dans l'ordre décroissant les moyennes de cinq élèves. A première vue, nous allons utiliser cinq variables de type réel Moy1, Moy2, Moy3, Moy4 et Moy5. Nous remarquons que ce nombre de variables deviendra encore beaucoup plus grand si nous avons 30 élèves et dans le tri nous serons amenés à manipuler autant de variables. Cela rendra les algorithmes très complexes et les tris impossibles. Cependant, ces variables moyennes ont toutes le même rôle (moyenne d'un élève); nous pouvons les regrouper sous le même nom MOYENNE, nous les distinguerons avec des indices de 1 à 5. Ce regroupement formera une nouvelle structure dite tableau ou vecteur de 5 éléments.

| Moyenne | | | | | |
|---------|---|---|---|---|---|
| Indice | 1 | 2 | 3 | 4 | 5 |

Moy1 Correspondra au 1^{er} élément du tableau MOYENNE
 Moy2 Correspondra au 2^{ème} élément du tableau MOYENNE
 Moy3 Correspondra au 3^{ème} élément du tableau MOYENNE
 Moy4 Correspondra au 4^{ème} élément du tableau MOYENNE
 Moy5 Correspondra au 5^{ème} élément du tableau MOYENNE

La déclaration du tableau se fera comme suit :

| Objet | Type / nature | Rôle |
|---------|--------------------|---|
| MOYENNE | Tableau de 5 réels | Tableau servant à contenir les moyennes de 5 élèves |

Dans le langage Pascal, on déclare ce tableau comme suit :

VAR

MOYENNE : **ARRAY**[1..5] OF REAL;

Dans le cas général, pour déclarer un tableau, on utilisera la forme suivante :
 Au niveau de l'analyse et de l'algorithme

Tableau de déclaration des objets

| Objet | Type / nature | Rôle |
|---------------|--------------------------------------|------|
| Ident_tableau | Tableau de taille et de type_élément | |

En Pascal**VAR**

```
ident_tableau : ARRAY[Borne_inf..Borne_sup] OF Type_élément;
```

Où :

Ident_tableau : Identificateur n nom du tableau.

borne_Inf .. Borne_Sup : intervalle correspondant à l'ensemble des valeurs des indices du tableau.

Borne_Inf : Borne inférieure de l'intervalle des indices.

Borne_Sup : borne supérieure de l'intervalle des indices.

Type_élément : type des éléments du tableau. Il peut être l'un des types vu précédemment.

Exemples : T : ARRAY [1..10] OF REAL ; {déclaration d'un tableau de 10 éléments réels }

Un tableau est une structure de données permettant de ranger un nombre fini d'éléments de même type.

Un tableau est caractérisé par ses dimensions. Nous nous limiterons à l'étude des tableaux unidimensionnels appelés aussi vecteurs.

L'accès à un élément d'un tableau est direct.

Reprenons l'activité 12

```
Moyenne [1] ← 12.30
Moyenne [2] ← 11.55
Moyenne [3] ← 13.60
Moyenne [4] ← 14.10
Moyenne [5] ← 10.38
```

Pour accéder au $i^{\text{ème}}$ élément d'un tableau, il suffit de donner l'identificateur du tableau et l'indice i indiquant le rang de l'élément. Cet indice doit être dans l'intervalle Borne_inf .. Borne_sup.

Remarques :

1- Nous pouvons lire et écrire un élément du tableau (voir chapitre IX). Nous pouvons modifier un élément par une instruction d'affectation.

Exemples : moyenne [4] ← 13.85

2- Les opérations possibles sur un élément du tableau sont les mêmes que celles définies sur une variable de même type.

II. Le type tableau

Il est possible de déclarer le type d'un tableau.

Exemple

Tableau de déclaration des nouveaux types

| Types |
|--|
| Ouvriers = tableau de 20 chaînes de caractères |
| T_Salaire = tableau de 20 réels |
| T_Compte = tableau de 26 entiers |

Tableau de déclaration des objets

| Objet | Type / Nature | Rôle |
|--------|---------------|---|
| T_OUV | Ouvriers | Tableau des 20 noms d'ouvriers |
| TS | T_Salaire | Tableau des 20 salaires des ouvriers |
| Compte | T_Compte | Tableau comptant le nombre de chaque lettre de 'A' à 'Z' du texte |

Activité 13

Soit la séquence suivante :

$$V[1] \leftarrow 20$$

$$V[2] \leftarrow 3$$

$$V[3] \leftarrow V[1] \text{ DIV } V[2]$$

$$V[4] \leftarrow V[3] * V[2]$$

$$V[5] \leftarrow V[1] + V[3] + V[4]$$

1) Déclarer le tableau V.

2) Quel est le contenu de chaque élément du tableau V ?

Réponse

1) Déclaration

Tableau de déclaration de nouveaux types

| Types |
|--------------------------------|
| vecteur = tableau de 5 entiers |

Tableau de déclaration des objets

| Objet | Type / Nature | Rôle |
|-------|---------------|-----------------------|
| V | vecteur | Tableau de 5 éléments |

2) Le contenu de chaque élément :

| | | | | | |
|---------------|----|---|----------|----|----|
| V | 20 | 3 | 6 | 18 | 44 |
| Indice | 1 | 2 | 3 | 4 | 5 |

Retenons:

- Les structures de données vues dans cette leçon sont les tableaux (suite de variables repérées par des indices).

Exercices

Exercice 1

Pour une classe comptant 30 élèves, élaborer un tableau de déclaration des objets relatifs aux noms des élèves, leurs notes en informatique, leurs notes en français et leurs notes en mathématiques.

Exercice 2

Reprenez l'exercice 1 en déclarant cette fois-ci un tableau de déclaration des nouveaux types et un tableau de déclaration des objets les utilisant.

Exercice 3

Soit le tableau de temps exprimé en heures

| | | | | | |
|---------------|---|----|---|---|---|
| Temps | 7 | 14 | 6 | 1 | 9 |
| Indice | 1 | 2 | 3 | 4 | 5 |

- 1) Donnez les cinq affectations permettant de remplir le tableau TEMPS par les données précédentes.
- 2) Rangez, dans un deuxième tableau TMN, les équivalences des éléments de TEMPS en minutes.
- 3) Stockez, dans un troisième tableau TS, les équivalences des éléments de TEMPS en secondes.

Exercice 4

- 1) Soit A et B deux tableaux d'entiers, de types respectifs TAB1 et TAB2 et de dimensions respectives 4 et 3.
- 2) Soit la séquence d'affectations suivante :

```

A [ 1 ] ← 20
A [ 2 ] ← 10
B [ 1 ] ← A [ 1 ] + A [ 2 ]
A [ 3 ] ← B [ 1 ] DIV 2
B [ 2 ] ← B [ 1 ] MOD A [ 1 ]
B [ 3 ] ← A [ 2 ] * 2 + 2
A [ 4 ] ← (A [ 2 ] DIV 60) MOD 60
A [ 14 ] ← A [ 4 ]

```

Questions

- 1) Déclarez les deux tableaux A et B.
- 2) Trouvez les erreurs d'affectation dans la séquence précédente.
- 3) Quel est le contenu de chaque élément des deux tableaux A et B ?



Chapitre 7

Les actions élémentaires simples

Objectifs :

- Utiliser les structures simples pour résoudre des problèmes.

Plan du chapitre :

Leçon 1 : L'affectation

Leçon 2 : Les opérations de sortie et les opérations d'entrée



Leçon 1

L'affectation

Objectifs spécifiques :

- Utiliser les structures simples pour résoudre des problèmes.
- Apprendre la manipulation de l'affectation.

Plan de la leçon

- I. Définition
- II. Vocabulaire et syntaxe

Retenons

Exercices

Leçon 1

L'affectation

Nous avons vu dans le chapitre "structures de données" la notion d'affectation. Convenons que cette instruction ou cette structure de contrôle algorithmique est l'une des trois actions élémentaires.

I. Définition

L'affectation est l'action qui consiste à attribuer une valeur à une variable. On convient de noter l'affectation par le symbole " \leftarrow ". La variable réceptrice est placée à gauche du symbole et la valeur ou l'expression donnant cette valeur est placée à sa droite.

Exemples :

| | | |
|---------------|--|---|
| L'instruction | $n \leftarrow 15$ | Après exécution de l'instruction, la valeur de la variable n vaut 15. |
| L'instruction | $x \leftarrow 10.25$ | Après exécution de l'instruction, la valeur de la variable x vaut 10.25. |
| L'instruction | $\text{test} \leftarrow \text{Vrai}$ | Après exécution de l'instruction, la valeur de la variable test vaut Vrai. |
| L'instruction | $\text{ch} \leftarrow \text{"Pascal"}$ | Après exécution de l'instruction, la valeur de la variable ch vaut "Pascal". |

Remarques

- Le nom d'une variable est un mot qui commence par une lettre et non pas par un chiffre.
- Après l'affectation d'une valeur à une variable, l'ancien contenu de la variable est perdu.

II. Vocabulaire et syntaxe

L'instruction d'affectation est notée :

| | Niveau analyse et algorithme | Niveau Pascal |
|-----------------------|--|--|
| Forme générale | $\text{variable} \leftarrow \text{expression}$ | $\text{variable} := \text{expression} ;$ |
| Exemple | $x \leftarrow 10,25$ | $x := 10.25 ;$ |

Remarques

- ✓ La valeur de l'expression doit être compatible avec le type de la variable.
- ✓ Il est possible d'affecter à une variable la valeur d'une autre variable.
- ✓ L'expression est calculée et la valeur trouvée est affectée à la variable.

Exemples :

| | |
|----------------------|---|
| $x \leftarrow y$ | La valeur de la variable x devient égale à la valeur de la variable y . |
| $x \leftarrow x + 1$ | La valeur de la variable x devient égale à sa valeur actuelle incrémentée de 1. |

Activité 1

Sachant que x et y sont deux variables réelles. Soit la séquence d'instructions ci-dessous :

- 1) $x \leftarrow 10.25$
- 2) $y \leftarrow 50.5$
- 3) $z \leftarrow x$
- 4) $x \leftarrow y$
- 5) $y \leftarrow z$

1. Remplissez le tableau ci-dessous.

| N° de séquence | Valeur de x | Valeur de y | Valeur de z |
|----------------|---------------|---------------|---------------|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |

2. Quelles sont les valeurs finales de x et de y après exécution de la séquence ci-dessus.
3. Que fait cette séquence ?.

Réponse :

| N° de séquence | Valeur de x | Valeur de y | Valeur de z |
|----------------|---------------|---------------|---------------|
| 1 | 10,25 | | |
| 2 | 10,25 | 50,5 | |
| 3 | 10,25 | 50,5 | 10,25 |
| 4 | 50,5 | 50,5 | 10,25 |
| 5 | 50,5 | 10,25 | 10,25 |

2- La valeur de x devient 50.5 et la valeur de y devient 10.25.

3- Cette séquence permet de permuter les deux variables en utilisant une variable auxiliaire.

Activité 2

Sachant que x et y sont deux variables réelles. Soit la séquence d'instructions ci-dessous:

- 1) $x \leftarrow 20$
- 2) $y \leftarrow 50$
- 3) $x \leftarrow x + y$
- 4) $y \leftarrow x - y$
- 5) $x \leftarrow x - y$

1. Remplissez le tableau ci-dessous :

| N° de séquence | Valeur de x | Valeur de y |
|----------------|---------------|---------------|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |

2. Quelles sont les valeurs finales de x et de y après exécution de la séquence ci-dessus.

3. Que fait cette séquence ?

Réponse :

1-

| N° de séquence | Valeur de x | Valeur de y |
|----------------|---------------|---------------|
| 1 | 20 | |
| 2 | 20 | 50 |
| 3 | 70 | 50 |
| 4 | 70 | 20 |
| 5 | 50 | 20 |

2-La valeur de x devient 50 et la valeur de y devient 20.

3- Cette séquence permet de permuter les deux variables sans utiliser une variable auxiliaire.

Activité 3

Sachant que t , h , m et s quatre variables de type entier. Soit la séquence ci-dessous :

- 1) $t \leftarrow 3665$
- 2) $h \leftarrow t \text{ div } 3600$
- 3) $m \leftarrow (t \text{ mod } 3600) \text{ div } 60$
- 4) $s \leftarrow t \text{ mod } 60$

1. Déterminez la valeur de h , m et s après exécution de la séquence ci-dessus.

2. Que fait cette séquence ?

Réponse :

1-

- La valeur de h est égale à 1.
- La valeur de m est égale à 1.
- La valeur de s est égale à 5.

2- Cette séquence convertit un temps donné en secondes en son équivalent en heures, minutes et secondes.

Activité 4

Exécutez la séquence d'instruction ci-dessous, sachant que ch est une variable de type chaîne de caractères, p et lg sont deux variables de type entier.

- 1) $ch \leftarrow \text{"programmation"}$
- 2) $p \leftarrow \text{pos}(m, ch)$
- 3) $\text{Supp}(ch, p + 2, 5)$
- 4) $\text{Insère}(\text{"e"}, ch, 9)$
- 5) $ch[1] \leftarrow \text{majus}(ch[1])$
- 6) $lg \leftarrow \text{long}(ch)$

Activité 5

Soit la séquence suivante :

- 1) $T[1] \leftarrow 20$
- 2) $T[2] \leftarrow 4$
- 3) $T[3] \leftarrow T[1] \text{ Div } T[2]$
- 4) $T[4] \leftarrow 2 * T[3]$
- 5) $T[5] \leftarrow T[4] + 2 * T[2]$

1. Quel est le type de l'objet T ?

2. Après exécution de la séquence ci-dessus, quel est le contenu de chacun des 5 premiers éléments de T.

Réponse :

1. T est un tableau d'au moins cinq entiers.

2-

| | | | | | |
|----------------|----|---|---|----|----|
| T [i] | 20 | 4 | 5 | 10 | 18 |
| i | 1 | 2 | 3 | 4 | 5 |

Retenons

- L'action d'affectation consiste à attribuer une valeur à une variable.
- L'instruction d'affectation est désignée ou exprimée par le symbole " \leftarrow ".
- La valeur à affecter à une variable doit être compatible avec le type de la variable.

Exercices

Exercice 1

Soit la déclaration des variables suivantes :

| Variable | Type |
|----------|----------------------|
| A | Réel |
| Y | caractère |
| Ch | Chaîne de caractères |
| Test | Booléen |
| C | caractère |

Dans la case de chacune des affectations ci-dessous, mettez V si l'opération est permise et F dans le cas contraire.

| Affectation | V / F | Affectation | V / F |
|--------------------------|-------|----------------------|-------|
| $A \leftarrow 3,14$ | | $A \leftarrow 0$ | |
| $Y \leftarrow "1"$ | | $Y \leftarrow "BAC"$ | |
| $Ch \leftarrow "1"$ | | $C \leftarrow "B"$ | |
| $test \leftarrow "VRAI"$ | | $test \leftarrow 1$ | |

Exercice 2

Soit la séquence d'instructions ci-dessous :

- 1) $x \leftarrow 10$
- 2) $y \leftarrow 13.75$
- 3) $z \leftarrow (2 * y + x) / 3$
- 4) $d \leftarrow "Moyenne = "$

1- Remplissez le tableau ci-dessous :

| Données | Type | Nature |
|---------|--------|--------|
| x | Entier | |
| y | | |
| z | | |
| d | | |

2- Exécutez à la main la séquence précédente.

Exercice 3

On donne la séquence d'instructions suivantes :

1)- $x \leftarrow ((A < B) \text{ OU } (A < 2)) \text{ ET } (A > 0)$

2)- $y \leftarrow (A < B) \text{ OUex } ((B < 2) \text{ ET } (A > 0))$

Remplissez le tableau ci-dessous par les valeurs de X et de Y selon les cas.

| (A, B) | (0, -1) | (0,0) | (-1,1) |
|--------|---------|-------|--------|
| x | | | |
| y | | | |

Exercice 4

Soit la séquence d'instructions ci-dessous :

1)- $T[1] \leftarrow \text{Pred}("N")$

2)- $T[2] \leftarrow \text{Succ}(\text{Car}(\text{Ord}("A") - 1))$

3)- $T[3] \leftarrow "T"$

4)- $T[4] \leftarrow \text{Car}(\text{Ord}("H"))$

Déclarez le tableau T :

| O.U. | Type |
|------|------|
| T | |

Ecrivez les valeurs finales de T.

| | | | | |
|-----------|---|---|---|---|
| Tableau T | | | | |
| Indice | 1 | 2 | 3 | 4 |

Leçon 2

Les opérations de sortie et les opérations d'entrée

Objectifs spécifiques :

- Savoir faire une lecture ou une écriture de données.
- Présenter les solutions sous forme d'un algorithme puis d'un programme.

Plan de la leçon

- I. L'opération de sortie
 1. Définition
 2. Vocabulaire et syntaxe
- II. L'opération d'entrée
 1. Définition
 2. Vocabulaire et syntaxe
- III. Activités
 1. Activité 1
 2. Activité 2

Retenons
Exercices

Leçon 2

Les opérations de sortie et les opérations d'entrée

I. L'opération de sortie

I.1- Définition

L'opération de sortie d'une donnée consiste en l'écriture de cette donnée sur un périphérique de sortie.

I.2- Vocabulaire et syntaxe

Pour faire une suite de sortie on l'écrit de la façon suivante :

Ecrire (S1, S2, ..., Sn)

où Si peut être : – le nom d'une variable (c'est son contenu qui sera écrit),
– un message (entre guillemets. Il sera écrit tel qu'il est),
– une expression de calcul (c'est sa valeur qui sera écrite).

Pour afficher la valeur d'une variable x, on appliquera la syntaxe suivante :

Niveau analyse et algorithme : Ecrire (x)

Niveau Pascal : Write (x) ;

Pour passer à la ligne suivante, on peut utiliser la procédure WRITELN qui force le passage à la ligne suivante pour le prochain affichage Writeln (x) ;

Exemple 1 : a := 10;
 b := 5;
 Write ('a x b = ', a x b) ;

Le résultat obtenu sur écran est : a x b = 50

N.B. En Pascal, les guillemets d'un message seront remplacés par des apostrophes.

Exemple 2 : Write(25) ;
 Write('Info') ;
 Le résultat obtenu sur écran est : 25Info

II. L'opération d'entrée

II.1- Définition

L'opération d'entrée consiste en la lecture d'une donnée via un périphérique d'entrée.

II.2- Vocabulaire et syntaxe

Pour saisir ou lire une donnée et l'affecter à une variable nommée x, on appliquera la syntaxe suivante :

Niveau analyse : x = donnée

Niveau algorithmique : Lire (x)

Niveau Pascal : Readln (x) ;

En Pascal, la lecture est symbolisée par le mot READLN. C'est la procédure READLN qui transfère les nombres ou chaînes de caractères du clavier vers la mémoire centrale.

Pascal admet aussi la procédure READ (qui a le même effet que READLN sans passage à la ligne pour la prochaine saisie) mais il s'est montré à l'usage.

II.3- Remarque

Dès que le programme rencontre une instruction de lecture, l'exécution s'arrête en attendant une donnée saisie à partir du clavier. La frappe sur la touche Entrée valide la saisie de donnée et reprend l'exécution de la suite du programme.

III. Activités

III.1- Activité 1

Calculez puis affichez l'intérêt i et la valeur acquise par le capital Ci placé pendant un an à intérêt simple de taux t donné.

Sachant que :

– l'intérêt est égale au capital initial multiplié par le taux sur 360. C'est à dire $i = \frac{C_i * t}{360}$

– le capital final Cf est égal au capital initial Ci plus l'intérêt i. C'est à dire $C_f = C_i + i$.

- Questions :**
- 1- Faites une analyse du problème intitulé Prg1.
 - 2- Déduisez l'algorithmique de cette analyse.
 - 3- Traduisez cet algorithme en Pascal.

Analyse :

| | Nom = prg1 | |
|---|--------------------------------|------|
| S | L. D. E. | O.U. |
| 5 | Résultat = Ecrire (i, Cf) | i |
| 3 | $i \leftarrow (C_i * t / 360)$ | Cf |
| 2 | Ci = donnée | Ci |
| 1 | t = donnée | t |
| 4 | $C_f \leftarrow C_i + i$ | |
| 6 | Fin prg1 | |

Tableau de déclaration des objets :

| O.U. | Type |
|------|------|
| i | Réel |
| Cf | Réel |
| Ci | Réel |
| t | Réel |

Algorithme :

- 0) Début prg1
- 1) Lire (Ct)
- 2) Lire (Ci)
- 3) $i \leftarrow Ci * t / 360$
- 4) $Cf \leftarrow Ci + i$
- 5) Ecrire (i, Cf)
- 6) Fin prg1

Traduction en Pascal :

En Pascal, on doit passer par trois étapes : la saisie, la compilation puis l'exécution. Pour saisir un programme, on a besoin d'un éditeur de textes. Le langage Pascal possède souvent son propre éditeur de textes. Il suffit de lancer Pascal et celui-ci crée un nouveau fichier vide portant le nom "noname00.Pas". Avant de commencer la saisie, il vaut mieux enregistrer le programme sous un nouveau nom et le mettre à jour au fur et à mesure. On peut suivre les étapes suivantes :

- . Pour lancer Pascal sous Windows aller vers Démarrer/programmes/Turbo Pascal/TPW ou faire un double clic sur son raccourci s'il existe sur le bureau.
- . Pour enregistrer le programme sous le nom demandé, aller vers fichier / enregistrer-sous, sélectionner l'emplacement, puis saisir le nom du programme puis valider.

Remarque

Le langage Pascal ne fait pas de différence entre une lettre écrite en minuscule ou en majuscule dans les mots réservés. Par exemple, a et A représente le même objet.

```

Turbo Pascal - [d:\livrev-2\chapvi-1\prg1.pas]
Fichier  Editer  Chercher  Exécuter  Compiler  Options  Fenêtre  Aide

Program prg1 ;
Uses
  WinCrt ;
Var
  i, t, ci, cf: Real ;
BEGIN
  Readln(t) ;
  Readln(ci) ;
  i := ci*t / 360 ;
  cf := ci + i ;
  Writeln(i, cf) ;
End.
12:42  Insertion

```

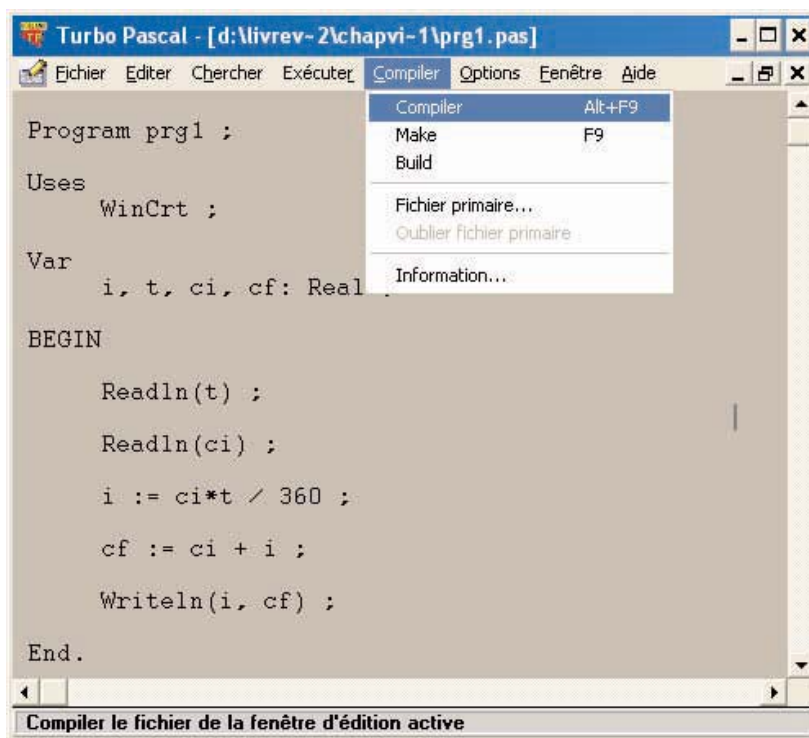
A retenir

- Un programme Pascal se compose de 2 parties :
 - une partie déclarative commençant par le mot PROGRAM suivi de l'identificateur du programme (son nom)
 - une partie appelée programme principal, délimitée par les mots clé **BEGIN** et **END**.
- La 1^{ère} partie commence par le mot réservée **PROGRAM** suivi du nom du programme.
- Un nom ou identificateur en Pascal doit débuter par une lettre suivie par un nombre quelconque de **lettres, chiffres** ou de "_" (**caractère souligné**). Les identificateurs ne peuvent contenir d'espacement (caractère "blanc") ou de caractères spéciaux tels que %, ?, *, ., -, ...

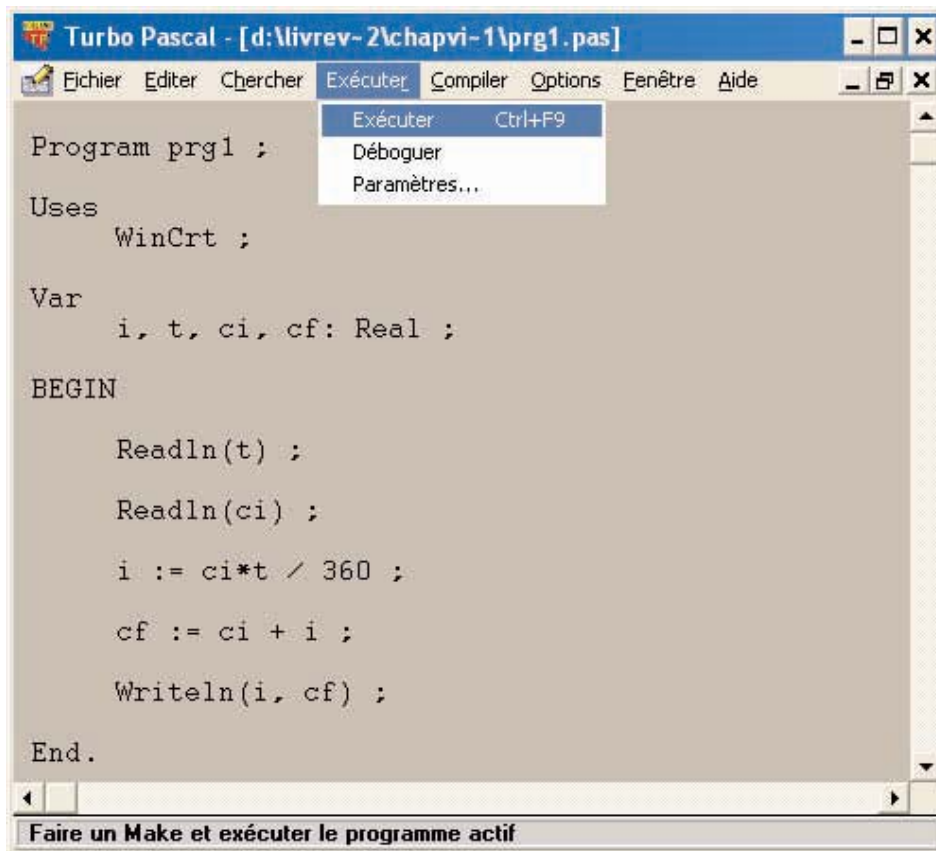
N.B. Les variables doivent faire l'objet d'une déclaration de type de la forme :
 VAR identificateur : type;

- Les instructions en Pascal sont séparées par un point virgule, même chose pour les déclarations.
- Les instructions de lecture et d'écriture se traduisent respectivement par **READLN** et **WRITE** (ou **WRITELN**) suivies d'une liste de variables ou d'expressions placées entre parenthèses et séparées par des virgules.

- L'ajout de **LN** après **WRITE** (**WRITELN**) force le retour du curseur au début de la ligne suivante quand il aura terminé l'affichage en cours.
 - L'affectation en Pascal se représente par ":=".
 - Les opérateurs arithmétiques sont identiques à ceux du langage de description algorithmiques (LDA). En effet, outre les quatre opérations + - * / , Pascal utilise deux opérateurs supplémentaires :
 - DIV** fournissant le quotient de la division euclidienne de deux nombres entiers.
 - MOD** fournissant le reste de la division euclidienne de deux nombres entiers.
- Exemple** : 23 / 5 fournit la valeur 4.6
 23 **DIV** 5 fournit 4
 23 **MOD** 5 fournit 3.
- Les mots clés **PROGRAM**, **USES**, **VAR**, **BEGIN**, **END**, ... ont un sens précis dans le langage: ce sont des **mots réservés** qui ne peuvent être choisis comme nom d'objets utilisateurs par le programmeur et les mots tels que **INTEGER**, **READLN**, **WRITE**, ... ont une signification prédéfinie. Pour éviter toute erreur, il faut éviter de les choisir comme identificateurs. Toutefois, il est souhaitable que l'identificateur soit significatif.
 - Les mots clés du langage et les identificateurs doivent être séparés les uns des autres par un ou plusieurs espaces.



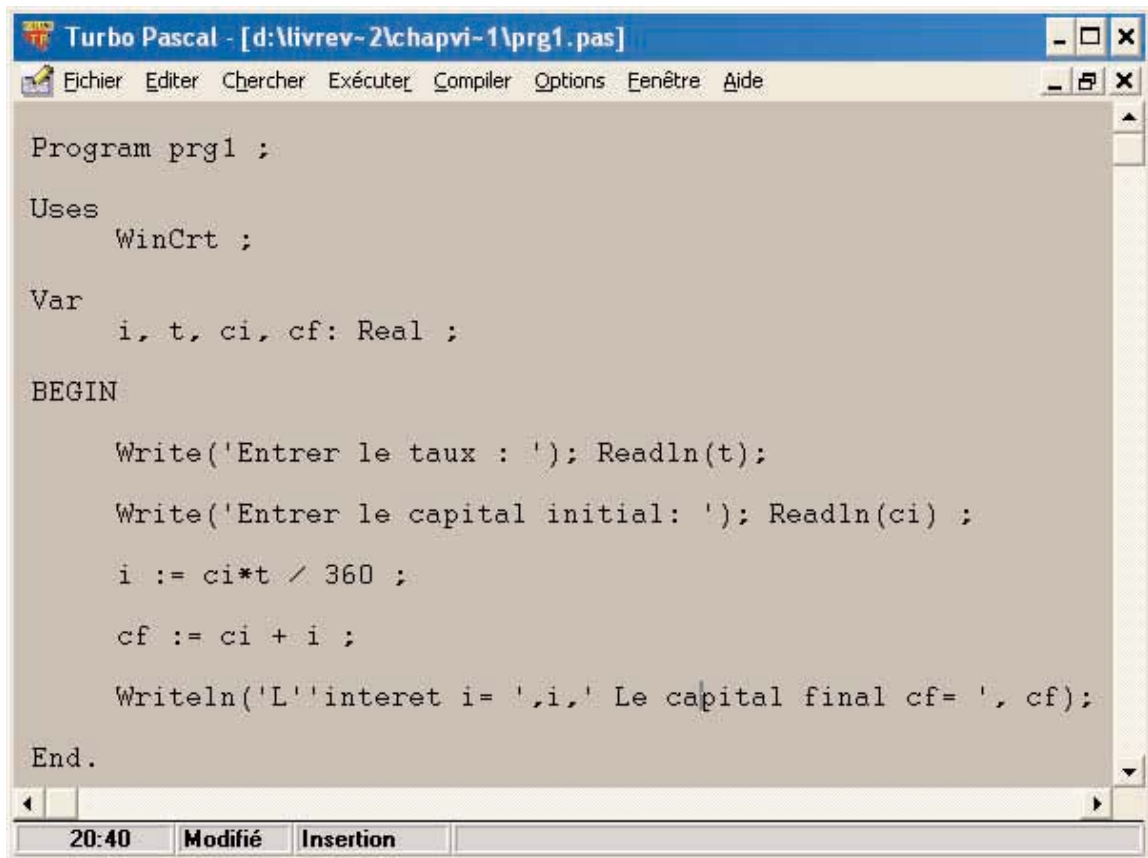
- Compilation du programme : On peut compiler le programme de la façon suivante : menu compilation/compiler ou directement Alt+F9.



■ Exécution du programme : On peut exécuter le programme de la façon suivante : menu exécuter/exécuter ou Ctrl + F9.

A retenir

Avant de lire ou d'afficher, il est conseillé d'afficher des commentaires à l'écran afin de prévenir l'utilisateur de ce qu'il doit faire.



```

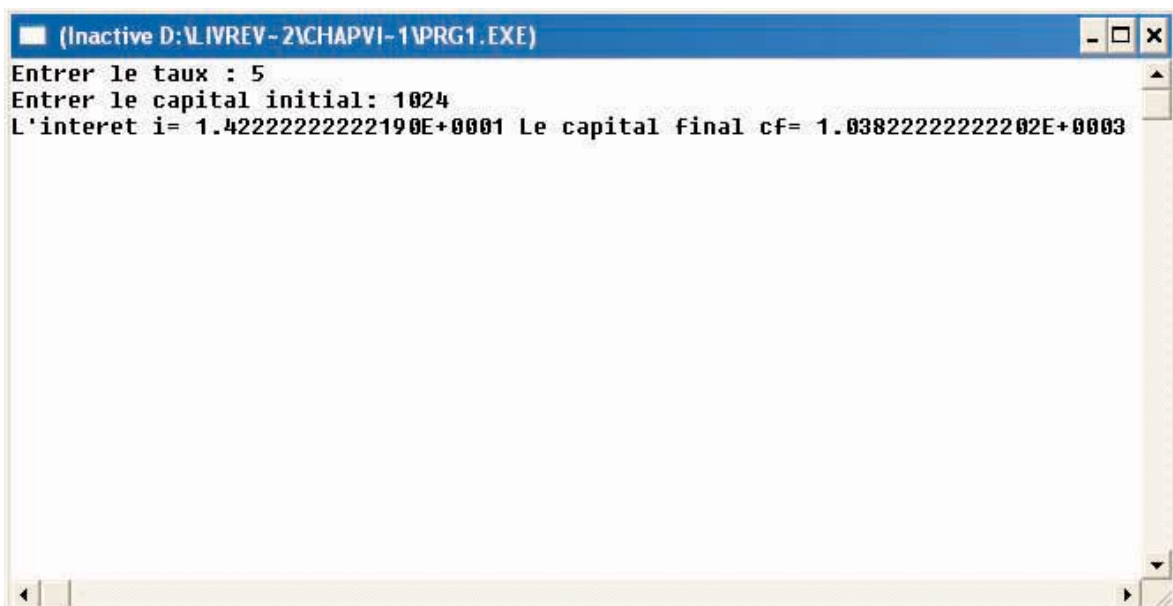
Turbo Pascal - [d:\livrev-2\chapvi-1\prg1.pas]
Fichier Editer Chercher Exécuter Compiler Options Fenêtre Aide

Program prg1 ;
Uses
  WinCrt ;
Var
  i, t, ci, cf: Real ;
BEGIN
  Write('Entrer le taux : '); Readln(t);
  Write('Entrer le capital initial: '); Readln(ci) ;
  i := ci*t / 360 ;
  cf := ci + i ;
  Writeln('L'interet i= ',i,' Le capital final cf= ', cf);
End.
20:40 Modifié Insertion

```

Formatage de l'affichage

Revenons à l'activité précédente et observons les résultats produits



```

(Inactive D:\LIVREV-2\CHAPVI-1\PRG1.EXE)
Entrer le taux : 5
Entrer le capital initial: 1024
L'interet i= 1.4222222222190E+0001 Le capital final cf= 1.0382222222202E+0003

```

- L'affichage des données : le taux est 5, la somme initiale est 1024.
- L'affichage des résultats : l'intérêt est 1,4222222222E+0001, le capital final est 1.0382222222202E+0003.

Par défaut, l'écran est divisé en colonnes de largeur 10 caractères. L'affichage des nombres se fait à partir de la droite de la colonne en cours.

WRITE (valeur_entière : n) affiche la valeur entière sur n positions (insertion d'espace-ment à gauche du nombre s' il y a trop peu de chiffres et ajustement automatique, si n est insuffisant).

WRITE(valeur_réelle) affiche le nombre en notation scientifique (x.xxxxxE+x précédé d'un espacement).

WRITE(valeur_réelle : n) affiche le nombre en notation scientifique sur n positions.

WRITE(valeur_réelle : n1 : n2) affiche le nombre avec n2 décimales avec n1 positions.

WRITE(chaîne : n) affiche la chaîne de caractères sur n positions (insertion d'espace-ment à gauche de la chaîne s'il y a trop peu de caractères et ajustement automatique, si n est insuffisant).

Exemples :

Si la variable entière x contient 12345, ("□" : désigne un espace)

WRITE (x) affiche 12345

WRITE (x : 8) affiche □□□□12345

WRITE (x : 2) affiche 12345

Si la variable réelle x contient 123.4567, ("□" : désigne un espace)

WRITE (x) affiche □1.23456E+2

WRITE (x : 7) affiche □1.2E+2

WRITE (x : 8 : 2) affiche □□123.46

WRITE (x : 2) affiche 1.2E+2

Si la variable du type chaîne x contient 'PASCAL', ("□" : désigne un espace)

WRITE (x) affiche PASCAL

WRITE (x : 8) affiche □□PASCAL

WRITE (x : 3) affiche PASCAL

Activité 2

Ecrire un programme qui saisit le nom et le prénom de l'utilisateur puis le nom complet formé par "prénom-nom".

1. Faites une analyse du problème intitulé élève.
2. Déduisez l'algorithme relatif à cette analyse.
3. Traduisez cet algorithme en Pascal.

Analyse

| Nom = élève | | |
|-------------|--|--------------|
| S | L. D. E. | O.U. |
| 4 | Résultat = Ecrire ("Votre nom complet est ", Ch) | Ch |
| 3 | Ch ← prenom + " " + nom | Ch prenom |
| 1 | prenom = donnée (" Entrer votre prénom") | nom |
| 2 | nom = donnée (" Entrer votre nom") | |
| 5 | Fin élève | |

Tableau de déclaration des objets :

| Nom | Type/Nature | Rôle |
|--------|-------------|---------------------|
| Ch | chaîne | le prénom et le nom |
| prenom | chaîne | le prénom |
| nom | chaîne | le nom |

Algorithme :

- 0) Début élève
- 1) Ecrire ("Entrer votre prénom"), Lire(prenom)
- 2) Ecrire ("Entrer votre nom"), Lire (nom)
- 3) Ch ← prenom + " " + nom
- 4) Ecrire ("Votre nom complet est ", Ch)
- 5) Fin élève

Traduction en Pascal :

```

PROGRAM eleve ;
USES    WinCrt ;
VAR     Ch, prenom, nom : String ;
BEGIN
    Write('Entrer votre prénom : ') ; ReadLn (prenom) ;
    Write ('Entrer votre nom : '); ReadLn (nom);
    Ch := prenom + " " + nom ;
    Writeln('Votre complet est : ', Ch) ;

End.

```

Retenons

- ✍ Une entrée consiste à affecter à partir du clavier une valeur dans une variable.
- ✍ Pour qu'un utilisateur donne une nouvelle valeur à une variable nommée x, il appliquera les syntaxes suivantes :
 - Niveau analyse : x = donnée
 - Niveau algorithme : Lire (x)
 - Niveau Pascal : ReadLn (x);
- Une sortie regroupe toutes les opérations de sortie des résultats sur écran.
- Pour que l'utilisateur affiche la valeur d'une variable nommée x, il appliquera les expressions ou instructions suivantes :
 - * Affichage sans commentaire :
 - Niveau analyse et algorithme : Ecrire (x)
 - Niveau Pascal : Write (x); ou Writeln (x);
 - * Affichage avec commentaire :
 - Niveau analyse et algorithme : Ecrire ("commentaire ", x)
 - Niveau Pascal : Write ('commentaire ',x);
 - ou Writeln ('commentaire ', x);
- Une structure est dite simple si elle est réduite à l'une des actions suivantes :
- Une sortie ou écriture de données.
- Une entrée ou lecture de données.

Exercices

Pour les exercices ci-dessous, on demande :

1. Faire une analyse du problème
2. Déduire l'algorithme de cette analyse
3. Traduire cet algorithme en Pascal.

Exercice 1

Affichez le résultat de la permutation de deux entiers donnés.

Exercice 2

Calculez puis affichez le bénéfice fait sur la vente d'un produit. Sachant que le prix d'achat, le prix de vente et les frais sont donnés.

Exercice 3

Convertissez puis affichez une capacité donnée en Octets en son équivalente exprimée en Mégaoctets puis en Gigaoctets.

Exercice 4

Convertissez puis affichez une mesure de température donnée en degré Celsius en son équivalente, exprimée en Kelvin puis en Fahrenheit.

Sachant que : $T^{\circ} F = 32 + 1.8 \times T^{\circ} C$ et $T^{\circ} K = 273 + T^{\circ} C$.

Exercice 5

Calculez puis affichez la distance euclidienne entre deux données du plan.

Soit les points M (a, b) et N (c, d); la distance entre eux est donnée par la formule suivante

$$d(M,N) = \sqrt{(a - c)^2 + (b - d)^2}$$

Exercice 6

Calculez puis affichez les valeurs des fonctions f et g tels que

$$f(x) = 2 * x^3 + 8 * x - 40, \quad g(x) = \sin(x) - \cos(2 * x^2) \text{ avec } x \text{ un réel donné.}$$

Exercice 7

Calculez puis affichez la vitesse d'une automobile qui démarre avec une accélération constante pour t secondes données

La distance parcourue est $d = \frac{1}{2} * \delta * t^2$

La vitesse est $v = \delta * t$

Exercice 8

Extraire puis afficher les chiffres de centaines, dizaines et unités d'un entier donné composé de trois chiffres.

Exercice 9

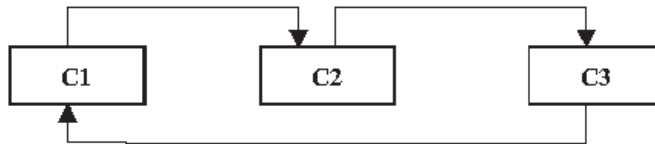
Calculez puis affichez la somme de deux fractions données par leurs numérateurs et leurs dénominateurs de type entier. Le résultat doit être donné sous forme d'une fraction : numérateur et dénominateur.

$$\text{Sachant que } \frac{p_1}{q_1} + \frac{p_2}{q_2} = \frac{p_1 q_2 + p_2 q_1}{q_1 q_2}$$

On suppose que b_1 et b_2 sont deux entiers non nuls.

Exercice 10

Affichez le résultat d'une permutation circulaire de droite à gauche de trois caractères donnés.

**Exercice 11**

Soit n un entier naturel donné, composé de trois chiffres. On demande de calculer puis d'afficher la somme des cubes des chiffres de n .

Exemple : pour $n = 253$

La somme des cubes de chiffres de $n = 2^3 + 5^3 + 3^3 = 160$

Exercice 12

Soient c et ch deux variables données tel que c est un caractère et ch est une chaîne de caractères. On vous demande d'afficher la deuxième position de c dans ch .

Exemple : Pour $c="m"$ et $ch="programmation"$, le programme affichera : 8

Pour $c="a"$ et $ch="programmation"$, le programme affichera : 0

Exercice 13

Déterminez puis affichez le nombre de chiffres d'un entier donné.

Exemple : Pour 2006, le programme affichera 4.

Exercice 14

Soient $chif$ et n deux variables données tel que $chif$ est un chiffre décimal et n un entier strictement positif. On vous demande d'afficher la deuxième position de $chif$ dans n .

Exemple : Pour $chif = 0$ et $n = 2006$, le programme affichera : 3

Pour $chif = 0$ et $n = 3764$, le programme affichera : 0

Pour $chif = 0$ et $n = 30764$, le programme affichera : 0



Chapitre 8

Les structures de contrôle conditionnelles

Objectifs :

- Résoudre des problèmes faisant appel aux structures de contrôle conditionnelles.

- Présenter les solutions sous forme d'un algorithme puis un programme.

Plan du chapitre :

Leçon 1 :
Formes simples ou alternatives

Leçon 2 :
Forme généralisée



Leçon 1

Formes simples ou alternatives

Objectifs spécifiques :

- .Résoudre des problèmes faisant appel aux structures de contrôle conditionnelles
- .Présenter les solutions sous forme d'un algorithme puis d'un programme

Plan de la leçon

- I. Introduction
- II. Forme simple
 - 1. Forme simple réduite
 - 2. Forme alternative ou forme complète

Retenons

Exercices

Leçon 1

Formes simples ou alternatives

I. Introduction

Un algorithme peut résoudre des problèmes très diversifiés où une décision devra être prise selon ce qu'on aura comme données. En effet, on est amené souvent à faire des choix, des décisions.

Par exemple, au moment de l'orientation, l'élève dira :

- si je fais option sciences alors je ferai des études de médecine.
- si je fais option informatique alors je ferai des études d'ingénieur.
- etc.

Notez bien ce premier cas : Si condition alors traitement

Comme il pourra raisonner de la manière suivante :

- Si j'ai une moyenne supérieure ou égale à 10 alors je réussis, sinon je risque de redoubler!

Notez bien ce deuxième cas : Si condition alors traitement 1 sinon traitement 2

Dans cette leçon, nous expliquerons comment écrire des algorithmes répondant à ce type de questions.

Définition

La structure de contrôle conditionnelle permet à un programme de réaliser un traitement en fonction d'une condition.

Il existe deux formes d'instructions conditionnelles :

- * Forme simple,
- * Forme généralisée (voir leçon 2).

II. Forme simple

II.1. Forme simple réduite

Une structure de contrôle conditionnelle a une forme simple réduite si on exécute un traitement donné quand la condition est évaluée à **VRAI**.

Vocabulaire et syntaxe

Au niveau de l'analyse

Cas d'une suite d'instructions :

R = [Initialisation]

Si condition Alors

Instruction 1

Instruction 2

- - - -

- - - -

Instruction N

} Ensemble d'instructions du traitement en question.

FinSi

NB :

[Initialisation] est une séquence d'instructions qui contiendra les éventuelles initialisations.

- Lorsque l'évaluation de la condition produit la valeur :
 - VRAI** les instructions entre Alors et FinSi sont exécutées.
 - FAUX** ces instructions ne sont pas exécutées.

Au niveau de l'algorithme

- - -

{ Initialisation }

Si condition Alors

Instruction 1

Instruction 2

- - - -

- - - -

Instruction N

FinSi

- - -

Au niveau du langage Pascal

Les instructions à traiter sont délimitées par les mots réservés "**BEGIN**" et "**END**".

- - - ;

{ Initialisation }

IF condition **THEN**

BEGIN

Instruction_1 ;

Instruction_2 ;

- - - - ;

- - - - ;

Instruction_N ;

END;

- - - ;

- - - ;

Activité 1

Écrivez une analyse, un algorithme et sa traduction en Pascal du programme nommé MIN_MAX qui détermine le minimum et le maximum de deux entiers saisis au clavier.

Réponse :
Analyse

| Nom = MIN_MAX | | |
|---------------|--|----------------------|
| S | L. D. E. | O.U. |
| 4 | Résultat = Ecrire (Min,Max) | |
| 3 | (Min, Max) = [Min← m, Max← n] Si (Max < Min) Alors Min ← n Max← m FinSi | Min Max m n |
| 1 | m = Donnée ("Entrer un premier entier : ") | |
| 2 | n = Donnée ("Entrer un deuxième entier : ") | |
| 5 | Fin MIN_MAX | |

Tableau de déclaration des objets

| Objet | Type / nature | Rôle |
|-------|---------------|----------------------------------|
| Min | Entier | sauvegarde du plus petit entier. |
| Max | Entier | sauvegarde du plus grand entier. |
| m | Entier | saisie du premier entier. |
| n | Entier | saisie du second entier. |

Algorithme

- 0) Début MIN_MAX
- 1) Écrire ("Entrer un premier entier : ") , Lire (m)
- 2) Écrire ("Entrer un deuxième entier : ") , Lire (n)
- 3) Min \leftarrow m
 Max \leftarrow n
 Si (Max < Min) Alors
 Min \leftarrow n
 Max \leftarrow m
 FinSi
- 4) Écrire (Min, Max)
- 5) Fin MIN_MAX

Traduction en Pascal

```

PROGRAM MIN_MAX;
USES WINCRT;

VAR      m, n, Min, Max : INTEGER;

BEGIN
  WRITE('Entrer un premier entier : '); READLN(m);
  WRITE('Entrer un deuxième entier : '); READLN(n);
  Min := m;
  Max := n;
  IF (Max < Min) THEN
    BEGIN
      Min := n;
      Max := m;
    END;

  WRITELN(Min,Max);
END.

```

II. 2 Forme alternative ou forme complète

Une structure de contrôle conditionnelle a une forme complète ou encore alternative si selon l'évaluation d'une condition, on exécute exclusivement soit un traitement 1 soit un traitement 2.

Vocabulaire et syntaxe

Forme générale :

[Initialisation]

Si condition **Alors**

```

Instruction 1
  ---
  ---
Instruction 1-n } Traitement 1
Sinon
Instruction 2-1 } Traitement 2
  ---
  ---
Instruction 2-m
    
```

FinSi

Lorsque l'évaluation de la condition produit la valeur :

VRAI seules les instructions de Traitement 1 sont exécutées.

FAUX seules les instructions de Traitement 2 sont exécutées.

Exemples :

Considérons le cas où traitement 1 et traitement 2 sont composés de plusieurs instructions.

Au niveau de l'analyse

[Initialisation] **Si** condition **Alors**

```

Instruction 1-1 de TR1;
Instruction 1-2 de TR1;
- - - -;
Instruction 1-n de TR1;
    
```

Sinon

```

Instruction 2-1 de TR2;
Instruction 2-2 de TR2;
- - - -
Instruction 2-m de TR2;
    
```

FinSi

N.B. : Les instructions de Traitement 1 sont délimitées entre Alors et Sinon et celles du traitement 2 le sont entre Sinon et FinSi.

Au niveau de l'algorithme

- - -

```

[ Initialisation ] Si condition Alors
                    Instruction 1 de TR1;
                    Instruction 2 de TR1;
                    - - - -
                    Instruction N de TR;

                    Sinon
                    Instruction 1 de TR2;
                    Instruction 2 de TR2;
                    - - - -
                    Instruction M de TR2;

                    FinSi

```

Au niveau du langage Pascal

Les instructions à traiter sont délimitées par les mots réservés "BEGIN" et "END".

- - -

```

{ Initialisation }
IF condition
THEN BEGIN
    Instruction 1-1;
    Instruction 1-2;
    - - - - ;
    Instruction 1-n;
    END (VOIR REMARQUE)
ELSE BEGIN
    Instruction 2-1;
    Instruction 2-2;
    - - - - ;
    Instruction 2-m;
END;

```

Remarque

Notez l'absence du ";" après le END qui précède le ELSE car l'instruction IF n'est pas encore finie. Par ailleurs, le ELSE ne doit être jamais précédé par ";" en Pascal.

Activité 2

Ecrivez une analyse, un algorithme et la traduction en Pascal du programme intitulé

DECISION qui, suivant la moyenne de l'élève, le résultat affiché sera succès au cas où l'élève a la moyenne ou échec sinon .

Réponse :

Analyse

| Nom : DECISION | | |
|----------------|---|--------------|
| S | L.D.E. | O.U. |
| 3 | Résultat = Ecrire (R) | R moyenne |
| 2 | R = [] Si (moyenne >= 10) Alors R ← "Succès" Sinon R ← "Echec" FinSi | |
| 1 | moyenne = Donnée(" Donner la moyenne de l'élève : ") | |
| 4 | Fin DECISION | |

Tableau de déclaration des objets

| Objet | Type / nature | Rôle |
|--------------|------------------------------|--|
| moyenne R | Réel Chaîne de caractères | saisie de la moyenne de l'élève. sauvegarde de la décision. |

Algorithme

- 0) Début DECISION
- 1) Ecrire (" Donner la moyenne de l'élève : ") Lire (moyenne)
- 2) Si (moyenne >= 10) Alors
 R ← "Succès"
 Sinon
 R ← "Echec"
 FinSi
- 3) Ecrire (R)
- 4) Fin DECISION

Traduction en langage Pascal

```

PROGRAM DECISION ;
USES WINCRT;
VAR
    moyenne : REAL ;
    R : STRING ;

BEGIN
    WRITE ('Donner la moyenne de l'élève :');
    READLN (moyenne) ;
    IF (moyenne >= 10)
        THEN BEGIN R := 'Succès';
        ELSE BEGIN R := 'Echec';END;
    WRITELN (R) ;
END.

```

Retenons

La forme alternative est utilisée pour définir un résultat dépendant d'une seule condition.

Exercices**Exercice 1**

On se propose d'écrire un programme qui calcule le salaire mensuel brut d'un ouvrier, en fonction du nombre total d'heures réalisées, du salaire horaire et de sa prime.

$\text{Salaire_SP} = \text{nombre_heures} * \text{salaire_horaire}$

$\text{Salaire_Brut} = \text{Salaire_SP} + \text{prime}$ (Salaire_SP signifie salarié sans prime)

Effectuez une analyse et écrivez l'algorithme. Traduisez ce dernier en langage Pascal.

Exercice 2

Écrivez un algorithme intitulé MIN_MAX permettant d'afficher le minimum et le maximum de trois entiers saisis au clavier.

Exercice 3

COLIS POSTAL

Les frais d'envoi d'un colis postal à l'intérieur du pays sont calculés comme suit :

- 3,600 dinars si le colis pèse moins de 2 kilogrammes.

- 2,150 dinars à chaque kilogramme ou partie de kilogramme en plus de 2 kilogrammes.

Effectuez une analyse du problème et écrivez un algorithme intitulé COLIS, permettant de saisir le poids d'un colis et d'afficher les frais d'envoi. Traduisez cet algorithme en langage Pascal.

Exercice 4

Écrivez un algorithme DECOMPOSITION qui permet de décomposer d'une manière optimale une somme d'argent en billets et pièces. La somme donnée est comprise entre zéro millime et 35 Dinars. Si la somme est strictement inférieure à 10 Dinars, on n'affichera que les pièces.

Leçon 2

Formes généralisées

Objectifs spécifiques :

- Résoudre des problèmes faisant appel à la forme généralisée du schéma conditionnel.
- Présenter les solutions sous forme d'un algorithme puis un programme.

Plan de la leçon

- I. Introduction
- II. Vocabulaire et syntaxe

Retenons
Exercices

Leçon 2

Forme généralisée

I. Introduction

Dans la leçon précédente nous avons traité la forme simple de la structure conditionnelle. Dans cette forme, la condition conduit à deux alternatives uniquement : le cas où la condition est satisfaite et le cas échéant.

La forme généralisée de la structure conditionnelle correspond à la situation dans laquelle plusieurs alternatives sont envisageables. Ainsi une discussion est à prévoir au niveau de la condition.

Activité 1

On désire associer à la moyenne des notes (moy) d'un élève, une appréciation (app) adéquate :

| Moyenne (moy) | Appréciation (app) |
|---------------------------|--------------------|
| $20 \geq \text{moy} > 18$ | Excellent |
| $18 \geq \text{moy} > 16$ | Très Bien |
| $16 \geq \text{moy} > 14$ | Bien |
| $14 \geq \text{moy} > 12$ | Assez Bien |
| $12 \geq \text{moy} > 10$ | Passable |
| $10 \geq \text{moy} > 0$ | Faible |

Nous constatons qu'une et une seule appréciation est associée à chaque moyenne.

II. Vocabulaire et syntaxe

Au niveau de l'analyse

Forme générale :

```
[Initialisation] Si condition 1 Alors Traitement 1
                  Sinon Si condition 2 Alors Traitement 2
                  Sinon Si condition 3 Alors Traitement 3
                  ----
                  Sinon Si condition n-1 Alors Traitement n-1
                  Sinon Traitement n
FinSi
```


Lorsque l'évaluation de la condition 1 produit la valeur :

VRAI, on exécute le traitement 1 puis on passe à l'instruction qui suit le Si généralisé.

FAUX on passe à l'évaluation de la condition 2, si elle produit la valeur :

VRAI, alors on exécute le traitement 2 puis on passe à l'instruction qui suit le Si généralisé.

FAUX, on passe à l'évaluation de la condition 3 et ainsi de suite.

Si aucune des n-1 premières conditions ne produit la valeur VRAI, c'est le traitement N qui sera exécuté.

Au niveau de l'algorithme

```

---
{Initialisation }
Si condition 1 Alors traitement 1
  Sinon Si condition 2 Alors traitement 2
  Sinon Si condition 3 Alors traitement 3
  ----
  ----
  Sinon Si condition N-1 Alors traitement N-1
  Sinon traitement N
FinSi

```

Au niveau du langage Pascal

```

- - ;
{ Initialisation }
IF condition_1 THEN traitement_1
ELSE IF condition_2 THEN traitement_2
ELSE IF condition_3 THEN traitement_3
- - -
- - -
ELSE IF condition_N-1 THEN traitement_N-1
ELSE traitement_N ;
- - ;

```


Algorithme

- 0) Début GENERALISEE
- 1) Écrire (" Donner un entier : ") , Lire (m)
- 2) Écrire (" Donner un autre entier : ") , Lire (n)
- 3) Si (m > n) Alors sig ← " est supérieur à "
 - Sinon Si (m < n) Alors sig ← " est inférieur à "
 - Sinon sig ← " est égal à "
- FinSi
- 4) Écrire (m, sig, n)
- 5) FIN GENERALISEE

Traduction en langage Pascal

```

PROGRAM GENERALISEE ;
USES WINCRT;
VAR
m, n : INTEGER ;
sig : STRING ;

BEGIN
WRITE('Donner un entier '); READLN(m);
WRITE('Donner un autre entier '); READLN(n);
IF ( m > n ) THEN
    sig := ' est supérieur à '
ELSE IF ( m < n ) THEN
    sig := ' est inférieur à '
ELSE
    sig := ' est égal à ' ;
WRITELN ( m, sig, n ) ;
END.

```

Activité 3

Reprenons l'Activité 1 en détail. Il s'agit de :

- Saisir les trois notes (note1, note2 et note3) des trois matières dont les coefficients sont successivement coef1 = 3, coef2 = 4 et coef3 = 2.
- Calculer la moyenne pondérée des trois notes.
- Déterminer l'appréciation adéquate (voir tableau de l'activité 1).
- Afficher la moyenne suivie de l'appréciation.

Questions :

- 1) Analysez le problème nommé BULLETIN.
- 2) Ecrivez l'algorithme correspondant.
- 3) Traduisez cet algorithme en Pascal.

Réponse :

1) Analyse

| Nom : BULLETIN | | |
|----------------|--|---|
| S | L.D.E | O.U. |
| 6 | Résultat = Ecrire (moy, décision) | |
| 4 | $moy \leftarrow ((coef1*note1) + (coef2*note2) + (coef3*note3)) / (coef1 + coef2 + coef3)$ | |
| 5 | décision = [] Si (moy > 18) alors décision ← " Excellent " Sinon Si (moy > 16) Alors décision ← " Très Bien " Sinon Si (moy > 14) Alors décision ← " Bien " Sinon Si (moy > 12) Alors décision ← " Assez Bien " Sinon Si moy >= 10 Alors décision ← " Passable " Sinon décision ← " Faible " FinSi | moy décision note 1 note 2 note 3 coef 1 coef 2 coef 3 |
| 1 | note 1 = Donnée (" Donner la première note : ") | |
| 2 | note 2 = Donnée (" Donner la deuxième note : ") | |
| 3 | note 3 = Donnée (" Donner la troisième note : ") | |
| | coef 1 : constante = 3 | |
| | coef 2 : constante = 4 | |
| | coef 3 : constante = 2 | |
| 7 | Fin BULLETIN | |

Tableau de déclaration des objets

| Objet | Type / nature | Rôle |
|----------|----------------------|-------------------------------------|
| moy | Réel | calcul de la moyenne. |
| décision | Chaîne de caractères | sauvegarde de la décision. |
| note 1 | Réel | saisie de la première note. |
| note 2 | Réel | saisie de la deuxième note. |
| note 3 | Réel | saisie de la troisième note. |
| coef 1 | Constante = 3 | Coefficient de la première matière |
| coef 2 | Constante = 4 | Coefficient de la deuxième matière |
| coef 3 | Constante = 2 | Coefficient de la troisième matière |

2) Algorithme

- 0) Début BULLETIN
- 1) Ecrire (" Donner la première note : ") Lire (NOTE1)
- 2) Ecrire (" Donner la deuxième note : ") Lire (NOTE2)
- 3) Ecrire (" Donner la troisième note : ") Lire (NOTE3)
- 4) $\text{moy} \leftarrow ((\text{COEF1} * \text{NOTE1}) + (\text{COEF2} * \text{NOTE2}) + (\text{COEF3} * \text{NOTE3})) / (\text{COEF1} + \text{COEF2} + \text{COEF3})$
- 5) Si (moy > 18) Alors
 - décision ← " Excellent "
 - Sinon Si (moy > 16) Alors
 - décision ← " Très Bien "
 - Sinon Si (moy > 14) Alors
 - décision ← " Bien "
 - Sinon Si (moy > 12) Alors
 - décision ← " Assez Bien "
 - Sinon Si (moy > 10) Alors
 - décision ← " Passable "
 - Sinon décision ← " Faible "
- 6) Ecrire (moy, décision)
- 7) Fin BULLETIN

NB : Si le module est réduit à une seule instruction, on pourra se passer des délimitations BEGIN et END au niveau du programme Pascal.

3) Traduction en Pascal

```
PROGRAM BULLETIN ;
USES WINCRT;
CONST
  COEF1=3; COEF2=4; COEF3=2;
VAR
  moy, NOTE1, NOTE2, NOTE3 ; REAL;
  decision : STRING;
BEGIN
  WRITE('Donner la première note : '); READLN(NOTE1);
  WRITE('Donner la deuxième note : '); READLN(NOTE2);
  WRITE('Donner la troisième note : '); READLN(NOTE3);
  moy:= ((COEF1*NOTE1)+(COEF2*NOTE2)+(COEF3*NOTE3)) /
        (COEF1+COEF2+COEF3);
  IF moy >18 THEN decision := 'Excellent'
    ELSE IF moy >16 THEN decision := 'Très Bien'
    ELSE IF moy >14 THEN decision := 'Bien'
    ELSE IF moy >12 THEN decision := 'Assez Bien'
    ELSE IF moy >=10 THEN decision := 'Passable'
    ELSE decision := 'Faible';
  WRITELN(moy, decision);
END.
```

Retenons

On utilise la structure conditionnelle généralisée pour définir un résultat astreint à plusieurs conditions dont chacune correspond à un traitement donné.

Exercices**Exercice 1**

Effectuez une analyse puis un algorithme qui permet de déterminer la saison à partir de la donnée d'un mois. Par exemple : à la donnée du mois 6, l'algorithme devrait afficher « Été ».

Traduisez cet algorithme en Pascal

Exercice 2

Effectuez une analyse puis écrivez un algorithme intitulé DEGRE 1, qui résout dans IR une équation du premier degré à une inconnue x réelle de la forme : $ax + b = 0$

Traduisez cet algorithme en Pascal

Exercice 3

Effectuez une analyse puis écrivez un algorithme intitulé CLASSEMENT_NBRE, qui lit trois entiers, les permute de façon à les classer par ordre croissant puis affiche le résultat.

Traduisez cet algorithme en Pascal

Exercice 4

Effectuez une analyse puis écrivez un algorithme et la traduction en Pascal du programme intitulé T_LETTRES, qui traduit en toutes lettres un entier naturel composé d'au maximum deux chiffres.

Exemples : 21 donne "VINGT ET UN"

22 donne "VINGT DEUX"

Exercice 5

Effectuez une analyse puis écrivez un algorithme permettant de saisir le sexe (M/F), la taille (cm), le poids (kg) d'une personne et d'afficher :

- PI, le poids idéal d'une personne, sachant que ce poids théorique est donné par la formule de Lorenz comme suit :

Pour un homme : $PI = (taille - 100) - (taille - 150) / 4$

Pour une femme : $PI = (taille - 100) - (taille - 120) / 4$

- BMI, l'indicateur d'obésité (Body Mass Index) où $BMI = poids / taille^2$ avec taille en mètre.

- Si une personne est considérée comme : Normale ($BMI \leq 27$), ou obèse ($BMI > 27$) ou Malade ($BMI \geq 32$).

Traduisez l'algorithme en Pascal.

Chapitre 9

Les structures de contrôle itératives

Objectifs :

- Utiliser la structure itérative complète pour résoudre des problèmes
- Résoudre des problèmes récurrents

Plan du chapitre :

Leçon :
Structures itératives complètes

Leçon

Structures de contrôle itératives

Objectifs spécifiques :

- . Utiliser la structure itérative complète pour résoudre des problèmes.
- . Résoudre des problèmes récurrents.

Plan de la leçon

I- Introduction et définition

1. Définition
2. Vocabulaire et syntaxe
3. traduction en Pascal

II- Traitement utilisant le compteur

III- Problèmes récurrents

Retenons

Exercices

Leçon

Les structures de contrôle itératives complètes

I- Introduction et définition

Certains résultats d'un programme donné nécessitent la répétition d'un traitement. Parfois, le résultat lui-même se construit par la répétition d'une action ou d'un traitement donnés. Prenons les deux exemples suivants :

Calculer les moyennes de n élèves d'une classe donnée. Un tel travail consiste à répéter n fois le même traitement avec des données différentes qui sont les notes du j^{ème} élève.

Déterminer l'état d'un compte après n opérations. Le solde du compte avant la première opération est de m dinars. Le résultat final du compte est trouvé après les n additions (ou soustractions) à faire sur le solde du compte. Dans ce dernier cas, on s'intéresse au dernier résultat de l'itération. On dit qu'on a un problème récurrent.

Dans chacun des cas, il s'agit de la répétition d'un traitement bien déterminé un nombre fini de fois.

Cette répétition est appelée structure de contrôle itérative complète.

I.1 Définition

Un résultat a une structure itérative complète s'il est obtenu après la répétition d'un traitement un nombre fini de fois connu d'avance.

Comment traduire une telle définition au niveau de l'analyse et de l'algorithme ?

I.2 Vocabulaire et syntaxe

Pour signifier la répétition, on utilise un compteur c.

```
R = [Initialisation] Pour c de 1 à n faire
    Traitement
FinPour
```

Traitement est une suite d'instructions écrites dans l'ordre d'exécution (penser à une approche ascendante)

On pourra formuler autrement le résultat R par :

```
R = [Initialisation] Pour c de 1 à n faire
      Instruction 1
      Instruction 2
      .
      .
      Instruction n
FinPour
```

L'initialisation qui est devant la structure Pour, comportera les définitions nécessaires au fonctionnement du processus itératif. Elle contient généralement les définitions constantes. Elle est souvent réduite à quelques instructions et remplie en dernier lieu après avoir défini l'itération.

Ce qui donne à la fin :

```
R = [Inst1, Inst2, ...,Instm] Pour c de 1 à n faire
      Instruction 1
      Instruction 2
      .
      .
      Instruction n
FinPour
```

Le compteur c devra être de type scalaire (entier, caractères, booléen, etc.). Chaque élément devra avoir un successeur dans le cas où le parcours est ascendant et un prédécesseur dans le cas où le parcours est descendant. L'incrément de c de la valeur en cours à la valeur suivante (successeur) est automatique. Dans le cas précédent, c est visiblement un entier, donc les valeurs successives de c sont 1, 2, 3, ..., jusqu'à la dernière valeur n. Dès que la valeur de c arrive à la dernière valeur n, l'opération d'itération s'arrête.

Dans le cas d'un parcours descendant, la définition s'écrit :

```
R = [Inst1, Inst2, ...,Instm] Pour c de n à 1 (pas = -1) Faire
      Instruction 1
      Instruction 2
      .
      .
      Instruction n
FinPour
```

I.3 Traduction en Pascal

```

PROGRAM nom_programme;
USES WINCRT;
VAR
.
.
BEGIN
.
.
  {Instructions d'initialization}
  FOR i:=1 TO n DO
    BEGIN
      .
      Instructions du traitement à répéter
      .
    END;
END.

```

Attention

Dans le cas où le parcours est descendant, on utilise l'instruction DOWNTO comme suit

```

{Instructions d'initialisation}
FOR i := n DOWNTO 1 DO
  BEGIN
    .
    Instructions du traitement à répéter
    .
  END;

```

Activité 1

Écrivez un programme qui calcule et affiche les moyennes annuelles de n élèves. La moyenne annuelle est égale à :

$$(\text{Moy_Trim1} + 2 \cdot \text{Moy_Trim2} + 2 \cdot \text{Moy_Trim3})/5$$

Moy_Trim1 étant la moyenne du trimestre 1.

Moy_Trim2 étant la moyenne du trimestre 2.

Moy_Trim3 étant la moyenne du trimestre 3.

Réponse : Nous établissons alors l'analyse suivante :

| NOM : Moyennes_annuelles | | |
|--------------------------|--|---|
| S | L.D.E. | O.U. |
| 1 | Résultat = Moy_annuelles Moy_annuelles = [Lire(n)] Pour i de 1 à n faire Moy_Trim1 = Donnée Moy_Trim2 = Donnée Moy_Trim3 = Donnée Moy_annuelle ← (Moy_Trim1 + 2 · Moy_Trim2 + 2 · Moy_Trim3)/5 Ecrire (Moy_annuelle) FinPour | n i Moy_Trim1 Moy_Trim2 Moy_Trim3 Moy_annuelle |
| 2 | i = compteur Fin Moyennes_annuelles | |

Remarques

✍ Le résultat "Moy_annuelles" a une définition itérative. Comme on connaît d'avance le nombre d'itérations. Cette définition itérative est donc complète.

✍ Après avoir établi le corps de l'itération, on a remarqué qu'elle ne fonctionnera pas sans avoir défini n . Par conséquent, nous avons mis la lecture de n dans l'initialisation de l'itération. Nous rappelons que cette partie initialise le procédé itératif qui suit. Il faudra donc définir l'itération en premier lieu puis procéder à son initialisation. Dans une itération complète, cette partie d'initialisation devrait comporter tous les objets qui sont constants ou encore indépendants du compteur.

✍ Les instructions parenthésées ou se trouvant entre Pour et FinPour constitue le corps du traitement à répéter.

Tableau de déclaration des objets.

| Objet | Nature/Type | Rôle |
|--------------|-------------|------------------------------------|
| n | Entier | Nombre d'itérations |
| i | Entier | Compteur |
| Moy_Trim1 | Réel | Contient la moyenne du trimestre 1 |
| Moy_Trim1 | Réel | Contient la moyenne du trimestre 2 |
| Moy_Trim1 | Réel | Contient la moyenne du trimestre 3 |
| Moy_annuelle | Réel | Contient la moyenne annuelle |

Algorithme

- 0) Début Moyennes_annuelles
- 1) [Lire(n)] Pour i de 1 à n faire
 - Moy_Trim1 = Donnée
 - Moy_Trim2 = Donnée
 - Moy_Trim3 = Donnée
 - $Moy_annuelle \leftarrow (Moy_Trim1 + 2 * Moy_Trim2 + 2 * Moy_Trim3) / 5$
 - Ecrire (Moy_annuelle)
- FinPour
- 2) Fin Moyennes_annuelles

Traduction en Turbo Pascal

```

PROGRAM Moyennes_annuelles;
USES WINCRT;
VAR   i, n : INTEGER;
      Moy_Trim1, Moy_Trim2, Moy_Trim3 : REAL;
      Moy_annuelle : REAL;
BEGIN
  READLN(n);
  FOR i := 1 TO n DO
    BEGIN
      READLN(Moy_Trim1);
      READLN(Moy_Trim2);
      READLN(Moy_Trim3);
      Moy_annuelle := (Moy_Trim1 + 2 * Moy_Trim2 + 2 * Moy_Trim3) / 5;
      WRITELN(Moy_annuelle);
    END;
  END;
END.

```

II. Traitement utilisant le compteur

Dans une itération, Il est possible d'exploiter le compteur dans le traitement à répéter.

Activité 2

Écrivez un programme qui saisit un entier naturel non nul n suivi de n réels à mettre dans un tableau A. Dans chaque saisie, on indiquera le rang correspondant dans un message.

Solution :

| Nom : Saisie_tableau | | |
|----------------------|---|-------------|
| S | L.D.E. | O.U. |
| 1 | Résultat = A A = [Lire(n)] Pour i de 1 à n Faire Écrire ("Saisie de l'élément n° ", i) Lire(A [i]) FinPour | A n i |
| 2 | Fin Saisie_tableau | |

Tableau de déclaration des objets

| Objet | Nature/Type | Rôle |
|-------|----------------------|---------------------------------|
| n | Entier | Le nombre d'éléments du tableau |
| i | Entier | compteur |
| A | Vecteur de 100 réels | contient les n réels |

Nous en déduisons l'algorithme suivant :

- 0) Début Saisie_tableau
- 1) [Lire (n)] Pour i de 1 à n répéter
 - Ecrire ("Saisie de l'élément n° ", i)
 - Lire (A[i])
 - FinPour
- 2) Fin Saisie_tableau

Traduction en Pascal

```
PROGRAM saisie_tableau ;
USES WINCRT;
VAR i, n :INTEGER;
    A : ARRAY[1..100] OF REAL;

BEGIN
    READLN(n);
    FOR i:=1 TO n DO
        BEGIN
            WRITE('Saisie de l''élément n°', i);
            READLN(A[i]);
        END;
    END.
```

Activité 3

Écrivez un programme qui saisit un entier n suivi de n noms à mettre dans un tableau MN. Il cherche à partir de la fin du tableau, tous les noms commençant par "R" et les affiche. (On suppose que la saisie est en majuscule).

Analyse

Nous remarquons que la recherche des mots commençant par "R" nécessite un parcours descendant du tableau. A chaque fois, nous vérifions si le premier caractère de l'élément du tableau en cours est égal à "R". Dans l'affirmative, on l'affiche et nous continuons le parcours jusqu'au premier élément du tableau.

| Nom : recherche_mots | | |
|----------------------|--|---------|
| S | L.D.E. | O.U. |
| 2 | Résultat = mots_R Mots_R = [] Pour i de n à 1 (pas=-1) faire [] Si (MN [i] [1] = "R") alors Écrire(MN [i]) FinSi | n i |
| 1 | FinPour MN = [Lire ("n = ",n)] Pour j de 1 à n Faire Lire(MN [j]) | MN j |
| 3 | FinPour Fin recherche_mots | |

Tableau de déclaration des objets

| Objet | Nature/Type | Rôle |
|-------|----------------------------------|---------------------|
| n | Entier | Nombre d'itérations |
| j | Entier | Compteur |
| i | Entier | Compteur |
| MN | Vecteur de chaînes de caractères | Contient les noms |

Algorithme

- 0) Début recherche_mots
- 1) [Lire ("n = ",n)] Pour j de 1 à n Faire
 Lire(MN [j])
 FinPour
- 2) Pour i de n à 1 Faire
 Si (MN [i] [1] = "R") alors
 Écrire(MN [i])
 FinSi
 FinPour
- 3) Fin recherche_mots

Traduction en Pascal

```

PROGRAM recherche_mots;
USES WINCRT;
VAR   i, j, n : INTEGER;
      MN   : ARRAY [1..100] OF STRING;

BEGIN
  READLN ( n );
  FOR j := 1 TO n DO
    BEGIN
      READLN (MN [ j ] );
    END;
  FOR i := n DOWNTO 1 DO
    BEGIN
      IF MN [ i ] [1] = 'R' THEN
        BEGIN
          WRITELN ( MN [ i ] );
        END;
    END;
  END;
END.

```


III. Problèmes récurrents

On dit qu'on a une itération récurrente si on s'intéresse au résultat final de cette itération. Dans une telle situation, l'itération de rang i dépend d'une ou de plusieurs itérations de rangs précédents.

Activité 4

Les moyennes des élèves d'une classe sont saisies dans un tableau M unidimensionnel de n éléments. On se propose de calculer et d'afficher la moyenne de la classe.

Écrivez un programme qui saisit M , calcule et affiche cette moyenne.

Solution

Analyse du problème

| NOM : Moyenne_classe | | |
|----------------------|---|---------------------|
| S | L.D.E. | O.U. |
| 4 | Résultat = Ecrire (moy_classe) | moy_classe total |
| 3 | moy_classe \leftarrow total / n | |
| 2 | total = [total \leftarrow 0] Pour i de 1 à n Faire total \leftarrow total + M[i] FinPour | n i |
| 1 | M = [Lire (n)] Pour i de 1 à n Faire Lire(M [i]) FinPour | M |
| 5 | Fin Moyenne_classe | |

Tableau de déclaration des objets

| Objet | Nature/Type | Rôle |
|------------|----------------------|----------------------------------|
| n | Entier | Nombre d'éléments de M |
| i | Entier | Compteur |
| total | Réel | Contient le total des moyennes |
| moy_classe | Réel | La moyenne de classe |
| M | Vecteur de 100 réels | Contient les moyennes des élèves |

De l'analyse précédente, nous obtenons l'algorithme suivant :

Algorithme

- 0) Début moyenne_classe
- 1) [Lire (n)] Pour i de 1 à n Faire
Lire (M [i])
FinPour

- 2) [total ← 0] Pour i de 1 à n Faire
 - total ← total + M [i]
 - FinPour
- 3) moy_classe ← total / n
- 4) Ecrire (moy_classe)
- 5) Fin moyenne_classe

Remarques

- ✍ Dans la ligne 1, l'algorithme commence par la saisie de n puis on utilise une boucle pour remplir le tableau M.
- ✍ Dans la ligne 2, on initialise total à 0 puis avec une boucle complète (n fois) on ajoute à total le ième élément de M.
- ✍ La ligne 3 procède au calcul de la moyenne.
- ✍ La ligne 4 affiche le résultat escompté.

Traduction en Pascal

```

PROGRAM moyenne_classe;
USES WINCRT;
VAR
    i, n : INTEGER;
    moy_classe, total : REAL
    M : ARRAY[1..100] OF REAL;

BEGIN
    READLN (n);
    FOR i := 1 TO n DO
        BEGIN
            READLN ( M [ i ] );
        END;
    total := 0;
    FOR i := 1 TO n DO
        BEGIN
            total := total + (M [ i ]);
        END;
    moy_classe := total / n;
    WRITELN (moy_classe);
END.

```

Retenons

- ✍ Un résultat a une structure itérative complète s'il est déduit de la répétition d'un traitement un nombre fini de fois connu d'avance.
- ✍ Le compteur d'une itération est de type scalaire, il admet un successeur et un prédécesseur à part les éléments situés aux bords.
- ✍ Dans un parcours ascendant, le compteur s'incrémente automatiquement.
- ✍ Dans un parcours descendant, le compteur se décrémente automatiquement.
- ✍ L'initialisation d'une itération comporte les instructions utiles au fonctionnement de celle-ci. On la définit après avoir mis en place l'itération en question.

Exercices

Exercice 1

Écrivez un programme qui reproduit 20 lignes successives contenant chacune votre nom et prénom.

Exercice 2

Écrivez un programme qui affiche des étoiles sur chacune des deux colonnes gauche et droite de l'écran.

Exercice 3

Écrivez un programme qui dessine un triangle rectangle par des étoiles comme le montre la figure ci-dessous.

```
*
**
***
****
```

Exercice 4

Écrivez un programme qui dessine un triangle isocèle par des étoiles comme le montre la figure ci-dessous.

```
*
 ***
*****
*****
```

Exercice 5

Écrivez un programme qui dresse la table de multiplication des 12 premiers naturels non nuls par un entier naturel n compris entre 1 et 9.

L'affichage est de la forme suivante :

Pour $n = 5$, la table est :

```
1 x 5 = 5
2 x 5 = 10
3 x 5 = 15
etc.
12 x 5 = 60
```

Exercice 6

Écrivez un programme qui saisit un entier naturel n puis n réels à mettre dans un tableau V , ensuite il affiche à partir de la fin du tableau tous les éléments de rangs pairs.

Exercice 7

Écrivez un programme qui saisit un entier naturel non nul n suivi de n réels à placer dans un tableau A . Il remplit un deuxième tableau B par les éléments de A en commençant par le dernier. C'est-à-dire : $B[1]$ contiendra $A[n]$, $B[2]$ contiendra $A[n-1]$ et ainsi de suite.

Exercice 8

Écrivez un programme qui saisit un texte à mettre dans une variable chaîne ch puis cherche toutes les occurrences des lettres de l'alphabet et les met dans un tableau OC . Les indices des éléments de OC vont de "A" à "Z". On suppose que le texte n'utilise pas de caractères accentués. Le programme affiche ensuite chaque lettre suivie de ses occurrences dans le texte.

On rappelle que les occurrences d'une lettre signifient le nombre d'apparitions.

Exercice 9

Écrivez un programme qui saisit un texte en français et détermine les fréquences des voyelles et les met dans un tableau de 6 éléments. Le programme affiche ensuite la fréquence de chacune des voyelles.

Le texte pourra comporter des lettres en majuscules ou en minuscules ainsi que les caractères accentués.

Exercice 10

Écrivez un programme qui cherche tous les carrés parfaits de la forme $aabb$ (dans la base 10).

Exercice 11

Écrivez un programme qui cherche les carrés parfaits de 4 chiffres tels que le nombre formé par les deux chiffres à gauche et le nombre formé par les deux chiffres à droite sont eux aussi des carrés parfaits.

Exercice 12

Écrivez un programme qui calcule les moyennes annuelles de n élèves d'une classe et décide du passage ou du redoublement. La moyenne annuelle est égale à : $(\text{moyenne_trimestre1} + 2 \cdot \text{moyenne_trimestre2} + 2 \cdot \text{moyenne_trimestre3})/5$

Un élève passe à la classe supérieure si sa moyenne annuelle est supérieure ou égale à 10, autrement, il redouble.

Exercice 13

Écrivez un programme qui génère, au hasard, n ($n > 100$) paires de coordonnées x et y comprises entre -1 et 1 et les met dans deux tableaux X et Y . Il calcule ensuite le nombre de couples qui vérifient $x^2 + y^2 = 1$, le multiplier par 0.04 et l'afficher. Que constatez vous ?

On pourra utiliser la fonction `RANDOM` du langage Pascal

Exercice 14

Écrivez un programme qui saisit une chaîne de caractères, la renverse et la met dans une deuxième chaîne puis il affiche les deux sur deux lignes successives.

Exercice 15

On donne une somme d'argent S . Écrivez un programme qui l'exprime en un minimum de billets et de pièces.

Exercice 16

Écrivez un programme qui saisit un entier positif n et détermine s'il est ou non premier. Rappelons qu'un entier autre que 1 est premier s'il n'est divisible que par 1 et par lui-même.

Exercice 17

Écrivez un programme qui cherche et affiche les 100 premiers entiers naturels premiers.

Exercice 18

On veut afficher à l'écran le calendrier d'un mois donné. On donne le mois, l'année et le jour de la semaine correspondant au premier du mois en question.

Écrivez un programme qui réalise cette tâche.

Exemple :

Pour le mois de janvier 2007, on obtient :

| | | | | | |
|----------|---|----|----|----|----|
| LUNDI | 1 | 8 | 15 | 22 | 29 |
| MARDI | 2 | 9 | 16 | 23 | 30 |
| MERCREDI | 3 | 10 | 17 | 24 | 31 |
| JEUDI | 4 | 11 | 18 | 25 | |
| VENDREDI | 5 | 12 | 19 | 26 | |
| SAMEDI | 6 | 13 | 20 | 27 | |
| DIMANCHE | 7 | 14 | 21 | 28 | |

[BOR] Borland International Inc.

Turbo Pascal 4.0

[BOR] Borland International Inc.

Turbo Pascal 6.0 (Guide du programmeur)

[BOR] Borland International Inc.

Turbo Pascal 6.0 (Guide de référence)

[BOY] J. BOYER, J.-D. LAMOITIER, M. TREILLET.Systèmes PC.DOS et MS.DOS version 2 à 4
Editests. 1987**[CAR] R. CARRE, J.-F. DEGREMONT, M. GROSS, J.-M. PIERREL, G. SABAH**Langage humain et machine
Presses CNRS 1990**[CHA] G. CHATY, J. VICARD**L'algorithmique de la pratique à la théorie.
CEDIC. 1983**[ENG] A. ENGEL**Mathématique élémentaire d'un point de vue algorithmique.
CEDIC. 1981**[ESP] B. D'ESPAGNAT**Penser la science ou les enjeux du savoir
Dunod 1990**[KNU] D. KNUTH**The Art Of Computer Programming Vol.1, 2 et3.
ADDISON WESLEY . 1968
Vol. 3 : Sorting and Searching**[LAU] J.-P. LAURENT**Initiation à l'analyse et à la programmation.
DUNOD. 1985**[LIG] P. LIGNELET**Algorithmique. Tome 1 et Tome 2.
MASSON. 1981**[MEY] J.-J. MEYER**Pratique du Turbo Pascal (Créez vos progiciels).
Éditions Radio. 1987**[PAI] C. PAIR**Sciences et Pratiques de l'Informatique.
BORDAS. 1987**[ROY] D. ROY**Routines en Turbo Pascal.
Éditions P.S.I. 1989**[TRU] J.-P. TRUMBLAY, R. B. BUNT, P. G. SORENSON**Logique de programmation
Traduction de G. DUPUIS et L. VAN BUSKIRK
McGraw-Hill. 1985**[WIR] N. WIRTH**Introduction à la programmation systématique.
MASSON. 1986

TABLE DES CODES ASCII

| HEX | DEC | CAR | HEX | DEC | CAR | HEX | DEC | CAR | HEX | DEC | CAR |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | NUL | 20 | 32 | | 40 | 64 | @ | 60 | 96 | ` |
| 1 | 1 | SHQ | 21 | 33 | ! | 41 | 65 | A | 61 | 97 | a |
| 2 | 2 | STX | 22 | 33 | " | 42 | 66 | B | 62 | 98 | b |
| 3 | 3 | ETX | 23 | 34 | # | 43 | 67 | C | 63 | 99 | c |
| 4 | 4 | EOT | 24 | 35 | \$ | 44 | 68 | D | 64 | 100 | d |
| 5 | 5 | ENQ | 25 | 36 | % | 45 | 69 | E | 65 | 101 | e |
| 6 | 6 | ACK | 26 | 37 | & | 46 | 70 | F | 66 | 102 | f |
| 7 | 7 | BEL | 27 | 38 | ' | 47 | 71 | G | 67 | 103 | g |
| 8 | 8 | BS | 28 | 39 | (| 48 | 72 | H | 68 | 104 | h |
| 9 | 9 | HT | 29 | 40 |) | 49 | 73 | I | 69 | 105 | i |
| A | 10 | LF | 2A | 42 | * | 4A | 74 | J | 6A | 106 | j |
| B | 11 | VT | 2B | 43 | + | 4B | 75 | K | 6B | 107 | k |
| C | 12 | FF | 2C | 44 | , | 4C | 76 | L | 6C | 108 | l |
| D | 13 | CR | 2D | 45 | - | 4D | 77 | M | 6D | 109 | m |
| E | 14 | SO | 2E | 46 | . | 4E | 78 | N | 6E | 110 | n |
| F | 15 | SI | 2F | 47 | / | 4F | 79 | O | 6F | 111 | o |
| 10 | 16 | DLE | 30 | 48 | 0 | 50 | 80 | P | 70 | 112 | p |
| 11 | 17 | DC1 | 31 | 49 | 1 | 51 | 81 | Q | 71 | 113 | q |
| 12 | 18 | DC2 | 32 | 50 | 2 | 52 | 82 | R | 72 | 114 | r |
| 13 | 19 | DC3 | 33 | 51 | 3 | 53 | 83 | S | 73 | 115 | s |
| 14 | 20 | DC4 | 34 | 52 | 4 | 54 | 84 | T | 74 | 116 | t |
| 15 | 21 | NAK | 35 | 53 | 5 | 55 | 85 | U | 75 | 117 | u |
| 16 | 22 | SYN | 36 | 54 | 6 | 56 | 86 | V | 76 | 118 | v |
| 17 | 23 | ETB | 37 | 55 | 7 | 57 | 87 | W | 77 | 119 | w |
| 18 | 24 | CAN | 38 | 56 | 8 | 58 | 88 | X | 78 | 120 | x |
| 19 | 25 | EM | 39 | 57 | 9 | 59 | 89 | Y | 79 | 121 | y |
| 1A | 26 | SUB | 3A | 58 | : | 5A | 90 | Z | 7A | 122 | z |
| 1B | 27 | ESC | 3B | 59 | ; | 5B | 91 | [| 7B | 123 | |
| 1C | 28 | FS | 3C | 60 | < | 5C | 92 | / | 7C | 124 | |
| 1D | 29 | GS | 3D | 61 | = | 5D | 93 |] | 7D | 125 | } |
| 1E | 30 | RS | 3E | 62 | > | 5E | 94 | ^ | 7E | 126 | ~ |
| 1F | 31 | US | 3F | 63 | ? | 5F | 95 | _ | 7F | 127 | |

Les autres pages de codes ne sont pas standards, nous ne les reproduisons pas.