

INFORMATIQUE

2ème année de l'enseignement secondaire

Technologies de l'Informatique

Les auteurs

Rached DOUARI

Inspecteur

Abdelhafidh ABIDI

Inspecteur Principal

Dorsaf CHELLY BANNA

Professeur Principal

Les évaluateurs

Samir El AMRI

Inspecteur

Mohamed El OUADI

Docteur en Informatique

Préface

Ce livre, support officiel d'apprentissage pour les élèves de 2^{ème} année de la filière Technologie de l'informatique est conforme au programme officiel et se veut assurer la totalité des objectifs visés par ce curriculum. En effet, la matière informatique à ce niveau et particulièrement dans cette filière revêt une grande importance pour les apprenants. Les apprentissages dispensés dans ce programme devront permettre à l'élève de s'accrocher à cette discipline et d'arriver à faire clairement la bonne continuité à la fin de cette deuxième année. Les différentes parties du programme font un bon tour d'horizon pour former chez l'apprenant un savoir minimal lui permettant de résoudre divers types de problèmes soit en utilisant des logiciels et des applications déjà prêtes ou en confectionnant ses propres programmes. Au-delà de cet objectif de taille, l'élève acquièrera une culture informatique et comprendra le fonctionnement de l'ordinateur. Trois grandes parties forment le contenu de ce livre. La première partie d'ordre culturel est composée des chapitres relatifs à une présentation de l'informatique et de son évolution dans le temps. Nous passerons ensuite à l'architecture générale d'un ordinateur sans toute fois entrer dans des détails de fonctionnement assez difficile pour ce niveau. Toutes les nouveautés technologiques sont de mise, évolution oblige. Nous devons déterminer cette partie par un apprentissage d'un système d'exploitation pour permettre un démarrage sur et solide de la suite de l'apprentissage.

La deuxième partie sera consacrée à l'utilisation de l'ordinateur et d'un ensemble de logiciels pour résoudre des problèmes de conception et de réalisation de produits multimédias. L'élève sera en mesure de produire un document intégrant plusieurs médias. Il pourra dans ces conditions enrichir ses apprentissages en navigant sur le réseau Internet et chercher les informations voulues et les exploiter. Nous finirons par apprendre à l'élève de communiquer avec les autres et de pouvoir présenter ses idées et ses productions avec les applications appropriées. Nous aurons ainsi offert à l'élève les outils nécessaires lui permettant de résoudre des problèmes d'ordre bureautiques et surtout comment exploiter le numérique pour mieux communiquer avec les autres.

La dernière partie sera entièrement consacrée à l'algorithmique et à la programmation. Elle comportera les concepts de base que ce soit en matière de structuration de données ou de structuration des contrôles. Cet apprentissage d'algorithmique sera appuyé par un cours parallèle de programmation. L'approche de résolution d'un problème est la méthode descendante qui se prête bien à ce niveau d'apprentissage dans la mesure où un certain déterminisme doit être assuré. Ceci n'exclut pas l'utilisation de temps à autre de l'approche ascendante. Cette approche descendante sera consolidée à la fin de cette partie par une présentation de l'approche modulaire en l'occurrence les fonctions et les procédures. L'analyse du problème que nous avons appelée spécification est développée dans une notation proche du langage naturel. Nous mentionnerons aussi les différentes structures de données à adopter selon le problème posé. La traduction en programmes sera bénéfique pour l'apprenant. Il comprendra beaucoup mieux le fonctionnement des applications informatiques et se développera un savoir faire en matière de programmation. Notons que le langage de programmation abordée dans ce manuel est le Pascal. Ce langage permet une présentation pédagogique des solutions trouvées aux problèmes posés. Sa notoriété en tant que langage de programmation n'est pas à démontrer quand on sait que seul le Pascal et le C sont permis dans les Olympiades Internationales d'Informatique.

Nous remarquons avant de terminer que chaque chapitre est basée sur plusieurs activités aidant à comprendre les notions et les concepts du cours et est clôturé par une série d'exercices de difficulté progressive permettant à l'élève de s'exercer et de s'auto-évaluer. Nous souhaitons à nos lecteurs le meilleur profit de ce manuel et nous restons à leur disposition pour toutes les remarques et les suggestions.

Les auteurs.

Table des matières

Préface	3
Table des matières	4
Chapitre 1 Culture informatique	6
I. L'informatique : définition, historique et domaine d'utilisation	7
II. Notions d'information et de numérisation	12
III. Notion de logiciels	14
IV. Retenons	19
Exercices	20
Chapitre 2 Architecture d'un micro-ordinateur	22
I. Définition d'un ordinateur	23
II. Architecture de base d'un micro-ordinateur	24
III. Retenons	40
Exercices	41
Chapitre 3 Systèmes d'exploitation et réseaux informatiques	43
A) Système d'exploitation	44
I. Présentation et rôles	44
II. Les principales fonctions	46
III. Notions de fichiers et de répertoires	47
IV. Apprentissage des fonctions de base d'un système d'exploitation	49
B) les réseaux informatiques	60
I. Présentation	61
II. Les différents types	61
III. Les avantages d'un réseau	62
IV. Les logistiques matérielles et logicielles	63
V. Les topologies	64
VI. L'exploitation d'un réseau local	66
VII. Retenons	67
Exercices	68
Chapitre 4 Eléments de multimédia	69
I. Introduction	70
II. Le traitement de texte	71
III. L'image	84
IV. Le son	86
V. La vidéo	91
VI. Retenons	94
Exercices	95
Chapitre 5 Internet	96
I. Présentation	97
II. Les services d'Internet	100
III. Projet	107
IV. Retenons	108
Exercices	109

Chapitre 6 Éléments de présentation	115
I. La production de présentation	116
II. La production de pages Web	122
III. Retenons	129
Exercices	130
Chapitre 7 Introduction à la résolution de problèmes et à la programmation	131
I. Introduction	132
II. Spécification et analyse d'un problème	132
III. Ecriture d'un algorithme	135
IV. Traduction en un programme exécutable par ordinateur	138
V. Exécutions et tests	141
VI. Retenons	142
Exercices	143
Chapitre 8 Les structures de données	145
I. Les constantes	146
II. Les variables	147
III. Les types de données	148
IV. Retenons	160
Exercices	161
Chapitre 9 Les structures simples	164
I. Introduction	165
II. Les opérations d'entrée et de sortie	165
III. L'affectation	168
IV. Retenons	172
Exercices	173
Chapitre 10 Les structures de contrôle conditionnelles	175
I. Introduction	176
II. La structure conditionnelle simple	178
III. La structure conditionnelle généralisée	182
IV. La structure conditionnelle à choix	188
V. Retenons	193
Exercices	194
Chapitre 11 Les structures de contrôle itératives	196
I. Introduction	197
II. La structure itérative complète	198
III. La structure itérative à condition d'arrêt	205
IV. Retenons	216
Exercices	217
Chapitre 12 Les sous programmes	220
I. Introduction	221
II. Les fonctions	221
III. Les procédures	231
IV. Retenons	236
Exercices	237

CHAPITRE 1

Culture informatique

Objectifs

- Définir l'informatique et l'information
- Enumérer des domaines d'utilisation de l'informatique
- Reconnaître des dates clés de l'informatique
- Définir et énumérer les différents types de logiciels

Plan du chapitre

I. L'informatique : définition, historique et domaine d'utilisation

- I-1. Définition
- I-2. Les domaines informatiques
- I-3. Les dates clés de l'informatique

II. Notions d'information et de numérisation

- II-1. Définition
- II-2. Les unités de mesure

III. Notion de logiciels

- III-1. Définition
- III-2. Les différents types de logiciels
- III-3. La diffusion des logiciels
- III-4. La sécurité de l'information

IV. Retenons

Lecture

Exercices

CHAPITRE 1

Culture informatique

I. L'informatique : définition, historique et domaine d'utilisation

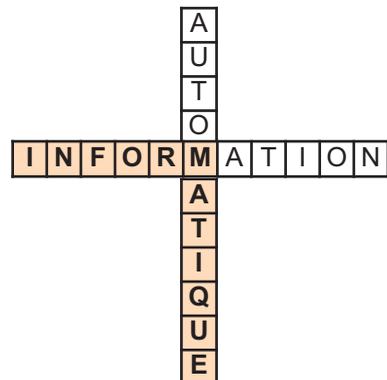
I.1. Définition

Activité 1

Consultez des dictionnaires et essayez de dégager la définition du terme " **informatique** ". Si vous n'en disposez pas, lisez attentivement les définitions attribuées au terme " **informatique** " telles qu'elles sont données dans les dictionnaires les plus connus.

Définition du terme " informatique "	D'après le 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 (logiciel) mis en oeuvre sur ordinateur."
	D'après l'Académie française	"L'informatique est la science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications dans les domaines technique, économique et social."
	D'après le dictionnaire de l'informatique et des sciences de l'information	"L'informatique est bien une science : la science du traitement de l'information elle ne se réduit pas à un recueil de recettes, de techniques et de savoir-faire, mais elle est au coeur de toute réflexion sur la modernité, car elle joue un rôle central dans de nombreuses disciplines scientifiques et dans la plupart des techniques contemporaines, son emprise sur notre environnement quotidien est tous les jours plus grande."

Le terme " **informatique** " désignant une discipline née avec l'**ordinateur**, est un néologisme français proposé par Philippe Dreyfus en 1962 condensant les mots *information* et *automatique*. Deux définitions de l'informatique se sont toujours heurtées : celle concernant l'ensemble des techniques mises en oeuvre pour l'utilisation des ordinateurs et celle concernant la science du traitement rationnel de l'information, notamment par des moyens automatiques.



I.2. Les domaines informatiques

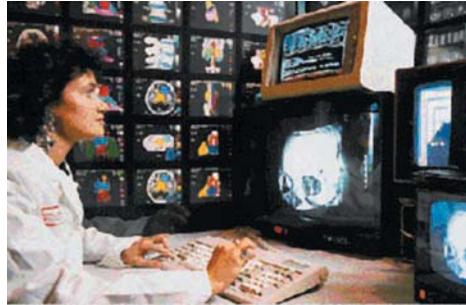


Fig I.1 : Images illustrant des domaines d'application de l'informatique

Domaines ou contextes de l'utilisation de l'ordinateur
La Recherche scientifique
L'industrie
L'échange de données
L'éducation
La médecine
Le développement d'applications et de logiciels

Activité 2

Aidé par votre professeur et en se référant à la figure **Fig I.1**, créez un tableau où vous indiquez les services informatiques correspondants à quelques domaines d'activité.

L'informatique est au centre de différents secteurs. C'est un outil qui investit toutes les professions. Cette révolution technologique est en train de transformer progressivement les méthodes usuelles et continue à provoquer une diversification du quotidien vécu (**Voir Tableau I.1**) et une innovation dans le marché de l'emploi.

Domaines d'activités habituelles	Domaine d'activité + Informatique = nouveau secteur généré
Travail de bureau	Bureautique
Télécommunication	Télématique
Banque	Monétique
Agriculture	Agrotique
Production industrielle	Productique
Habitation	Domotique

Tableau I.1 : Les nouveaux secteurs

I.3. Les dates clés de l'informatique

Activité 3

- 1/ Utiliser le schéma de la (Fig.I.2) pour découvrir les dates clés de l'informatique.
- 2/ Dédire dans un tableau les différentes générations d'ordinateurs en insistant sur l'évolution matérielle et logicielle.

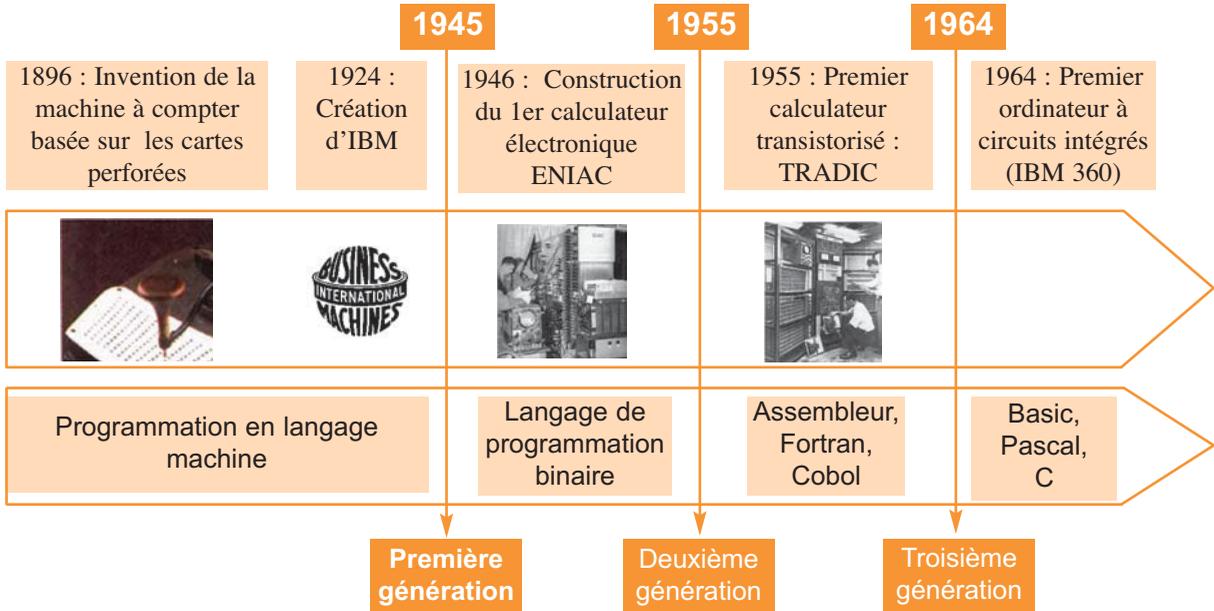


Fig I.2 : Dates clés de l'informatique

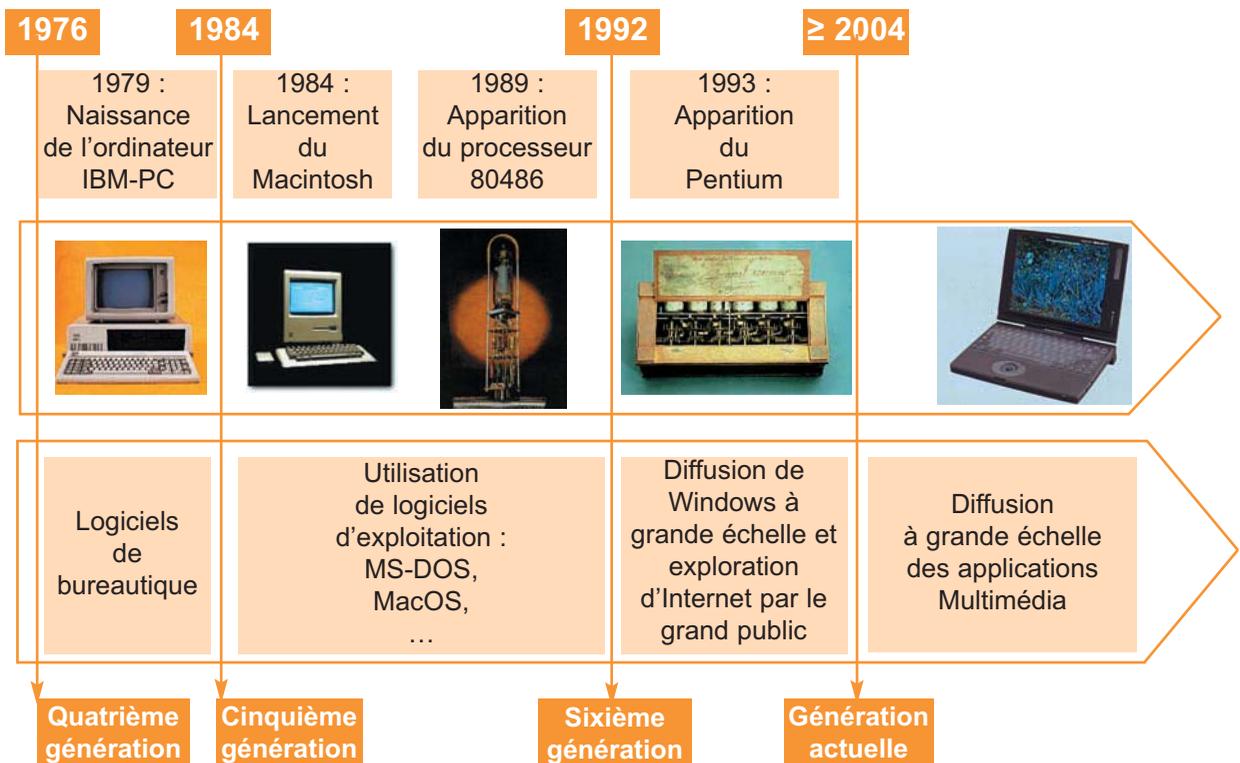


Fig I.2 : Dates clés de l'informatique (suite)

Les différentes générations

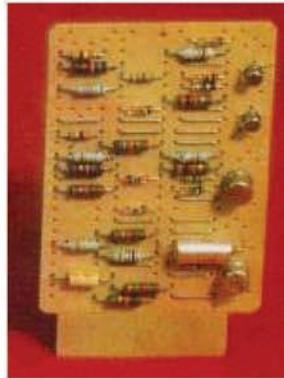
L'histoire de l'informatique est composée de générations successives, correspondant à des innovations majeures dans l'évolution du matériel et du logiciel. Voici une description chronologique de l'histoire de l'informatique classée par de générations (Tableau I.2) :

	1ère génération 1946	2ème génération 1955	3ème génération 1964	4ème génération 1976	5ème génération 1984	6ème génération 1992	Génération actuelle et future ≥2004
Matériel	Ordinateurs à lampes électroniques	Ordinateurs à transistors et à circuits imprimés	Ordinateurs à circuits intégrés (IBM 360, CDC 6000, mini-ordinateur, ...)	Ordinateurs à microprocesseurs Invention des micro-ordinateurs (Apple II , IBM PC, ...)	Macintosh, stations Sun (Microprocesseur : 8 MHz, RAM 512 Ko à 1 Mo)	Microprocesseur : 33 MHz, 640 Ko à 4Mo de RAM (1992), 10M de transistors, 1GHz, 128Mo Super ordinateurs : 150 Giga flops (Fujitsu VPP 700, ...)	Portable miniature, 3 GHz en 2003, sans fil, et/ou invisible : "intelligence ambiante" intégrée aux objets de la vie courante (exemple de la puce RFID) Super ordinateurs : Cray X1 52,4 Téra flops (début 2003) – 1 Péta flops en 2010
Logiciel	Programmation en langage machine	Système d'exploitation batch, programmation en assembleur, puis en FORTRAN 1956, Cobol 1960, Algol 1960	Langages de programmation évolués : Basic 1964, APL 1969, Lisp, Pascal 1969, C 1970	Applicatifs: traitement de textes, tableur, mainframes à mémoire virtuelle	- Système d'exploitation MSDOS, MacOS avec icônes. - Souris	- Butineur pour le WEB (1994), messagerie électronique, - Multimédia, - Jeux 3D	- Entrée/ sortie vocale - Assistance à l'utilisateur par des agents intelligents - Web sémantique
Usage	Calcul de tables pour la balistique	Calcul numérique répétitif	Ordinateur central d'entreprise (mainframe), spécialisé gestion ou calcul	- Micro-ordinateur individuel de bureau, - Services informatiques d'entreprise centralisés	- Micro-ordinateur individuel, familial et de bureau, - apparition progressive des réseaux locaux	- Généralisation de l'ordinateur, - accès au traitement d'images et de sons, - Le Web devient accessible au grand public en 1994	- Concept d'objet - Terminal d'Internet - L'informatique est complètement intégrée à la vie quotidienne : technologie devenue omniprésente.

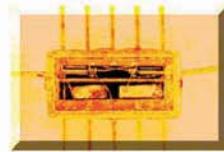
Tableau I.2 : Description chronologique de l'histoire de l'informatique sous forme de générations



IBM 704
1^{ère} génération - tubes



IBM 1400 et 7000
2^{ème} génération - transistors



IBM 300
3^{ème} génération - micromodules

4^{ème} génération
Microprocesseur



Fig I.3 : IBM à travers les quatre premières générations

II. Notions d'information et de numérisation

II.1. Introduction :

Activité 4

Lire le texte suivant :

" Qu'est-ce que cette " information " qu'on stocke, transporte, transforme ? Tout le monde croit savoir ce que c'est. Pour le scientifique, (physicien, informaticien, etc.) l'information a une définition précise qui se réfère toujours à une modification apportée à un support matériel : L'information reçue par un système est le rapport des réponses possibles de ce système avant et après qu'on l'ai reçue.

L'ordinateur impose pour toutes les données la numérisation ou représentation de toutes les informations sous forme de tables numériques stockées en mémoire mais aussi sur des supports divers. D'une part ce mode de représentation des données conduit naturellement à tout mettre sur les mêmes supports (CD-ROM par exemple). Mais surtout les données ainsi représentées deviennent accessibles aux immenses possibilités des moyens modernes de traitement d'information. On peut rechercher, transformer, identifier, comparer, trier n'importe quoi. Mentionnons aussi qu'une information numérisée peut être conservée indéfiniment sans dégradation.

L'information est tout ce qui rend compte d'une situation, d'un état ou d'une action... "

*Histoire, épistémologie de l'informatique
et révolution technologique*

Par **Thibault LIENHYPERTEXTE**, 2003

1/ Définir l'information

2/ Donner les avantages de l'utilisation de l'ordinateur dans le traitement de l'information.

II.2. Les unités de mesure de l'information

Pour pouvoir traiter et conserver l'information, il a fallu créer un système de codage et de mesure. L'information manipulée dans le domaine informatique est devenue mesurable.

Le bit (binary digit) est la plus petite unité de mesure de l'information ; il peut avoir la valeur 1 ou la valeur 0. Pourquoi ce choix ? Pensez à l'interrupteur de votre salle de classe, ses deux états donnent en fait deux informations : lampe allumée ou lampe éteinte. Si la salle contient plus d'un interrupteur, cherchez de combien de manière vous pouvez éclairer la salle.

Vers la fin des années 30, Claude Shannon¹ démontra qu'à l'aide de "contacteurs" (interrupteurs) fermés pour "vrai" et ouverts pour "faux" il était possible d'effectuer des opérations logiques en associant le nombre " 1 " pour "vrai" et "0" pour "faux".

1 Le mathématicien Claude Shannon (1916-2001) est considéré comme le père de la transmission numérique des informations.

Ce codage de l'information utilise la base binaire. C'est avec ce codage que fonctionnent les ordinateurs. Il consiste à utiliser deux états (représentés par les chiffres 0 et 1) pour coder les informations.

L'association de **deux** bits donnent **quatre** états différents (2^2) (Tableau I.3).

Etat	Bit 1	Bit 2
Etat1	0	0
Etat2	0	1
Etat3	1	0
Etat4	1	1

Tableau.I.3 : Valeurs obtenues avec 2 bits

L'association de **trois** bits donnent **huit** états différents (2^3) (Tableau I.4).

Bit 2	Bit 1	Bit 0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Tableau.I.4 : Valeurs obtenues avec 3 bits

Activité 4

Combien de valeurs sont-elles données par une association de n bits ?

Convenons d'appeler " mot " une association de n bits.

Coin de réflexion

Il est évident que si on connaît le nombre de caractères à coder, il devient alors possible de déterminer le nombre de bits nécessaires dans un mot pour les coder. Bien entendu, le minimum de bits coûterait moins cher et donc conviendrait le mieux..

Historiquement, le premier codage a été établi avec un mot de 7 bits donc donnant 128 codes soit 2^7 . Reste après à établir la correspondance entre les caractères à coder et les codes disponibles de telle façon que deux caractères distincts auront deux codes distincts. En fait, il faudra établir une application injective entre l'ensemble des caractères à coder et l'ensemble des codes disponibles. L'un des codages établi par les américains s'appelle l'ASCII (American Standard Code Interchange Information). Il est suivi jusqu'à nos jours (voir tableau à l'annexe I)

Ensuite, il y a eu la convention de prendre des mots à 8 bits générant 256 codes et cette plage est assez large pour pouvoir coder d'autres caractères tels que les caractères grecs, les caractères arabes et les caractères semi graphiques. Ce mot de huit bits est appelé " octet " ou " byte " en anglais. L'octet devient l'unité d'information la plus connue.

Voici des unités multiples de l'octet :

- Un kilooctet (Ko) = 2^{10} octets = 1024 octets
- Un Mégaoctet (Mo) = 2^{20} octets = 1024 Ko = 1 048 576 octets
- Un Gigaoctet (Go) = 2^{30} octets = 1024 Mo = 1 073 741 824 octets
- Un Téraoctet (To) = 2^{40} octets = 1024 Go

N.B. Ces mêmes unités sont représentés différemment dans certains documents diffusés sur Internet. Ils sont même standardisées depuis décembre 1998 par un organisme intitulé l'IEC :

- Un kilooctet (Ko ou Kb) = 1000 octets
- Un Mégaoctet (Mo ou MB) = 1000 Ko = 1 000 000 octets
- Un Gigaoctet (Go ou GB) = 1000 Mo = 1 000 000 000 octets
- Un téraoctet (To ou Tb) = 1000 Go = 1 000 000 000 000 octets

III. Notion de logiciels

III.1. Définition

Un logiciel est un programme ou un ensemble de programmes informatiques assurant un traitement particulier de l'information. Le mot logiciel est la traduction de l'anglais du mot "**software**". Remarquons que le mot anglais "**hardware**" désigne la partie matérielle.

Un logiciel est considéré comme une œuvre scientifique originale. De ce fait, il est protégé juridiquement contre la reproduction et le piratage, ce qui donne lieu à certains droits pour ses propriétaires. Il est développé individuellement ou par équipe. Rappelons que c'est pour cette raison que le piratage est interdit par la loi dans pratiquement tous les pays, y compris la Tunisie.

III.2. Les différents types de logiciels

On peut distinguer deux grandes familles de logiciels : les logiciels systèmes et les logiciels d'application.

1) Les logiciels systèmes

Un logiciel système contrôle le fonctionnement de l'ordinateur, jouant par conséquent le rôle de première interface entre l'homme et la machine. Il gère les travaux essentiels, mais souvent invisibles, relatifs à la maintenance des fichiers sur le disque dur, à la gestion de l'écran, etc. Un logiciel système constitue donc une partie d'un système d'exploitation.

2) Les logiciels d'application

Un logiciel d'application, appelé généralement progiciel, est un ensemble de programmes standards produits par des particuliers ou des sociétés et est plutôt destiné à des utilisations diverses.

Activité 5

Le tableau I.5 présente certaines catégories de logiciels professionnels, lisez-le, et avec l'aide de votre professeur, discutez des caractéristiques et citez d'autres exemples.

Catégories	Caractéristiques du logiciel	Exemples de logiciels
Les traitements de textes	Ce sont des programmes d'application qui permettent de composer et de mettre en forme des textes. Les traitements de texte facilitent toutes les tâches liées à l'édition de documents (suppression, insertion, reformulation, etc.). Certains, permettent même de créer des documents hypertextes et d'assurer le publipostage.	- WORD de Microsoft - TT d'openOffice - WORDPERFECT de Corel
Les tableurs	Ce sont des programmes d'application couramment utilisés pour l'établissement d'un budget, la planification, et d'autres tâches comptables et financières. Dans un tableur, les données et les formules se présentent sous forme de tableaux appelés feuilles de calcul. Les tableurs utilisent des lignes et des colonnes dont les intersections s'appellent des cellules. Chaque cellule peut contenir du texte, des données numériques ou des formules utilisant des valeurs d'autres cellules pour calculer un résultat souhaité. Pour faciliter le calcul, les tableurs disposent de fonctions intégrées qui exécutent des calculs standards.	- EXCEL de Microsoft - Tableur d'openOffice - Lotus1-2-3 de IBM
Les gestionnaires de bases de données	Ils permettent de créer des bases de données et d'effectuer la recherche, le tri ou la fusion de données, ainsi que toute autre requête relative à ces données.	- MYSQL - MS-ACCESS - PARADOX ...
Les logiciels de publication assistée par ordinateur	Ils permettent de combiner textes et graphiques et de produire des documents de qualité professionnelle. Il s'agit d'un processus en plusieurs étapes, dans lequel interviennent différentes classes de matériel et de logiciel. Comme produits finaux, on peut trouver des dépliants, des calendriers, ...	- MS-PUBLISHER - PAGEMAKER ...

Tableau I.5 - Exemples de catégorie de logiciels

III.3. La diffusion des logiciels

Outre la classification par tâches vue précédemment, on peut aussi ranger les logiciels selon leur méthode de diffusion. On distingue alors :

- les logiciels commerciaux standards, développés et vendus essentiellement par des détaillants (revendeurs ou grossistes) ou par de grandes chaînes de distribution pour les produits les plus populaires. Ils demandent beaucoup d'efforts et d'investissements,
- les logiciels du domaine public, à distribution gratuite comprennent les gratuits ou en anglais "freeware" qui sont des logiciels à usage libre et sans droits à payer,
- les partagiciels ou en anglais "shareware" sont des logiciels dont l'usage régulier est assujéti au paiement d'une somme symbolique à son créateur. Le principe du partagiciel, relativement populaire dans les pays anglo-saxons, tend aujourd'hui à se généraliser, notamment grâce aux vecteurs de communication très étendus du réseau Internet,
- les logiciels libres sont des logiciels fournis avec leur code source et dont la diffusion, la modification et la diffusion des versions sont autorisées, voire encouragées. Ils ne sont pas forcément gratuits, même si c'est souvent le cas. A l'opposé, un logiciel propriétaire est un logiciel dont le code source n'est pas librement disponible, c'est à dire dont le code source n'est fourni que sous forme exécutable.

III.4. La sécurité de l'information

1) Introduction

La sécurité de l'information (ou sécurité informatique) est l'ensemble des mesures prises pour protéger un micro-ordinateur et les données qu'il contient. La sécurité informatique dépend du type du système et de l'importance des informations qu'il contient. Elle nécessite l'intervention d'un personnel possédant une expertise à la fois légale et technique. La sécurité des informations peut être réalisée par différentes mesures telles que la sauvegarde des fichiers dans un local séparé, l'utilisation de programmes de défense contre les virus,...

2) Les intrusions

Les intrusions peuvent se faire soit par les virus, et dans ce cas on parle d'infection soit par des programmes espions.

a) Les virus

Un virus est un programme informatique introduit dans un système à l'insu de l'utilisateur. Il existe plusieurs types de virus. Leur effet est souvent destructeur.

Certains sont plus dangereux que d'autres mais d'une façon générale, ils sont nuisibles. Le virus a pour mission première de se propager en infectant les cibles désignées par son concepteur. Après une période d'incubation, il se manifeste de façon plus ou moins agressive. Les actions des virus sont très diverses : affichage d'un message humoristique, destruction de fichiers, blocage de la machine, formatage complet du disque dur, ... Il existe plusieurs types de virus (**Tableau I.6**).



Type de virus	Infection
Virus de fichiers	Ils se greffent sur une application en ajoutant leur code à celui des fichiers exécutables.
Virus compagnons	Ils se contentent de copier un fichier exécutable et ajoute l'extension .COM à la copie infectée qu'ils placent dans le même répertoire..
Virus de démarrage de disque	Ils remplacent le code du démarrage d'un disque par leur propre code. De la même manière, ils peuvent contaminer d'autres lecteurs. Ces virus ne sont pas nombreux mais très répandus.
Virus furtifs	Ce type de virus échappe à la plupart des moyens de détection en se camouflant. Il soustrait, en effet, sa longueur à la longueur du fichier infecté.
Virus polymorphes (ou mutants)	Ces virus ont la capacité de modifier leur aspect à chaque nouvelle contamination d'un fichier. L'élimination de tels virus est ardue car ils sont présents sous différentes identités.
Virus pièges	Ce type de virus associe deux virus : l'un connu et l'autre inconnu. L'utilisateur détecte le premier et le supprime alors que le second virus se propage et infecte le système.
Les chevaux de Troie	Le cheval de Troie (ou Trojan Horse) est un programme qui effectue certaines actions à votre insu comme ouvrir un port, permettant alors à n'importe qui de prendre le contrôle de votre ordinateur.

Tableau I.6 – Quelques types de virus

Remarques

- 1** Les symptômes d'une infection virale ne sont pas toujours évidents. Le ralentissement général de l'ordinateur, les messages d'erreurs inattendus, le plantage de l'ordinateur, la modification de fichiers et en règle générale le comportement anormal du système, sont souvent des signes d'infection.
- 2** Il est conseillé de mettre fréquemment son système d'exploitation à jour en téléchargeant les mises à jour.

b) Les logiciels espions

Un logiciel espion appelé en anglais spyware est, à la base, un programme dont le but est de collecter des informations démographiques et des informations sur l'utilisation de votre ordinateur à des fins publicitaires.

Mais un logiciel espion peut également décrire un logiciel installé sur votre système, qui fonctionne de manière invisible et qui envoie à un tiers des informations sur vous.

Ces logiciels espions sont souvent fournis avec des logiciels gratuits ou des logiciels téléchargés d'Internet. Les logiciels espions sont à la source de nombreux problèmes : non-respect de la vie privée, failles de sécurité, plantage de l'ordinateur,...

3) Les remèdes et les protections

Pour remédier aux intrusions des virus et des logiciels espions, on recommande :

- la prévention contre les intrusions en évitant les téléchargements d'information de sites douteux et l'ouverture des pièces jointes aux mails suspects. Un message électronique suspect est un message dont on ne connaît pas la provenance ou dont on ne reconnaît pas l'expéditeur.
- la protection des fichiers système et des données contre les accès non autorisés
- l'utilisation d'antivirus mis à jour périodiquement.

a) Les antivirus

Un antivirus est un programme utilitaire qui permet de lutter contre les virus. Il empêche l'exécution de codes malveillants qu'il reconnaît. L'interdiction d'accès aux CD-ROM, aux disquettes et autres périphériques d'entrée/sortie, augmente la protection, mais n'arrête pas le risque d'infection. Lors de l'exécution d'un antivirus, l'utilisateur définit essentiellement les éléments à scanner (**Fig I.4**).

Exemples d'antivirus : McAfee, Norton, Unvirus,.

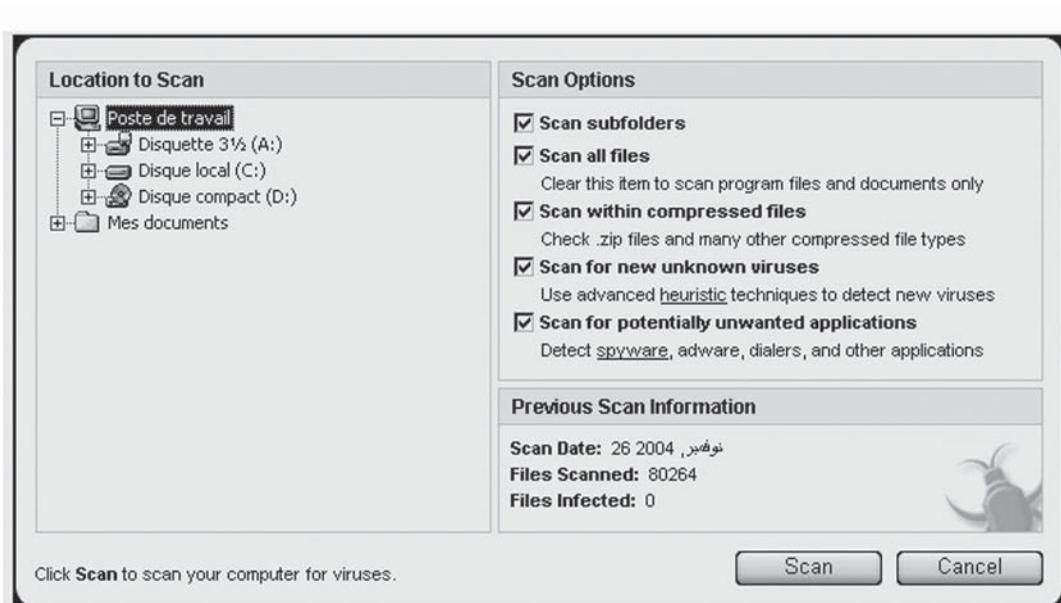


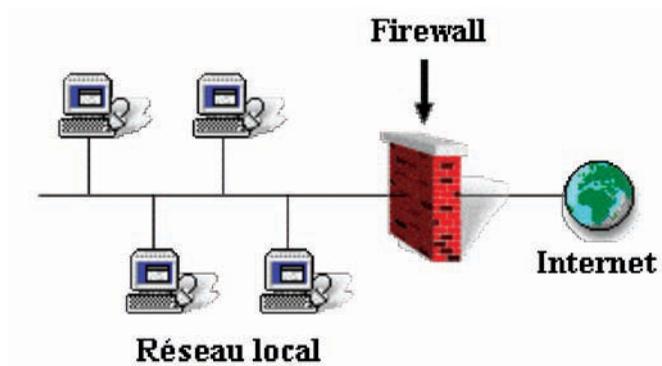
Fig I.4 Exemple d'un écran de démarrage d'un antivirus

b) Le pare-feu

Un pare-feu, appelé aussi coupe-feu ou pare-feu à pour but de contrôler et de filtrer l'accès à un ordinateur ou un réseau d'ordinateurs pour les protéger contre les diverses intrusions.



Il peut être matériel (Fig 1.5) ou logiciel. Dans les deux cas, le pare-feu est activé à l'entrée du système à protéger. Il filtre aussi bien le flux entrant que le flux sortant. Il peut ainsi empêcher l'accès à certains sites et domaines d'un réseau comme celui d'Internet ou interdire à un utilisateur certains services comme la messagerie ou le transfert de fichiers.



IV - Retenons

- L'informatique est, d'une part, une science, et d'autre part, un ensemble de techniques matérielles et logicielles mises en œuvre par cette science.
- L'information traitée par machine peut être du texte, de l'image, du son, de la vidéo, ...
- Le bit est la plus petite unité de mesure de l'information.
- L'octet est l'unité de mesure de l'information la plus utilisée.
- Un logiciel est un ensemble de programmes informatiques assurant un traitement particulier de l'information.
- Il existe deux grandes catégories de logiciels : les **logiciels systèmes** et les **logiciels d'application**.
- On distingue plusieurs **types de distribution de logiciels** selon qu'ils sont **commerciaux**, **publics**, **partagés** ou **libres**.
- L'information doit être protégée contre les virus informatiques et les logiciels espions.

EXERCICES

Exercice n° 1

a) Compléter la grille ci-dessous en se référant aux définitions données :

- 1) Mot en anglais synonyme de logiciel.
- 2) Programme ou logiciel permettant de contrôler et de filtrer l'accès entre un poste et un réseau.
- 3) Unité de mesure composée de 8 bits.
- 4) Nom du projet donnant naissance à un calculateur électronique destiné aux calculs balistiques.
- 5) Mot en anglais désignant l'ensemble des éléments matériels.
- 6) Mot en anglais désignant un logiciel espion qui s'infiltré dans la machine et collecte des informations sur son propriétaire pour un tiers.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										

- 7) Unité de mesure de l'information supérieure au giga.
- 8) Forme possible de l'information pouvant être fixe ou animée.

b) Définir le mot trouvé dans la colonne E.

Exercice n° 2

Donner la valeur maximale et la valeur minimale d'un octet.

Exercice n° 3

Un élève veut transférer un logiciel d'un poste 1 à un poste 2. Sachant que le logiciel occupe 0.0043 Go et qu'une disquette à une capacité de stockage réelle égale à 1.44 Mo. Combien de disquettes sont-elles nécessaires au transfert de ce logiciel ? (Notez toutes les étapes de calcul)

Exercice n° 4

Citer trois moyens pour protéger l'information sur un micro-ordinateur.

Exercice n° 5

Evaluer chacune des affirmations suivantes en écrivant, dans la case correspondante, « Valide » ou « Invalide ». En cas d'invalidité, corrigez l'affirmation en question.

Affirmation proposée	Valide / Invalide	Affirmation corrigée
L'unité de stockage de l'information est le hertz.		
Un gigaoctet est plus petit qu'un megaoctet.		
La domotique est un nouveau domaine généré à la suite de l'emploi de l'informatique dans le secteur bancaire.		
Un kilooctet vaut 1024 octets.		
Le pare-feu peut être matériel ou logiciel.		
Le virus est un élément matériel nuisible.		
L'IBM 360 a été inventé lors de la deuxième génération.		

CHAPITRE 2

Architecture d'un micro-ordinateur

Objectifs

- Identifier les principaux composants internes d'un micro-ordinateur.
- Identifier les principaux périphériques.
- Définir les caractéristiques techniques des principaux composants matériels d'un système informatique.

Plan du chapitre

I. Définition d'un ordinateur

II. Architecture de base d'un micro-ordinateur

- II-1. Le processeur
- II-2. La mémoire
- II-3. Les unités d'entrée/sortie
- II-4. Le bus

III. Retenons

Lecture

Exercices

CHAPITRE 2

Architecture d'un micro-ordinateur

Dans le chapitre 1, vous avez découvert les différentes définitions attribuées au terme Informatique ainsi que ses principaux domaines d'application. Vous avez aussi perçu les différentes formes d'information et le mode de présentation de l'information numérique. L'important maintenant est de savoir comment l'information est-elle transmise vers la machine? Comment y sera-t-elle traitée? et enfin comment pourrons-nous la restituer? Ce chapitre fournit des réponses à ces questions en présentant les différents outils matériels entrant en jeu. Bonne découverte!

I. Définition d'un ordinateur

Le mot "**ordinateur**" a été proposé en 1955 par Jacques PERRET, professeur à la Sorbonne. Il est appelé aussi machine électronique de traitement de l'information; mais le mot le plus répandu est "computer": le calculateur. L'ordinateur est commandé par des programmes enregistrés dans une mémoire. Il est capable d'effectuer des ensembles complexes d'opérations en grande quantité et en un temps très court sans risque d'erreur.

Les ordinateurs sont classés en trois catégories :

- Les **ordinateurs universels (mainframes)** sont très puissants. Ils sont utilisés pour les besoins de la recherche des grandes entreprises et de l'armée. Ils sont de grands calculateurs.



- Les **minis ordinateurs** sont conçus pour des applications spécialisées comme le contrôle de machines industrielles complexes.

- Les **micro-ordinateurs** sont plus petits et plus lents que les précédents, bien que la grande évolution technologique les rapproche de plus en plus des mini-ordinateurs. Leur faible coût et leur encombrement très réduit sont les principales raisons de leur considérable développement.



- Un **ordinateur portable** (en anglais *laptop* ou *notebook*) est un ordinateur intégrant l'ensemble des éléments dont il a besoin pour fonctionner : une alimentation électrique sur batterie, un écran et un clavier. Tous ces éléments sont logés dans un boîtier de faible dimension. L'intérêt principal d'un ordinateur portable par rapport à un ordinateur de bureau est sa mobilité ainsi que son encombrement réduit. En contrepartie le prix est généralement plus élevé pour des performances légèrement moindres.

II. Architecture de base d'un ordinateur

Activité 1

- Mettez sous tension votre micro-ordinateur et activez la fenêtre intitulée «Propriétés système» (Fig.II.1).
- Lisez la partie relative à l'ordinateur et recueillez les termes utilisés dans un tableau en séparant les nouveaux termes des anciens.

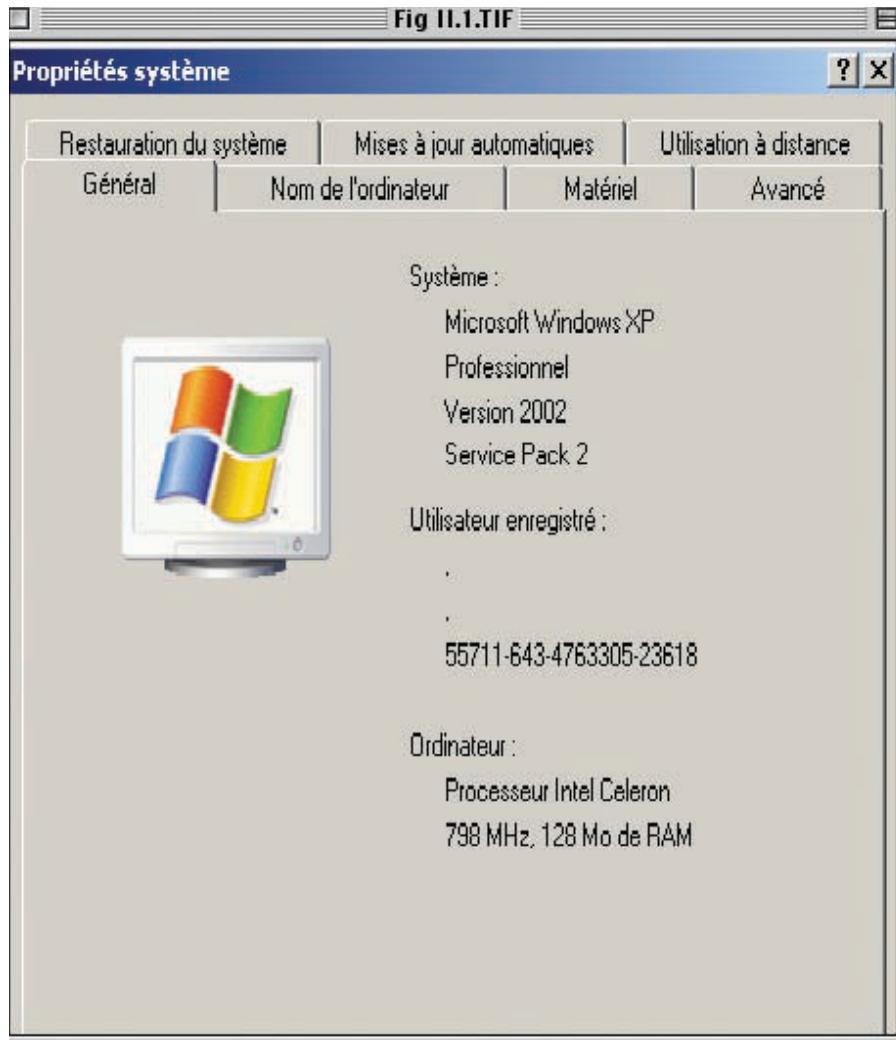


Fig.II.1 : Fenêtre montrant les propriétés système

Activité 2

- Ouvrez la fenêtre du «gestionnaire de périphériques» (Fig II.2).
- Lisez attentivement les outils matériels cités.
- Recueillez, dans un premier tableau, les différents termes utilisés en séparant les nouveaux termes des anciens.
- Avec l'aide de votre professeur, regroupez, dans un deuxième tableau, les différents outils par catégorie.

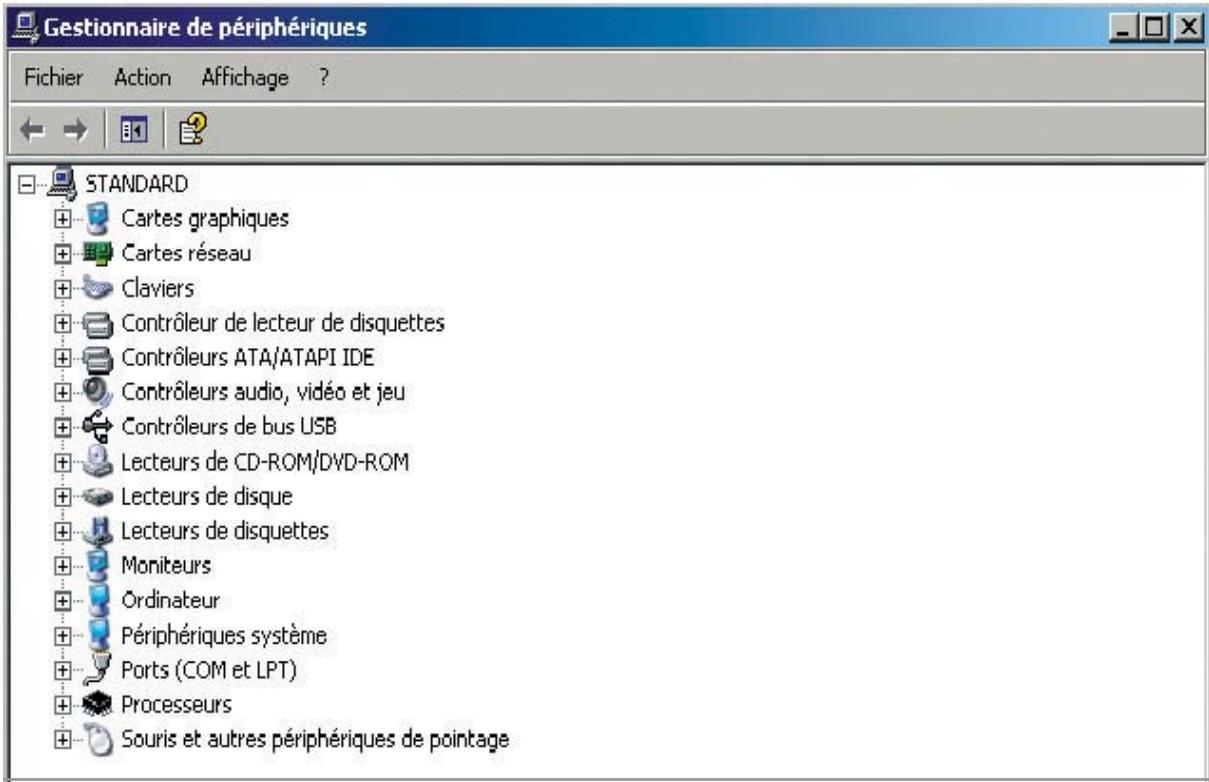


Fig II.2 : Fenêtre relative au gestionnaire de périphériques

On constate que l'univers matériel est immense. Comment allons-nous identifier les différents composants de la machine, comprendre leurs fonctions et surtout trouver un lien entre eux ?

Un ordinateur effectue des traitements sur des informations. Ces traitements sont exécutés par le **processeur**. Un traitement est un ensemble **d'instructions** stockées dans une mémoire avec les données à traiter. Pour que la communication entre le processeur et la mémoire soit possible, un ensemble de **fils conducteurs appelés bus** est mis en place. Par ailleurs il faut que l'utilisateur puisse fournir à l'ordinateur des données et les instructions à suivre, tout comme il doit avoir connaissance des résultats. Il faut donc des dispositifs ou **périphériques d'entrées et de sorties** (Fig II.3).

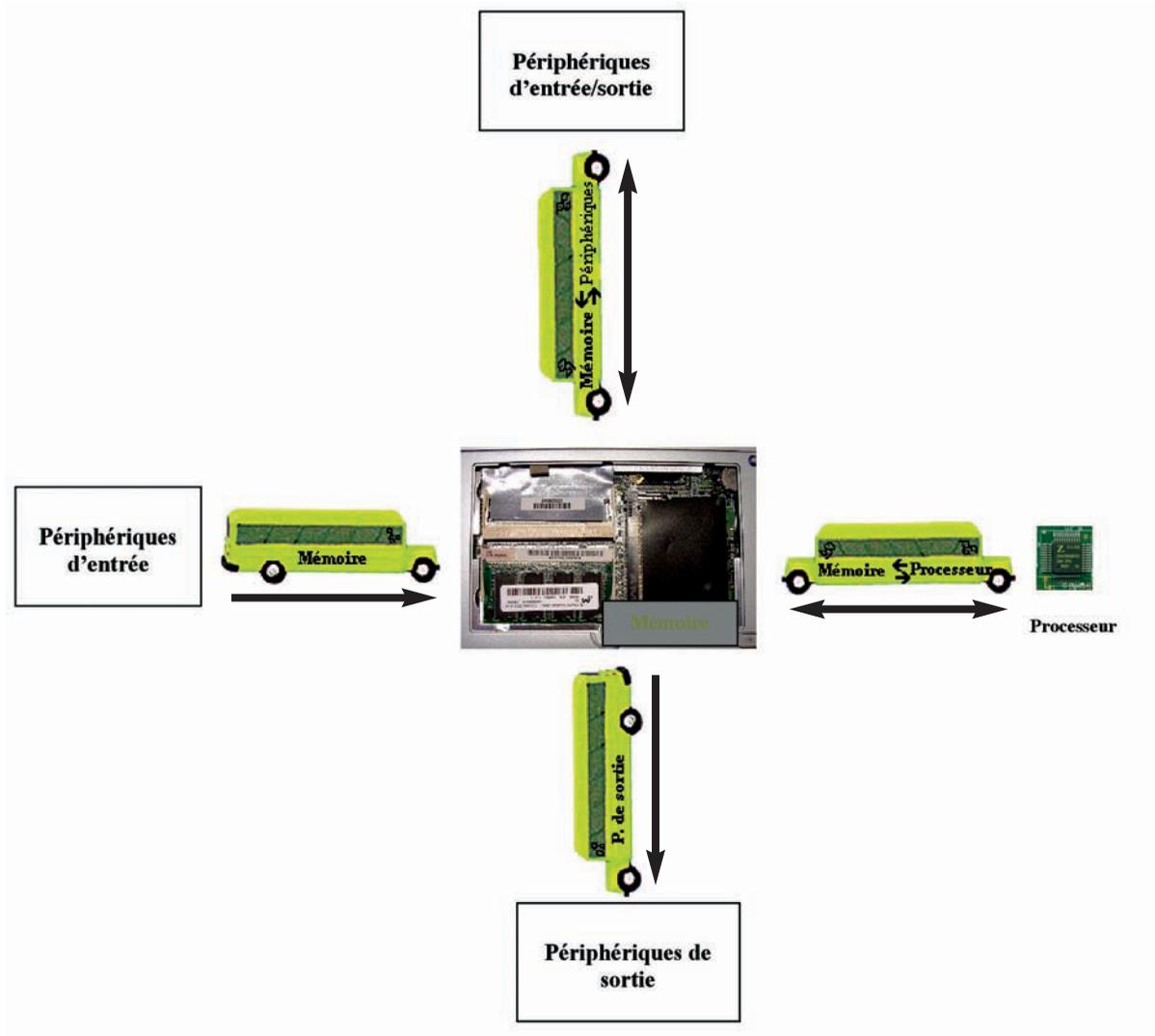
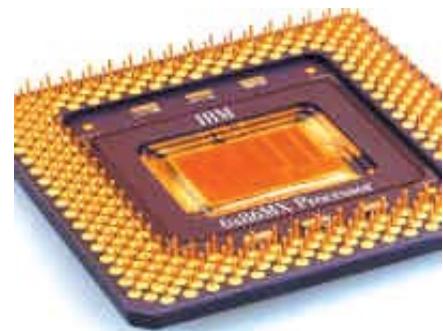


Fig II.3 : Schéma simplifié illustrant le fonctionnement d'un ordinateur

II.1. Le processeur

1) Présentation

Le processeur est un **circuit électronique** destiné à interpréter et à **exécuter rapidement** des instructions : le Pentium d'Intel est un exemple de processeur. L'activité du processeur, est **cadencée** par une **horloge**. Plus la fréquence de l'horloge est élevée, plus rapide sera l'ordinateur. La **fréquence** de cette horloge s'exprime en **Méga Hertz (MHz)** ; par exemple, un ordinateur «PC PENTIUM 3.8 GHz» possède un processeur de type Pentium et une horloge à 3.8 GHz.



Le microprocesseur a connu une évolution impressionnante (**Tableau II.1**) :

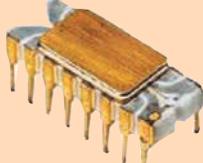
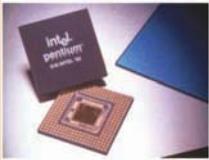
Génération de microprocesseurs	Illustration	Date de création	Caractéristiques
Intel 4004		1971	Sa fréquence est de 108 KHz.
Intel 8086		1986	Sa fréquence est de 5, 8 ou 10 MHz.
Intel 80186		C'est une version légèrement améliorée : sa fréquence est de 8, 10 ou 12.5 MHz	
Intel 80286		1982	Il est capable d'adresser 16777216 emplacements mémoire. L'innovation la plus importante est la capacité de gérer plusieurs programmes à la fois (multi-tâche), si le système d'exploitation le permet.
Intel 80386		1985	Le 386 présentait une fréquence de 16, 20, 25, et 33 MHz.
Intel 80486		1989	Il intègre une mémoire cache de 8 Ko, permettant d'accroître sa vitesse de traitement. Le 486 est le premier microprocesseur à être équipé de plus d'un million de transistors. Le 486 présentait des fréquences 33, 40 et 66 MHz.
Pentium		1993	Le Pentium original fonctionnait à la fréquence de 66 MHz.

Tableau II.1 : Différentes générations de microprocesseurs

Génération de microprocesseurs	Illustration	Date de création	Caractéristiques
Pentium Pro		1995	La fréquence d'horloge peut atteindre 200 MHz.
Pentium II		1997	Il est doté de la technologie MMX, qui permet d'exploiter plus efficacement le son, la vidéo et le traitement d'image. Le Pentium II atteint la fréquence d'horloge de 450 MHz.
Pentium III		1999	70 nouvelles commandes ont été ajoutées pour améliorer son utilisation dans la vidéo et le traitement d'image 3D. Le Pentium III atteint une fréquence de 1GHz.
Pentium 4		Le plus récent	Il est capable d'atteindre des fréquences particulièrement élevées allant jusqu'à 3.2 GHz pour le présent.

Tableau II.1 (suite) : Différentes générations de microprocesseurs

2) Fonctionnement

L'activité du processeur peut être décrite comme suit :

1. le processeur charge, à partir de la mémoire centrale, l'ensemble des instructions à exécuter et les place dans une mémoire caractérisée par un accès rapide aux données appelée **mémoire cache niveau 1** ;
2. l'**unité de commande (U.C.)** charge l'instruction à exécuter, à partir de la mémoire cache, et la décode ;
3. le **compteur ordinal (C.O.)**, un registre spécial, pointe sur la prochaine instruction à exécuter ;
4. l'**unité arithmétique et logique (U.A.L.)** exécute les instructions arithmétiques et logiques ;
5. les résultats sont stockés temporairement dans des **registres** spéciaux.

Cette suite d'étapes s'appelle couramment cycle de chargement-décodage-exécution, cycle qui est au coeur du fonctionnement de tous les ordinateurs.

II.2. Les mémoires internes

1) Présentation

La mémoire est la partie de l'ordinateur dans laquelle programmes et données sont rangés : c'est le principe de la **machine de VON NEUMANN**.

Une mémoire est formée d'un certain nombre de cellules ou cases. Une cellule sert à stocker de l'informations. Chaque cellule à un numéro appelé **adresse (Fig II.4)**.

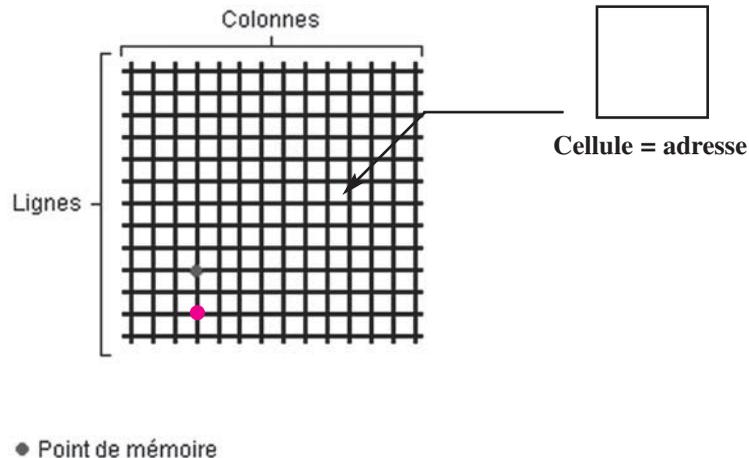


Fig II.4 : Présentation simplifiée de la mémoire

Les performances d'une mémoire sont caractérisées par :

- a) La capacité d'une mémoire correspond au nombre total d'octets pouvant y être enregistrés. La capacité est couramment exprimée en **Mégaoctets (Mo.)**.
- b) Le temps d'accès à une mémoire correspond à l'intervalle de temps qui s'écoule entre le lancement, par l'organe de commande, d'une opération de lecture d'un octet en mémoire et le moment où le contenu de cet octet est disponible à la sortie de la mémoire. Il est exprimé en nanosecondes. L'ordinateur comporte plusieurs mémoires : la mémoire vive, la mémoire morte, la mémoire centrale,...
- c) Le temps de cycle représente l'intervalle de temps minimum entre 2 accès successifs.
- d) Le débit : volume d'information échangé par unité de temps (b/s).

2) Types de mémoire

a) La mémoire vive.

Appelée communément RAM. (Random Access Memory = Mémoire à accès aléatoire). Elle est accessible en lecture et en écriture. Dès la mise en marche de l'ordinateur, tout programme à exécuter par l'utilisateur doit être installé dans cette mémoire avec toutes les structures de données qu'il va manipuler. Cette mémoire est volatile. En effet, toute coupure de courant détruit son contenu.

Durant les années 80, la mémoire RAM se présentait sous la forme de circuits intégrés individuels que l'on implantait directement sur la carte mère. Ce procédé n'est plus utilisé de nos jours. Depuis la fin des années 80, la mémoire vive se présente exclusivement sous la forme de petites cartes d'extension, communément appelées "barrettes de mémoire" sur lesquelles sont fixés plusieurs circuits intégrés (Tableau II.2).

	Types de barrettes	Caractéristiques
Single Inline Memory Module	SIMM 30	Elle équipait des ordinateurs 286 et 386. Elles sont à 30 broches.
	SIMM 72	Elle équipait des ordinateurs 486 et les premiers pentium. Elles sont à 72 broches.
Dual Inline Memory Module	DIMM	Elle est utilisée actuellement.

Tableau II.2 – Exemples de types de barrettes mémoire



Fig II.5 : Barrette de mémoire SIMM30, qui n'est plus utilisée actuellement

Fig II.6 : Barrette de mémoire SIMM72, peu utilisée de nos jours



Fig II.7 : Barrette de mémoire vive DIMM

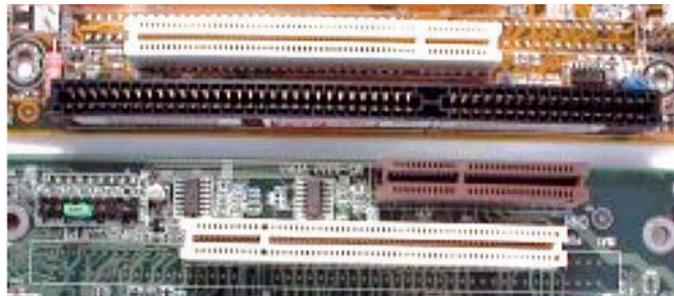


Fig II.8 : Connecteurs dans lesquels prennent place les barrettes de mémoire DIMM

b) La mémoire morte

Notée ROM. (**Fig.II.10**) (de l'anglais Read Only Memory, appelée en français Mémoire à lecture seulement), elle est accessible en lecture seulement. Cette mémoire, programmée par le constructeur, est essentiellement utilisée pour contenir les programmes de base servant à la mise en route de l'ordinateur. En effet elle contient la **ROM BIOS** qui est une puce contenant les programmes et pilotes permettant le démarrage de l'ordinateur. Contrairement à la RAM., la mémoire morte ne perd pas son contenu lors d'une coupure du courant : c'est une mémoire permanente.



Fig II.9 : Mémoire ROM

Les ROM. admettent des variantes :

- les **PROM** (Programmable Read Only Memory) sont des mémoires livrées vierges par le fabricant et qui se transforment en ROM. une fois qu'elles ont été programmées une première fois par l'utilisateur ;

- les **EPROM** (Erasable Programmable Read Only Memory), sont des PROM. effaçable et programmable. La différence majeure entre les EPROM et les PROM est que les EPROM sont dotés d'une cavité en quartz placée sur la puce et qui, exposée à des rayons ultraviolets, fait effacer le contenu de l'EPROM. Lorsqu'elle est exposée à des rayons U.V d'une intensité suffisante, une EPROM revient à son état initial.

- les **EEPROM**. (Electrically Erasable Programmable Read Only Memory), sont aussi des PROM. effaçables par un simple courant électrique. Ces mémoires sont appelées aussi **mémoires flash** (ou ROM. flash).

Activité 3

- Mettez votre ordinateur sous tension.
- Soyez attentifs et ne ratez pas l'affichage de la configuration **BIOS Fig II.10** (en appuyant sur la touche appropriée (souvent il s'agit de la touche SUPPR ou F2) pour ouvrir le programme SETUP).
- Explorez le contenu de cette fenêtre.

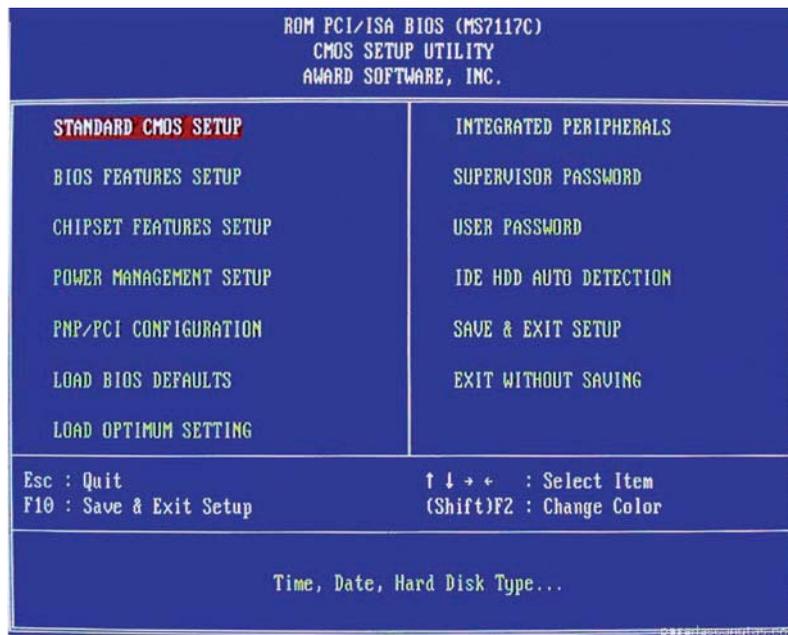


Fig II.10 : Menu principal de la configuration BIOS

c) La mémoire cache

C'est une mémoire dont le temps d'accès est très court comparé à celui de la RAM. Elle est de type **SRAM (Static RAM)** c'est à dire que son contenu n'a pas besoin d'être périodiquement rafraîchi d'où sa rapidité. Elle est coûteuse.

Elle peut être soit :

- 1) intégrée au processeur et elle est appelée dans ce cas mémoire cache de niveau 1 (**Level 1 cache en anglais**);
- 2) installée sur la carte mère, entre le microprocesseur et la RAM et dans ce cas, elle est dite du niveau 2 (**Level 2 cache en anglais**).

d) La mémoire CMOS

Ce type de mémoire est utilisé pour sauvegarder la configuration personnalisée de la BIOS (date, heure, mot de passe, options de fonctionnement,...). Elle est alimentée par une pile sur la carte mère. La durée de vie de cette pile est d'environ trois ans. (Fig II.11).

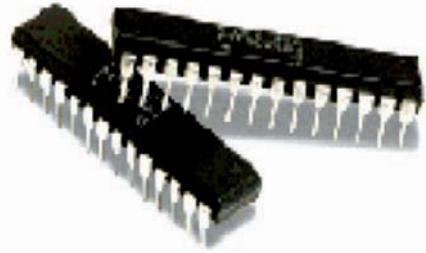


Fig II.11 : La mémoire CMOS et le pile d'alimentation

e) La mémoire flash

La mémoire flash est une mémoire à semi-conducteurs, non volatile et réinscriptible, c'est-à-dire une mémoire possédant les caractéristiques d'une mémoire vive mais dont les données ne se volatilisent pas lors d'une mise hors tension. En raison de sa vitesse élevée, de sa durabilité et de sa faible consommation, la mémoire flash est idéale pour de nombreuses applications comme les appareils photos numériques, les téléphones cellulaires, les imprimantes, les assistants personnels (PDA), les ordinateurs portables, ou les dispositifs de lecture ou d'enregistrement sonore tels que les baladeurs MP3.

Les précédents composants sont disposés sur un circuit imprimé appelé **carte mère** (Fig II.12) logée dans le boîtier de l'unité centrale. La carte mère est caractérisée par le type de son Chipset, le nombre de slots libres et parfois le nombre de processeurs.

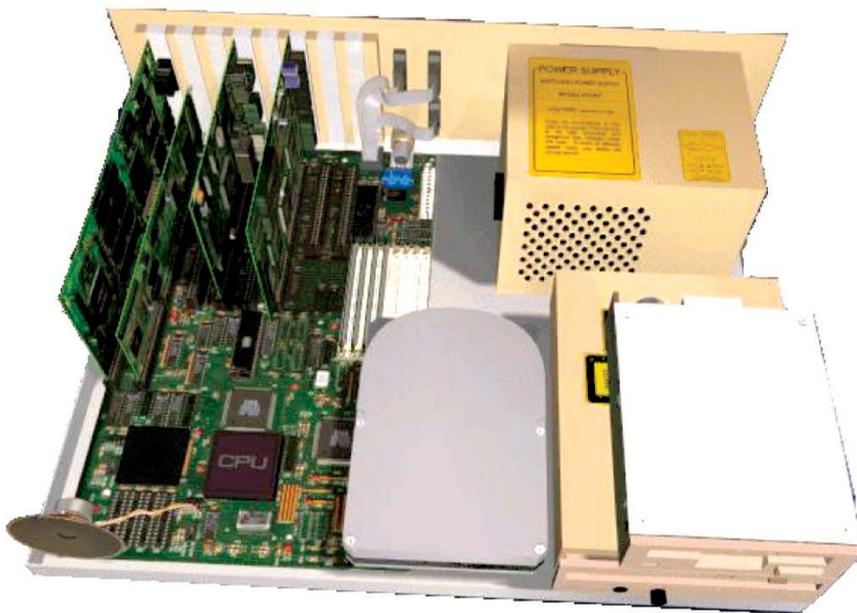
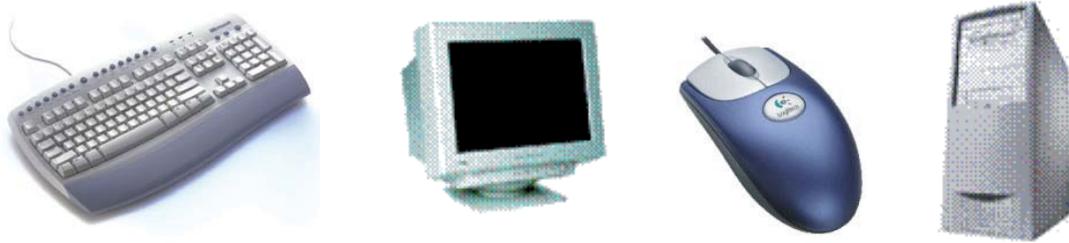


Fig II.12 : Composants d'une carte mère

II.3. Les unités d'entrée/sortie



1) Définition

Les unités périphériques sont les parties de l'ordinateur qui permettent à l'utilisateur de communiquer avec l'unité centrale. Elles sont donc indispensables à l'utilisation d'un ordinateur. Elles comprennent les organes d'**entrée**, les organes de **sortie** et les organes d'**entrée /sortie**. Les périphériques ont des tâches spécifiques dont le fonctionnement est parfois très différent de celui de l'unité centrale. L'écran effectue l'affichage, l'imprimante est conçue pour les impressions, le scanner numérise des images, etc. Pour que l'ordinateur puisse dialoguer et communiquer avec un périphérique il faut passer par une **interface** souvent pilotée par un programme appelé **pilote** ou **driver** en anglais.

2) Les périphériques d'entrée

Les périphériques d'entrée recueillent les informations (programmes ou données), qui sont ensuite transformées de façon à être utilisables par la machine et transférées à la mémoire centrale.

a) Le clavier : considéré comme le périphérique d'entrée principal. Tous les claviers d'ordinateur comportent un jeu standard de caractères, disposés suivant une normalisation de dactylographie (française : AZERTY ou anglaise : QWERTY). Cette nomination provient de la position des lettres de la première rangée du pavé alphanumérique.



b) La souris : dispositif de pointage. Il devient indispensable pour tous les environnements graphiques vu la facilité qu'il apporte. Une souris est composée d'un système de repérage (mécanique ou optique) et un système de commande souvent avec deux touches et parfois avec une roulette. La souris, comme le clavier, utilise un **port série** pour se brancher à l'ordinateur. La communication se fait par fils, par infrarouge ou par onde radio.

c) Le scanner : appelé aussi numériseur d'images, c'est un périphérique d'entrée utilisant des capteurs sensibles à la lumière pour balayer les données imprimées sur papier ou sur un autre support. Les zones claires et sombres (ou les couleurs) du papier sont alors converties en format numérique.



d) Le microphone : périphérique d'entrée de l'information sous forme sonore. Il est connecté à une **carte son** qui joue le rôle d'**interface** entre le microphone et l'ordinateur..



e) La caméra vidéo : permet de capter des images réelles et les numériser afin d'être traitées par l'ordinateur. La Web Cam est une toute petite caméra vidéo branchée sur un PC, elle est, surtout, utilisée par les internautes.

f) Autres : On peut citer d'autres périphériques d'entrée tels que :

Le lecteur de codes à barres,



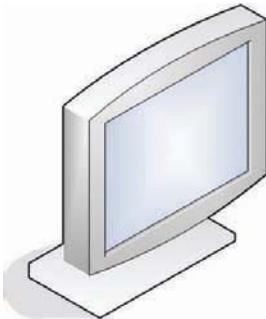
la manette de jeux
(joystick)



3) Les périphériques de sortie

Les périphériques de sortie transmettent à l'extérieur, sous une forme compréhensible par l'utilisateur, les résultats recueillis dans la mémoire centrale.

a) L'écran : c'est un support permettant de visualiser des données. On distingue les écrans à tube cathodique ou CRT (Cathode Ray Tube) et les écrans à cristaux liquides ou plasma (Liquid Crystal Display, écran à cristaux liquides abusivement appelés "écrans plats"). L'écran est relié à la machine grâce à une carte graphique qui se charge de l'affichage des images et des textes sur l'écran. La résolution graphique est le nombre de points lumineux (pixel : PICture Element) sur l'horizontale et sur la verticale de l'écran. Cette résolution est gérée aussi par logiciel. Le rendu dépend d'une autre caractéristique de l'écran qui est sa grandeur ou sa dimension. C'est la longueur d'une diagonale de l'écran et elle est souvent donnée en pouces et parfois en cm.



b) L'imprimante : c'est un périphérique permettant de transférer des textes ou des images sur du papier ou tout autre support d'impression (transparents, tissus, etc...).

L'imprimante joue un rôle important puisqu'elle imprime le résultat final d'un traitement fait par l'ordinateur.

On distingue:

- les imprimantes matricielles à aiguilles,
- les imprimantes à jet d'encre,
- les imprimantes laser.



Activité 4

Aidé par votre professeur, établissez un tableau comparatif des trois types d'imprimantes.

Considérer les points suivants :

- Technique de fonctionnement
- Avantages
- Inconvénients
- Unités de mesure de la vitesse d'impression

c) **Autres** : Il existe d'autres périphériques de sortie tels que :



les haut-parleurs,



le vidéo projecteur,

4) Les périphériques d'entrée/sortie

Les périphériques d'entrée/sortie permettent l'échange de données dans les deux sens (lecture et écriture). Les mémoires de masse sont considérés comme des périphériques d'entrée/sortie et comme autre exemple, on peut citer le modem.

a) **Les supports de stockage** : Les supports de stockage sont des outils matériels très utilisés, les plus utilisés sont :

• **Le disque dur** : Le disque dur est le lieu de stockage des données de l'ordinateur, contrairement à la mémoire vive qui est une mémoire volatile servant uniquement de zone de transit d'informations lors du fonctionnement de la machine. La caractéristique la plus importante du disque dur est sa capacité (exprimée en gigaoctets), car elle détermine la quantité de données et de programmes que l'on peut stocker. Toutefois il ne faut pas oublier sa vitesse de rotation mesurée en tours par minute.



- **Le flash disque** : De la taille d'un briquet, ce disque miniature est la solution idéale pour transporter et échanger tous types de fichiers : l'utilisateur transfère les données de son disque dur sur le disque dur flash USB, qu'il glisse dans sa poche et qu'il n'a plus qu'à brancher sur le port USB de l'ordinateur de destination.



- **La disquette** : Malgré sa capacité réduite de stockage (1.44 M.O.) et une lenteur parfois pénible, la disquette est un support d'échange encore largement utilisé à cause du grand parc informatique ancien. Son avenir n'est pas prometteur avec l'apparition d'autres supports plus performants. Notons que pour lire ou écrire dans une disquette, l'ordinateur devra être équipé du **lecteur approprié**.

- **La disquette ZIP** : Elle permet la sauvegarde d'une grande quantité de données compressées. Une disquette ZIP peut avoir une capacité de stockage de 100 Mo, 200 Mo et même plus. On a besoin d'un lecteur zip pour pouvoir exploiter ce type de disquettes.

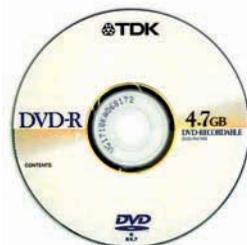


- **La bande magnétique** : L'utilisation des cartouches à bande magnétique reste la solution la plus pratique pour sauvegarder une grande quantité d'informations et d'archives. L'inconvénient majeur de ce type de support est que l'accès aux données est assez lent et se fait de façon **séquentielle**.

- **Le CD (Compact Disk)** : Le disque compact rivalise avec les autres supports sur pratiquement tous les plans. C'est un support de stockage optique de capacité considérable de l'ordre de 700 Mo, soit 500 fois plus qu'une disquette. En plus, le transfert de données est plus élevé. Le CD se présente sous plusieurs formes : CD-ROM, CD-RW, CD-AUDIO, CD-Vidéo. Comme la disquette, le CD ne peut être consulté qu'à partir d'un **lecteur CD**.



- **Le DVD (Digital Versatile Disc)** : c'est le rival du CD. Il a une capacité de stockage qui dépasse les 4 Go. Actuellement, les micro-ordinateurs sont équipés d'un lecteur DVD et parfois d'un graveur DVD. Ce dernier lit toutes sortes de CD et de DVD selon les zones et grave sur les CDR, CDRW, DVDR et les DVDRW.



b) Modem : Modulateur- DEModulateur, il est utilisé généralement entre deux ordinateurs pour échanger des données à travers le réseau téléphonique (**Fig.II.13**). En effet, le modem permet l'accès à Internet, la gestion de la communication téléphonique, les connexions par télécopie et par minitel. C'est un convertisseur analogique/numérique. Il permet de transformer les informations analogiques en des informations numériques compréhensibles par un système informatique, et inversement. Le modem peut être interne ou externe ; sa vitesse est exprimée en BPS (Bits Par Seconde) ou en Bauds.

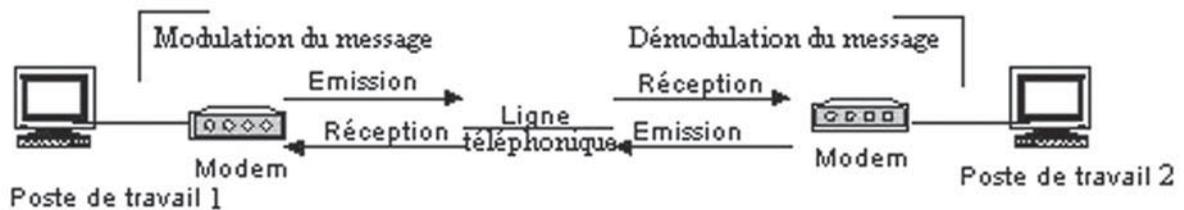


Fig II.13 : Fonctionnement d'un modem

5) Les ports d'entrée/sortie



Activité 5

Demandez à votre camarade de débrancher quelques périphériques pour que vous puissiez les remettre. Inversez les rôles.

Les ports d'entrée-sortie sont des éléments matériels de l'ordinateur, permettant au système de communiquer avec des éléments extérieurs, c'est-à-dire d'échanger des données, d'où l'appellation d'*interface d'entrée-sortie*, notée parfois **interface d'E/S**.

a) Les ports série : Les ports série (exemple : Le port RS232, PS2, ...) sont les premières interfaces ayant permis aux ordinateurs d'échanger des informations avec des périphériques. Le terme série désigne un envoi de données via un fil unique: les bits sont envoyés les uns à la suite des autres.



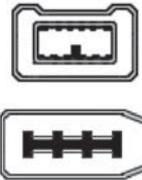
Ports série DB25

b) Les ports parallèles : La transmission de données en parallèle consiste à envoyer des données simultanément sur plusieurs canaux ou fils. Les ports parallèles présents sur les ordinateurs personnels permettent d'envoyer simultanément 8 bits (un octet) par l'intermédiaire de 8 fils.

c) Les ports USB : Le bus **USB** (Universal Serial Bus en anglais = Bus série universel) est, comme son nom l'indique, basé sur une architecture de type série. Il s'agit toutefois d'une interface entrée-sortie beaucoup plus rapide que les ports série standards. Le port USB2 peut aller jusqu'à **400Mbps (mégabits par seconde)**, un **taux de transfert** suffisamment élevé pour satisfaire les plus exigeants. De nos jours, de **plus en plus** de périphériques se branchent sur des ports USB : imprimante, scanner, modem, webcam, etc.



d) Firewire ou IEEE 1394 : Le bus **IEEE 1394** (nom de la norme à laquelle il fait référence) a été mis au point à la fin de l'année 1995 afin de fournir un système d'interconnexion permettant de faire circuler des données à haut débit en temps réel. Il s'agit ainsi d'un port, équipant certains ordinateurs, permettant de connecter des périphériques (notamment des caméras numériques).



6) Les adaptateurs

Ce sont des cartes électroniques d'extension appelées également des cartes contrôleurs ou simplement des interfaces. Elles contrôlent le fonctionnement d'un composant interne ou externe de l'ordinateur. Parmi ces cartes, on peut citer :

a) La carte graphique : Carte permettant de traiter les informations concernant l'affichage afin de les envoyer au moniteur. Initialement, la carte graphique servait uniquement à traiter l'affichage. Avec la généralisation de la 3D, la carte graphique soulage le processeur pour le traitement des calculs 3D en utilisant son propre processeur.



b) La carte son : capable de convertir des signaux numériques en signaux analogiques et inversement, elle permet de brancher un microphone, des enceintes acoustiques, éventuellement de faire l'interfaçage avec des instruments MIDI.

c) La carte réseau : permet la connexion d'un ordinateur sur un réseau local. Elle s'occupe du transfert des données entre l'ordinateur et le système de communication du réseau.

d) La carte contrôleur disques : permet de connecter au boîtier différentes mémoires de masse (disque dur, disquette, CD, DVD, ...)

II.4. Les bus

1) Définition et rôle

Tous les organes internes et périphériques fonctionnent sous le contrôle permanent de l'unité de commande. Ils sont reliés par des lignes de matériaux conducteurs appelés bus qui servent à transporter les informations sous forme de signaux électriques.

Un bus est capable de véhiculer des signaux correspondant fondamentalement à trois types d'informations : adresses (qui identifient l'emplacement d'un périphérique ou d'une donnée en mémoire), données et commandes (telles que par exemple les signaux d'horloge de synchronisation).

Un bus est enfin caractérisé par le nombre et la disposition de ses lignes ainsi que par sa fréquence de fonctionnement exprimée en mégahertz (MHz) : un bus qui comporte par exemple 16 lignes parallèles est un bus 16 bits, ce qui signifie que chacune des lignes permet de transporter 1 bit à la fois. La fréquence de fonctionnement de ce bus permet de calculer le débit d'informations qu'il peut véhiculer dans un même laps de temps. Si notre bus 16 bits évolue à une fréquence de 8 MHz (1 Hertz = 1 impulsion/s), on peut estimer débit théorique maximal à :

$$8 \times 16 = 128 \text{ Mbit/s, soit } 128/8 = 16 \text{ Mo/s}$$

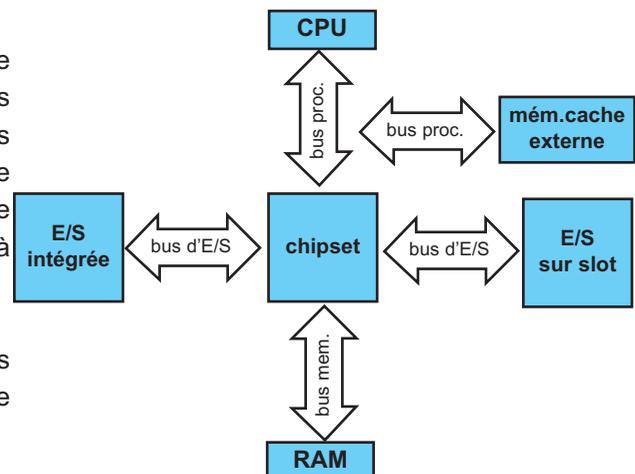
2) Les différents types de bus présents dans un ordinateur

a) Le bus processeur : Il constitue le canal de communication entre le processeur central, les puces complémentaires qui lui sont associées (habituellement appelées chipset) et la mémoire cache externe (Niveau 2). Le rôle de ce bus est de transférer des signaux de et vers le processeur à une rapidité maximale.

b) Le bus mémoire : Il assure le transfert des données entre le processeur et la mémoire principale (RAM).

c) Le bus d'adresses et le bus de commandes : Ils sont en fait des sous-ensembles des bus processeur et bus mémoire. Ces derniers sont constitués de différentes lignes spécialisées, certaines d'entre elles chargées de véhiculer des adresses, d'autres des données ou des commandes ; les lignes d'adresses constituent le bus d'adresses, les lignes de commandes déterminent le bus de commandes. Ce dernier véhicule tous les signaux utilisés pour synchroniser les différentes activités qui se déroulent dans les unités fonctionnelles de l'ordinateur : signaux d'horloge, signaux de lectures/écritures, signaux d'interruptions, etc.

d) Le bus de données ou bus d'entrées/sorties : Il est tout comme les bus d'adresses et de commandes un sous-ensemble des bus processeur et mémoire. Il assure la transmission des données à traiter entre les différents éléments de l'ordinateur.



III - Retenons

- Un ordinateur est une machine électronique capable d'effectuer automatiquement des traitements.
- Un ordinateur est composé d'une unité centrale de traitement et d'une mémoire centrale.
- L'unité centrale de traitement regroupe l'unité de commande, l'unité arithmétique et logique, et d'autres éléments tels que le compteur ordinal et les registres. Ces différentes parties sont reliées aux organes extérieurs de l'ordinateur par des bus.
- Les mémoires sont de différents types : RAM, ROM, mémoire cache (niveau 1 et niveau 2), mémoire CMOS.
- Les mémoires auxiliaires sont essentiellement utilisées pour stocker les informations exploitables par l'ordinateur.
- Les périphériques sont de trois types : périphériques d'entrée, périphériques de sortie et périphériques d'entrée/sortie.
- Les adaptateurs sont des cartes d'extension appelées également des cartes contrôleurs ou simplement des interfaces.
- Les ports sont des circuits servant à l'entrée ou à la sortie de données.
- Les bus permettent de véhiculer des informations (données, adresses et commandes) entre les différents composant du micro-ordinateur.

EXERCICES

Exercice n° 1

Citez quelques exemples de mémoires auxiliaires et donner leurs caractéristiques. Formulez la réponse sous forme de tableau comparatif.

Exercice n° 2

Soit la liste des mots suivants :

**réseau téléphonique – données – Internet –
mémoires de masse – échanger – d'entrée/sortie – modem**

Complétez le texte suivant par les mots de la liste ci-dessus :

Les périphériques permettent l'échange de dans les deux sens (lecture et écriture). Les sont considérés comme des périphériques d'entrée/sortie et comme autre exemple, on peut citer le (MODulateur DEModuleur) qui est utilisé généralement entre deux ordinateurs pour des données à travers le..... En effet, le modem permet l'accès à....., la gestion de la communication téléphonique, les connexion par télécopie et par minitel.

Exercice n° 3

Evaluez chacune des affirmations suivantes en écrivant, dans la case correspondante, «Valide» ou «invalide». En cas d'invalidité, corrigez l'affirmation en question.

Affirmation proposée	Valide/Invalide	Affirmation corrigée
Le processeur est cadencé par une pile.		
La longueur de la diagonale du moniteur est une caractéristique de ce dernier.		
Un moniteur de 17" veut dire qu'il pèse 17 demi kilogrammes.		
Plus la fréquence de l'horloge est grande, plus le processeur est rapide		
Les claviers Azerty sont les plus utilisés dans les pays anglo-saxons.		
La mémoire cache niveau 2 est à l'extérieur du processeur.		
La RAM sert à contenir les programmes à exécuter et les données à traiter.		
La CMOS contient le nom du dernier programme utilisé.		
Un lecteur/graveur DVD est un périphérique de sortie.		
Le bus processeur est caractérisé par une grande vitesse.		

Exercice n° 4

Quel est le contenu de la BIOS ?

Une partie de la BIOS contient des informations relatives à la configuration de la machine. Par quel moyen pouvez-vous introduire ces informations ?

Exercice n° 5

Par quel moyen physique, le processeur communique-t-il avec la RAM ?

Exercice n° 6

Remplissez le tableau suivant en attribuant à chaque élément les caractéristiques demandées :

Eléments	Unité de mesure	Fixe/amovible	volatile/permanente	Interne/externe
ROM				
Disquette				
RAM				
Disque dur				
Flash disque				

CHAPITRE 3

Systèmes d'exploitation et réseaux informatiques

Objectifs

- Définir le rôle d'un système d'exploitation.
- Utiliser les principales fonctions d'un système d'exploitation.
- Travailler dans un environnement réseau.

Plan du chapitre

A) Système d'exploitation

- I. Présentation et rôles
- II. Les principales fonctions
- III. Notions de fichiers et de répertoires
- IV. Apprentissage des fonctions de base d'un système d'exploitation

B) Réseaux informatiques

- I. Présentation
- II. Les différents types
- III. Les avantages
- IV. Les ogistiques matérielles logicielles
- V. Les topologies
- VI. L'exploitation d'un réseau local
- VII. Retenons

Exercices

CHAPITRE 3

Systèmes d'exploitation et réseaux informatiques

Dans les chapitres précédents, vous étiez amenés à identifier différents éléments matériel et logiciel en relation avec le micro-ordinateur. Qui permet de gérer tous ces éléments ? Et surtout qui permet à l'utilisateur d'exploiter toutes ses richesses en exprimant aisément ses besoins ? Son nom illustre bien son rôle : c'est le système d'exploitation. Exploitez alors votre machine en maîtrisant ses fonctions de base... Bon apprentissage !

A) Système d'exploitation

Activité

- 1) Découvrir le système d'exploitation installé sur les postes de votre laboratoire.
- 2) Donner la différence entre un système utilisant des interfaces (comme windows) et un autre utilisant des commandes (MS - DOs).

I. Présentation et rôles

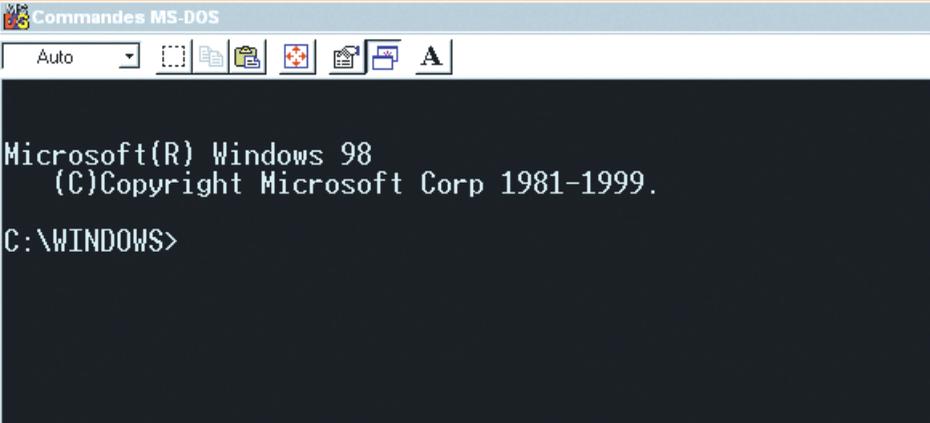
Un système d'exploitation est un ensemble de programmes nécessaires au lancement et à l'utilisation d'un ordinateur. Il est généralement installé dans un disque dur de la machine. Il est possible d'installer plus qu'un système d'exploitation dans le même micro-ordinateur. Toutefois, on devra choisir un seul système au cours d'une session de travail.

Au moment du démarrage de l'ordinateur et après avoir choisi le système d'exploitation à utiliser, l'ordinateur cherchera l'emplacement de ce dernier pour le mettre en marche. Une partie du système sera placée dans la RAM : c'est le noyau du système comportant les fonctionnalités les plus utilisées. D'autres fonctionnalités disponibles dans le système d'exploitation existant sous forme de programmes appelés utilitaires seront lancés dans le cas de besoin.

Une fois chargé dans l'ordinateur, un système d'exploitation permet à l'utilisateur un certain nombre d'opérations telles que le lancement et l'exécution de programmes, la gestion des dossiers et des fichiers, le formatage de supports de stockage, ...

A l'origine, chaque ordinateur possédait son propre système d'exploitation. Le problème fût que les utilisateurs devaient sans cesse s'initier à de nouvelles syntaxes dès qu'ils changeaient d'ordinateurs. Dans les années 70, la standardisation fut la solution et l'on vit apparaître tour à tour, Unix, CP/M, MS/DOS, MacOS, OS/2, Windows 3.1, 3.11, 95, 98, 2000, XP, Linux, ...

Le tableau suivant donne un aperçu de quelques systèmes d'exploitation qui ont marqué le parcours de l'informatique.

Type de système d'exploitation	Caractéristiques
	<p>Unix est un système d'exploitation en constante évolution, dont le codage est à source libre. Ce système s'est développé dans les années 70 et continue à être utilisé. Plusieurs programmeurs l'utilisent et ont développé une interface graphique et conviviale.</p>
	<p>Avant l'élaboration du logiciel Windows, Microsoft avait déjà mis au point un autre système d'exploitation, MS-DOS. Ne fonctionnant pas avec une interface graphique, l'utilisateur devait exploiter l'ordinateur par des commandes tapées au clavier. On retrouve encore le DOS à travers Windows sous l'utilitaire Invite de commandes sous l'onglet Accessoires du menu Démarrer.</p> 
	<p>Un des premiers systèmes ayant une interface graphique, le Mac OS d'Apple est un système qui ne peut être installé que sur les ordinateurs Apple. La dernière version, X, connu sous le nom de code Jaguar intègre des éléments de UNIX.</p>
	<p>Mis sur pied par la firme Microsoft, le système d'exploitation Windows demeure l'un des plus répandus chez les utilisateurs de PC. Il possède une interface graphique facile à utiliser.</p> <p>Diverses versions ont marqué ce système d'exploitation : Windows3, Windows3.1, Windows95, Windows98, WindowsNT, Windows2000, ...</p> <p>La version actuelle est connue sous le nom de WindowsXP.</p>
<p><i>Linux</i></p> 	<p>Linux est un système d'exploitation à code source libre (Open Source), ce qui signifie que les utilisateurs peuvent en modifier la programmation et le codage pour le personnaliser. Donc, Linux n'appartient à personne, contrairement à Windows et Mac OS qui appartiennent à des compagnies. Le logiciel fut développé, depuis 1991, par des milliers de programmeurs volontaires apportant chacun une amélioration au code. Plusieurs distributions de Linux existent, chacune avec des interfaces.</p>

II. Les principales fonctions

Les principales fonctions d'un système d'exploitation sont :

- **La gestion du processeur** : Le système d'exploitation est chargé de gérer l'allocation du processeur entre les différents programmes.
- **La gestion de la mémoire vive** : Le système d'exploitation est responsable de la gestion de l'espace mémoire alloué à chaque application. En cas d'insuffisance de mémoire physique, le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée "mémoire virtuelle". La mémoire virtuelle permet de faire fonctionner des applications nécessitant plus de mémoire qu'il n'y a de mémoire vive disponible sur le système.
- **La gestion des entrées/sorties** : le système d'exploitation permet d'unifier et de contrôler l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes appelés également gestionnaires de périphériques ou gestionnaires d'entrée/sortie.
- **La gestion des fichiers** : le système d'exploitation gère la lecture et l'écriture dans le système des fichiers et les droits d'accès aux fichiers par les utilisateurs et les applications.

Activité

Copiez le tableau suivant sur votre cahier puis remplissez le nom et le rôle du système d'exploitation pour chaque fonctionnalités.

Fonctionnalités	Rôles
Gestion du processeur	
Gestion de la mémoire vive	
Gestion des entrées/sorties	
Gestion des fichiers	

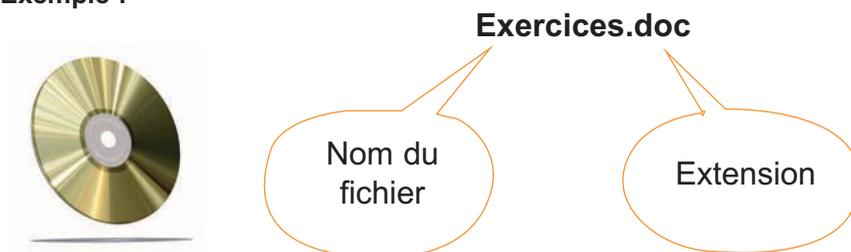
III. Notion de fichiers et de répertoires

III.1. Les fichiers

1) Définition

Un ordinateur permet le traitement des informations. Celles-ci sont stockées dans des entités logiques appelées "fichiers". Chaque fichier peut renfermer des données ou des programmes. Un fichier est identifié par un nom obligatoire et une extension facultative sous la forme **nom.extension**.

Exemple :



2) Rôle de l'extension

L'extension (ou le suffixe) sert à identifier le type du fichier, en voici quelques exemples.

Exemples d'extension	Signification
DOC	Document créé sous le MS-WORD
XLS	Feuille de calcul créée sous MS-EXCEL
PPT	Présentation élaborée sous MS-POWERPOINT
BMP, JPG, GIF, WMF...	Image, graphique, dessin ou photo numérique
MID, WAV, MP3, ...	Son, Musique, Parole, ...
AVI, MP4, CAM, ...	Séquence vidéo
HTM	Page WEB
EXE, COM, BAT	Fichier exécutable
PAS, LOG, BAS, ...	Programme écrit en un langage de programmation

Quelques extensions de fichiers

Activité

Enrichissez le **tableau précédent** avec l'aide de votre enseignant.

III.2. Les répertoires

1) Définition

Un répertoire (appelé également dossier) est un objet informatique pouvant contenir des fichiers et/ou des dossiers. Imaginez une grande boîte contenant des objets et plusieurs autres petites boîtes, qui à leurs tours peuvent contenir des objets et des boîtes, ...

2) L'arborescence

Les répertoires peuvent être organisés sous forme d'arborescence.

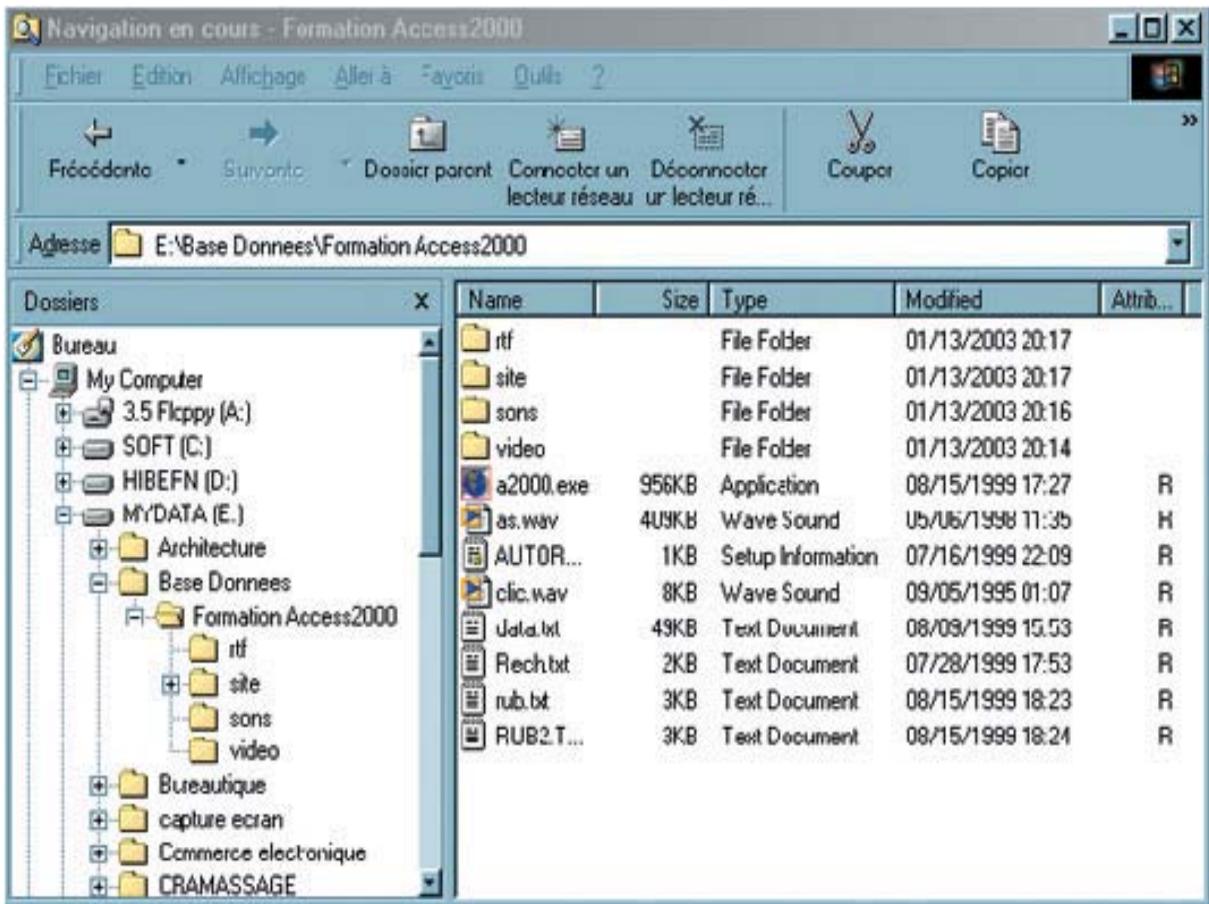


Fig.III.1 – Représentation d'un ensemble de répertoires sous forme arborescence

Le répertoire «Formation Acces 2000» est un **répertoire père** par rapport aux dossiers «Site» et «Sons», ces derniers sont appelés **répertoires fils**.

Activité

- 1/ Présentez les composants d'un ordinateur sous forme d'arborescence.
- 2/ Décrivez votre arborescence en employant les termes **répertoire racine**, **répertoire fils** et **répertoire père**.

IV. Apprentissage des fonctions de base d'un système d'exploitation

IV.1. Le Bureau

Le Bureau est l'écran qui s'affiche une fois que vous avez ouvert une session de travail. Le Bureau contient généralement des icônes **raccourcis** vers des **programmes** et des **documents**, des barres d'état et de tâches, des images de fond comme arrière plan, ... L'utilisateur peut choisir les caractéristiques des éléments de son Bureau de travail à sa convenance.



Fig III.2- Exemple de Bureau de Windows

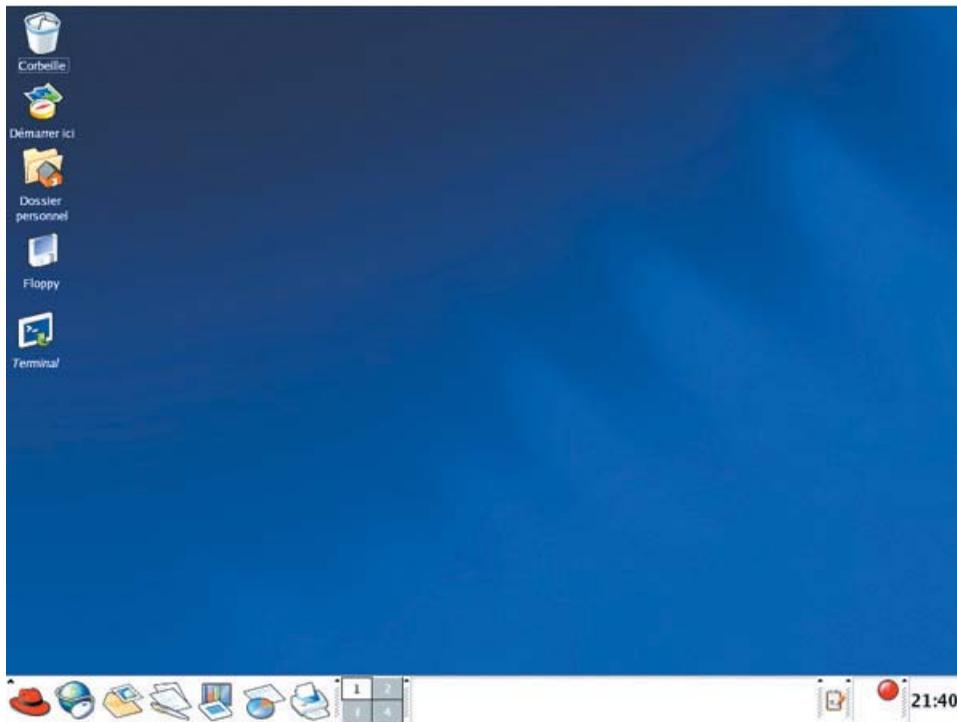


Fig III.3- Exemple de Bureau de Linux

Arrière plan :

C'est une image ou un document qui apparaît en arrière plan sur le Bureau lors du démarrage du micro-ordinateur.

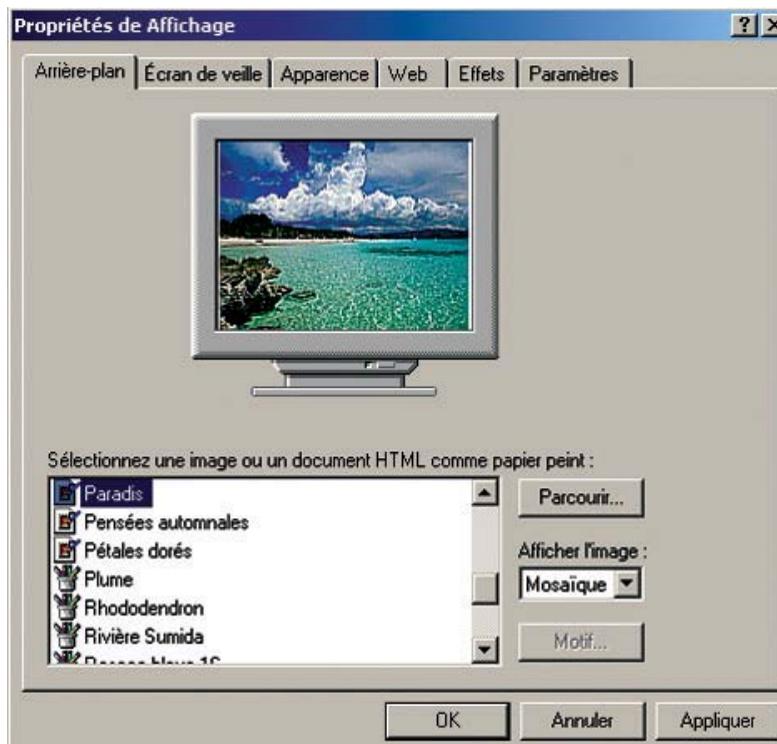


Fig III.4- Choix d'un arrière plan du Bureau sous Windows



Fig III.5 Choix d'un arrière plan du Bureau sous Linux

L'écran de veille :

Généralement une image qui apparaît à l'écran quand on n'utilise pas la souris ou le clavier pendant une durée fixée par l'utilisateur.

Fig III.6 Choix d'un écran de veille sous Windows



Fig III.7 Choix d'un écran de veille sous Linux

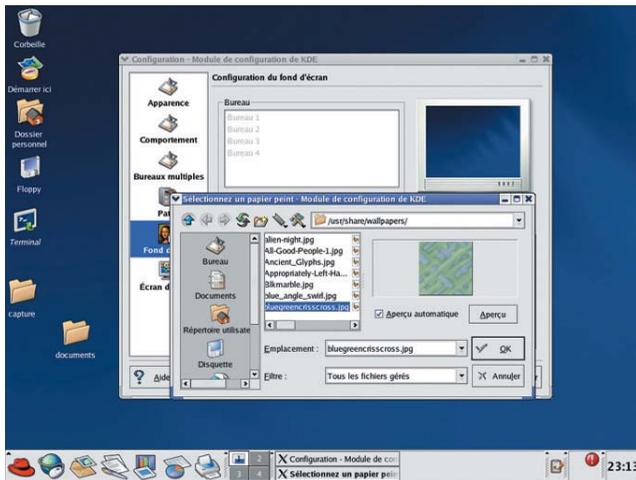


Fig III.7 Choix d'un écran de veille sous Linux

Par défaut, le Bureau est la page d'accueil des fonctionnalités suivantes :

- **La barre des tâches :** La barre des tâches s'affiche en bas de l'écran. Elle comprend le bouton **Démarrer**, que vous pouvez utiliser pour démarrer rapidement un programme, rechercher un fichier ou accéder à l'aide.

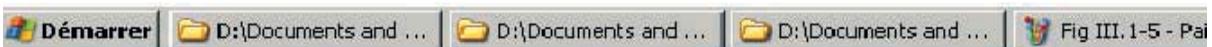


Fig III.8 Barre des tâches



Fig III.9 Menu démarrer

- **Mes documents** : Utilisez ce dossier comme l'emplacement de stockage par défaut pour les documents, les graphiques et les autres fichiers, y compris les pages Web enregistrées.
- **Le poste de travail** : On l'utilise pour afficher nos lecteurs de disquette, de disque dur, de CD-ROM, ainsi que celui des lecteurs du réseau.

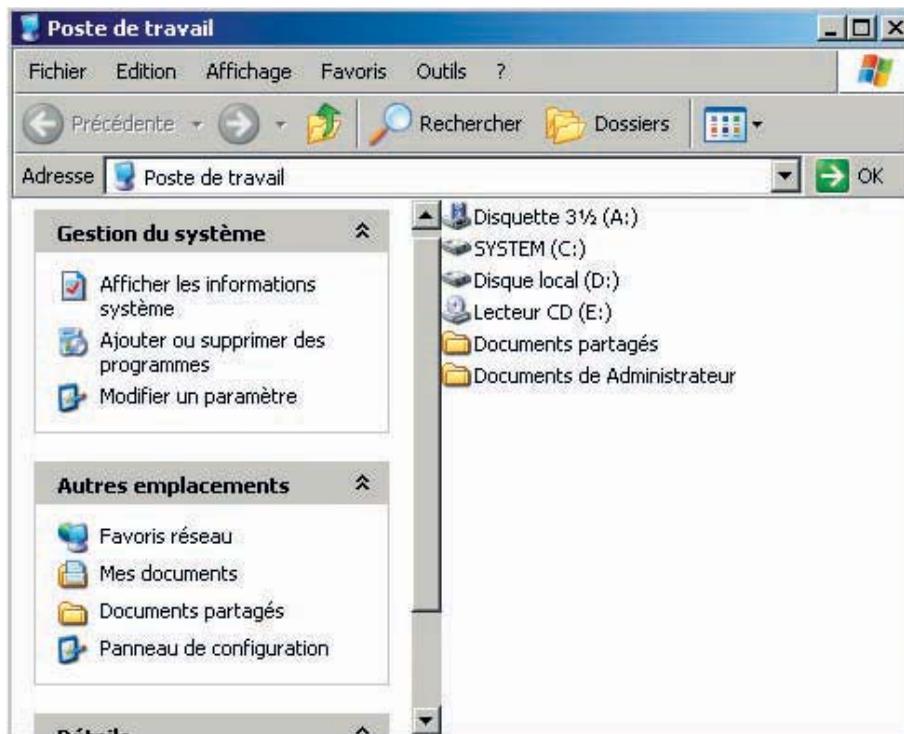


Fig III.10 Poste de travail

- **Les favoris réseau** : Ils sont utilisés pour détecter les ressources partagées dans la totalité du réseau auquel votre ordinateur est connecté.

Remarques

- 1 A chaque démarrage d'un programme, le système d'exploitation lui associe une fenêtre et 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.
- 2 Si un programme n'apparaît pas dans le menu Programmes ou l'un de ses sous-menus, vous pouvez consulter le Bureau pour chercher un **raccourci** ou pour **effectuer une recherche** de ce programme.

b) Quitter un programme

Pour quitter un programme, il suffit de :

- Cliquer sur **Quitter** du menu **Fichier** de votre programme,
- Vous pouvez également quitter un programme en cliquant sur **X** se trouvant à l'extrémité droite de la barre de titre.

c) Changer la taille d'une fenêtre

Cliquez sur le bouton adéquat dans le coin supérieur droit de la fenêtre. 

- i) Cliquez sur le bouton  pour réduire la fenêtre à un bouton de la barre des tâches. Pour rétablir la taille précédente de la fenêtre réduite, cliquez sur le bouton de la barre des tâches.
- ii) Cliquer sur le bouton  pour afficher la fenêtre en plein écran.
- iii) Une fois que vous avez agrandi une fenêtre, cliquez sur le bouton  pour rétablir la taille précédente de la fenêtre.

IV.3. Fichiers et dossiers

1) L'organisation des fichiers et des dossiers

Pour retrouver l'ensemble des fichiers et des dossiers contenus dans un support de stockage, vous pouvez activer l'**explorateur** de Windows en pointant sur le bouton **Démarrer** et en cliquant sur le bouton droit de la souris.



Fig III.13 Activer l'explorateur

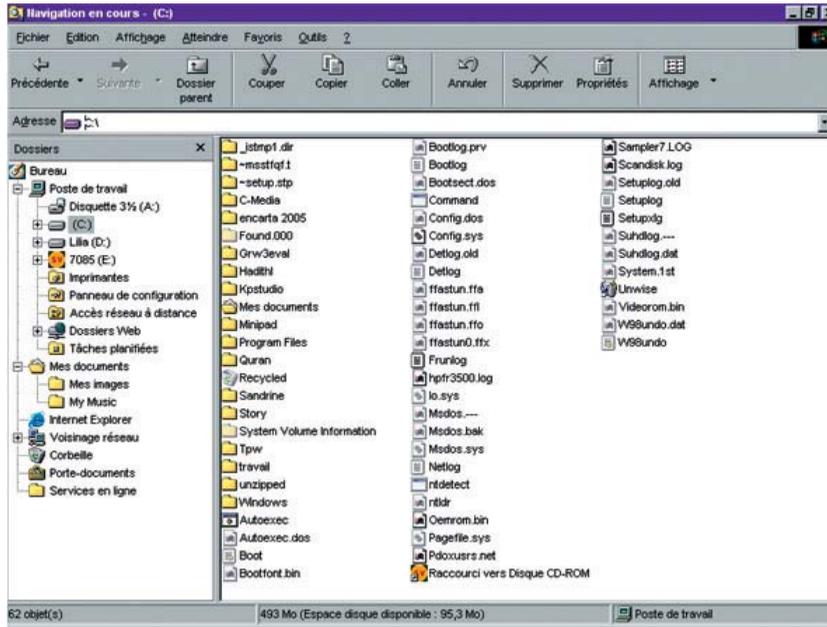


Fig III.14 Explorateur Windows

2) La gestion des fichiers

Plusieurs opérations peuvent être effectuées sur un fichier.

En cliquant sur le bouton droit de la souris, on obtient :

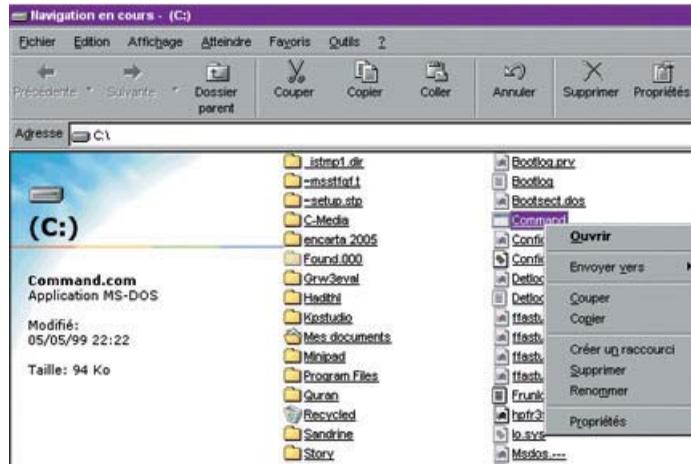


Fig III.15 Gestion des fichiers

Activité

Localisez un fichier puis cliquez sur le bouton droit de la souris.

En se basant sur cette fenêtre, vous pouvez énumérer les opérations que peut subir un fichier :

- ouvrir,
- imprimer,
- numériser,
- compresser,
- copier dans une disquette ou dans le répertoire **Mes documents**, transformé sous forme d'un raccourci sur le Bureau, envoyé par E-mail à un destinataire. Toutes ces possibilités sont contenues dans un sous menu intitulé **Envoyer vers**,

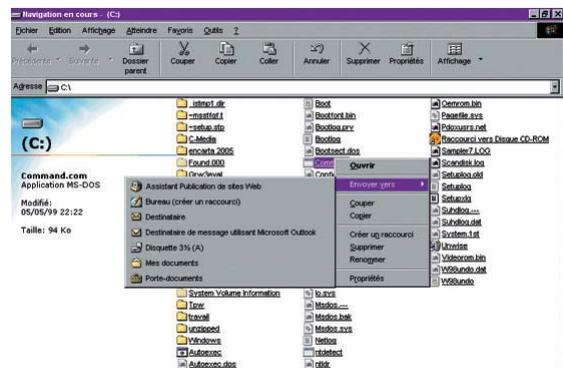


Fig III.16 Opérations sur un fichier

- déplacer,
- copier,
- supprimer si on veut l'envoyer à la corbeille,
- renommer / modifier.
- afficher les propriétés (taille, type, emplacement...)

Activité

Dans cette activité, on vous demande de choisir un fichier puis d'appliquer chacune des actions suivantes :

- Ouverture,
- Copie dans une disquette,
- Renommage du fichier copié sur la disquette sous le nom " Fnew ",
- Déplacement de " Fnew " dans le dossier " Mes documents ",
- Suppression de " Fnew " de " Mes documents ".
- Modifier les propriétés.

Dans un tableau, relevez les étapes à suivre pour chaque action.

Attention

N'utilisez aucun des caractères suivants pour nommer un fichier ou un dossier :

| / \ * < > ? ' ' _

En activant **Propriétés**, on aura accès à différentes informations sur le fichier (type, taille, emplacement, date de création, de modification et du dernier accès) et on pourra, par mesure de sécurité, cacher le fichier ou le rendre accessible en lecture seulement.

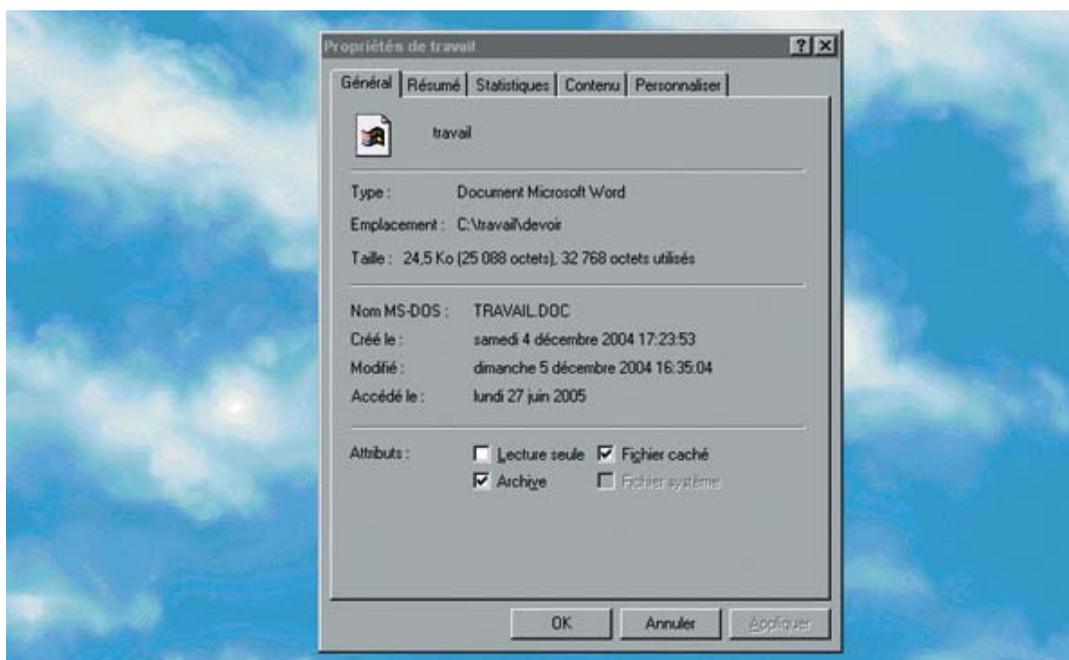


Fig III.17 Propriétés d'un fichier

3) La gestion des dossiers

Plusieurs opérations peuvent être effectuées sur un dossier. Mais, d'abord, un dossier doit être créé. Pour ceci, il suffit :

- d'accéder à l'emplacement voulu de la création.
- de cliquer sur le bouton droit de la souris
- de choisir la Commande Nouveau Dossier
- de nommer le dossier

Remarque : On peut utiliser l'explorateur disponible.

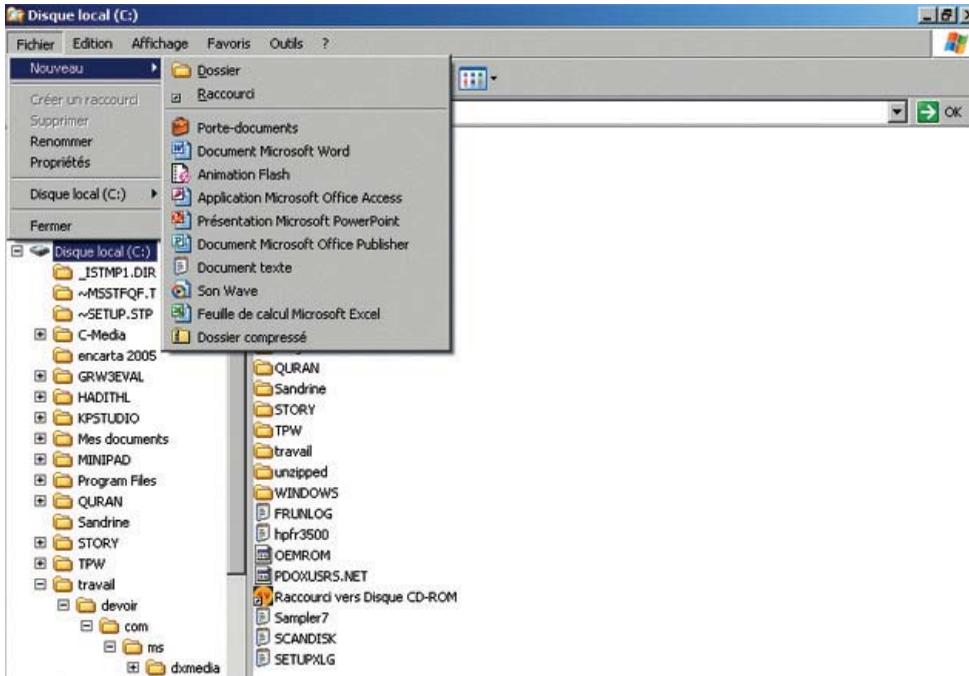


Fig III.18 Création d'un dossier

Activité

Localisez un dossier puis cliquez sur le bouton droit de la souris. En se basant sur cette fenêtre, vous pouvez énumérer les opérations que peut subir un dossier. En effet, un dossier peut être :

- ouvert,
- exploré,
- partagé,
- compressé,
- scanné,
- copié dans une disquette ou dans **Mes documents**, transformé sous forme d'un raccourci sur le Bureau, envoyé par E-mail à un destinataire. Toutes ses possibilités sont contenues dans le sous menu intitulé **Envoyer vers**,
- coupé si on veut le déplacer ailleurs,
- copié,
- supprimé si on veut l'envoyer à la corbeille,
- renommé.

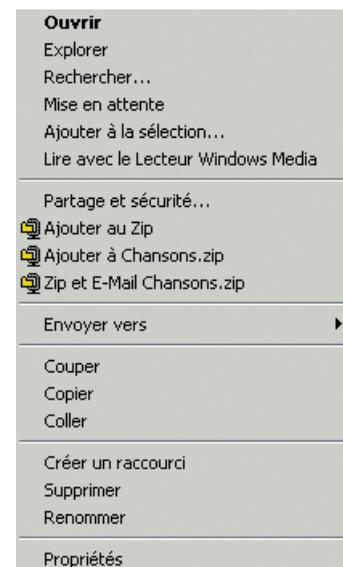


Fig III.19 Fenêtre relative à un dossier

Activité

Dans cette activité, on vous demande de choisir un dossier puis d'appliquer chacune des actions suivantes :

- Ouverture,
- Partage,
- Copie dans une disquette,
- Renommage du dossier copié sur la disquette sous le nom " Fdnew ",
- Déplacement de " Fdnew " dans le dossier " Mes documents ",
- Suppression de " Fdnew " de " Mes documents ".

Dans un tableau, relevez les étapes à suivre pour chaque action.

Comme pour le fichier, en activant **Propriétés**, on aura accès à différentes informations sur le dossier (type, taille, emplacement, contenu, date de création, de modification et du dernier accès) et on pourra, par mesure de sécurité, cacher le dossier ou le rendre accessible en lecture seulement.

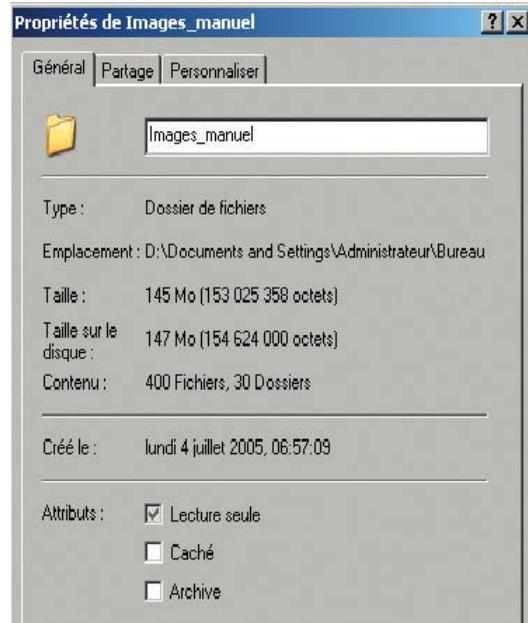


Fig III.20 Propriétés d'un dossier

4) La recherche des fichiers et des dossiers

Activité

On se propose de mettre tous les fichiers textes du disque dur dans un dossier qu'on appellera «Documents».

- 1/ Créez le dossier " Documents ",
- 2/ Recherchez les fichiers textes,
- 3/ Copiez ces fichiers dans le dossier créé.

Pour rechercher un fichier ou un répertoire :

1. Cliquez sur **Démarrer**, pointez sur **Rechercher**, puis cliquez sur Fichiers ou dossiers,
2. Dans la zone **Rechercher les fichiers ou les dossiers nommés**, tapez tout ou une partie du nom du fichier ou du dossier que vous voulez trouver,
3. Pour rechercher des fichiers contenant du texte spécifique, dans **Contenant le texte**, tapez le texte que vous souhaitez trouver,
4. Dans la zone **Regarder dans**, cliquez sur le lecteur, le dossier ou le réseau sur lequel vous souhaitez effectuer la recherche,

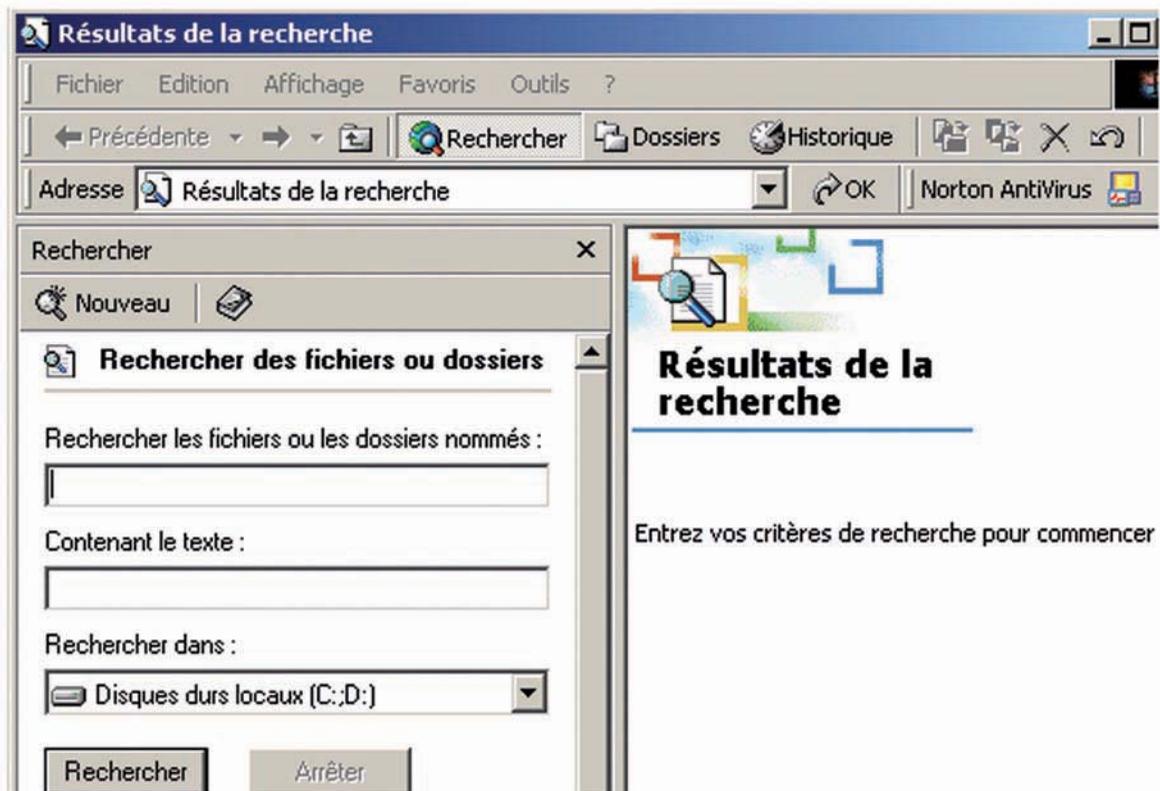


Fig III.21 Recherche des fichiers ou dossiers

5. Pour spécifier des critères de recherche supplémentaires, cliquez sur **Options de recherche**, puis sur une ou plusieurs des options suivantes pour approfondir votre recherche :

- Sélectionnez **Date** pour rechercher les fichiers qui ont été créés ou modifiés à une date précise ou entre deux dates spécifiques,
- Sélectionnez **Type** pour rechercher un type spécifique de fichiers, tels qu'un document texte ou un document compressé,
- Sélectionnez **Taille** pour rechercher des fichiers d'une taille spécifique. Sélectionnez **Options avancées** pour spécifier des critères de recherche supplémentaires,

6. Cliquez sur **Rechercher**.

N.B. Vous pouvez employer les caractères génériques suivants :

- L'astérisque (*) : lorsqu'il s'agit de remplacer un ensemble de caractères.
- Le point d'interrogation (?) : lorsqu'il s'agit de remplacer un seul caractère.

B) Les réseaux informatiques

Face à votre ordinateur, vous utilisez vos ressources et vos documents. Regardez autour de vous et plus précisément les postes de vos camarades. Pouvez-vous échanger avec eux des données, des messages ? Est-il possible d'imprimer votre document sur l'imprimante installée au poste du professeur ? La réponse est positive, vous n'avez qu'à porter un intérêt aux différents câblages et à la manière dont sont disposées les machines de votre laboratoire. Bonne découverte !

Activité

Soit les équipements informatiques suivants :

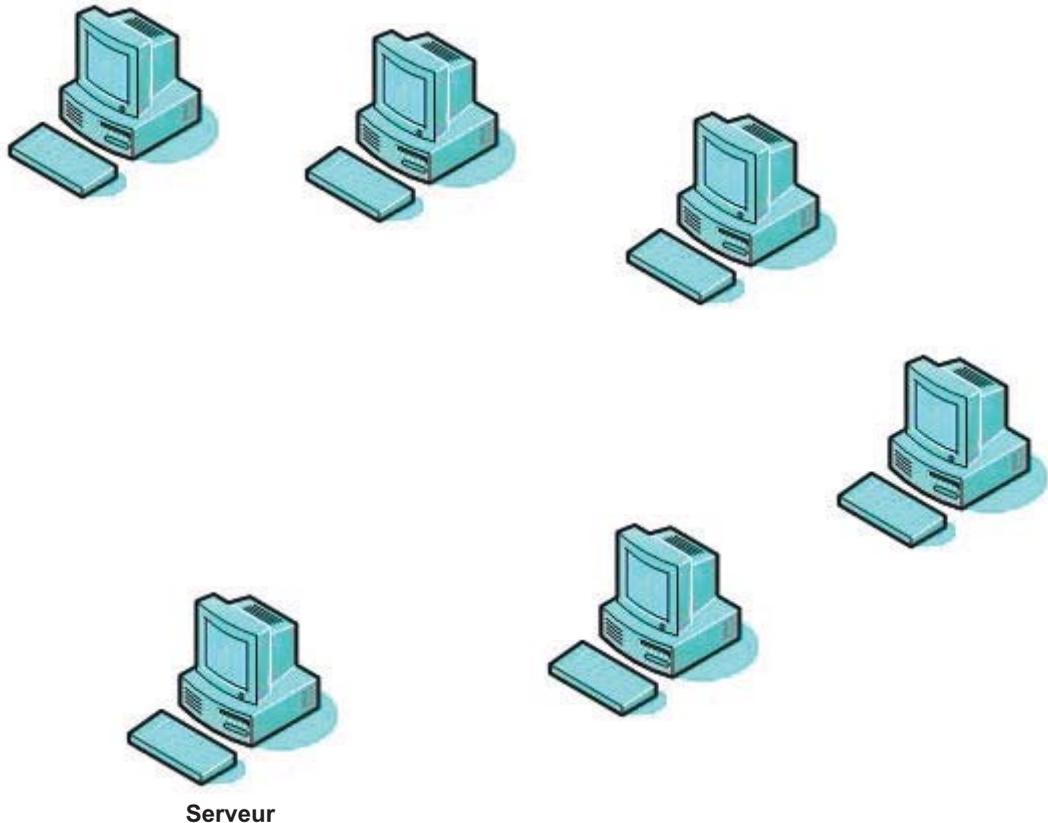


Fig.B-1 : Equipements informatique

- 1/ Décrivez les types de micro-ordinateurs présentés dans la figure B-1.
- 2/ Comment peut-on connecter les postes de travail au serveur ? Donnez les différents types de connexion.
- 3/ A votre avis, quels types de connectiques doit-on utiliser pour relier les postes clients au serveur ?
- 4/ Est-il possible d'utiliser une application installée dans le serveur par un utilisateur du réseau ?
- 5/ Comment peut-on sécuriser l'information dans un réseau ?

I. Présentation

Un réseau informatique permet de relier des ordinateurs pour qu'ils puissent s'échanger des données. Il est constitué d'au moins deux micro-ordinateurs liés ensemble par des supports de transmission. Il sert essentiellement à partager des ressources logicielles et matérielles entre les différents utilisateurs.

Un réseau informatique offre plusieurs services à ses utilisateurs. Parmi ces services, on peut citer :

- 1- l'accès aux fichiers et aux répertoires partagés (partage des données)
- 2- l'utilisation d'applications (partage d'applications)
- 3- l'utilisation d'une imprimante, d'un scanner, de modem, ... (partage de périphériques)

II. Les différents types

Il existe plusieurs types de réseaux classés soit :

- ✓ suivant la taille et la nature des réseaux
- ✓ suivant l'étendue et l'appartenance du réseau

1) Classement des réseaux suivant la taille :

Le tableau suivant présente les noms et les définitions de quelques types de réseaux classés suivant leurs tailles :

Type	Acronyme	Nom	Définition
Réseau local	LAN	Local Area Network	C'est un réseau qui ne dépasse pas environ les 2 kilomètres d'un bout à l'autre.
Réseau étendu	WAN	Wide Area Network	C'est un réseau qui s'étend sur plus de 2 kilomètres. Il est souvent formé de plusieurs LAN reliés ensemble.

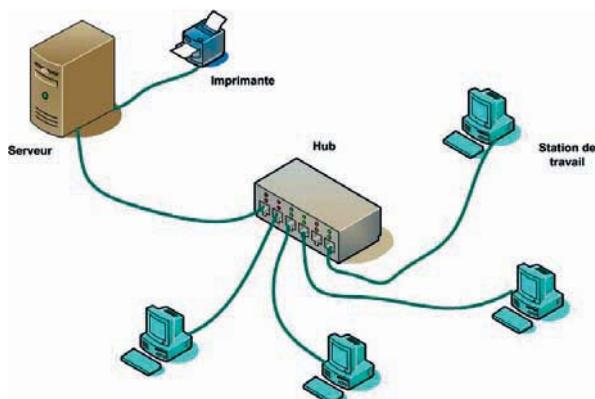


Fig. B-2 : Schéma d'un LAN

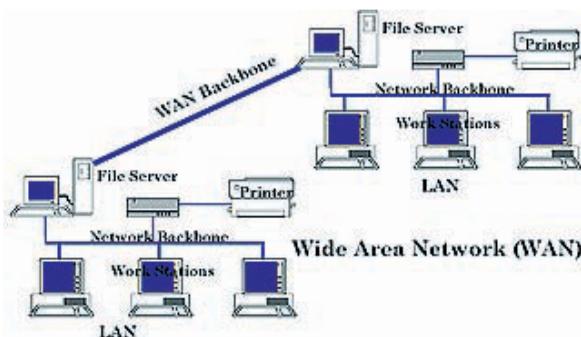


Fig. B-3 : Schéma d'un WAN

2) Le classement des réseaux suivant leurs appartenances :

Le tableau suivant présente les noms et les définitions de quelques types de réseaux classés suivant leurs appartenances :

Nom	Définition
Internet	<ul style="list-style-type: none"> – C'est un réseau à l'échelle mondiale formé par le regroupement de réseaux utilisant le protocole TCP/IP. – Les utilisateurs d'Internet n'appartiennent pas à la même société ou le même établissement ou organisme. – Internet est un exemple d'un WAN très large.
Intranet	<ul style="list-style-type: none"> – C'est un réseau qui appartient à une société ou à un établissement donné. – Intranet utilise la même technologie qu'Internet et surtout en ce qui concerne le protocole TCP/IP. – Intranet peut être implémenté sous forme de LAN ou de WAN.

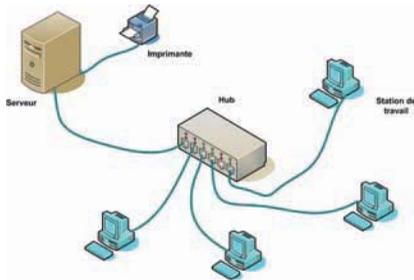


Fig. B-4 : Schéma d'un Intranet

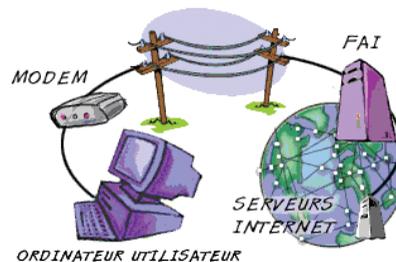


Fig. B-5 : Schéma d'Internet

III. Les avantages d'un réseau

Les réseaux informatiques présentent des avantages énormes au niveau de l'utilisation du système informatique ainsi qu'au niveau logiciel et matériel. Les avantages principaux sont :

1) Le partage des données :

L'avantage majeur de l'utilisation d'un réseau réside dans la possibilité de partager des données sous différentes formes (textes, tableau de calcul, fichiers de bases données, programmes,...). Les utilisateurs ont la possibilité d'utiliser simultanément les mêmes documents. Des règles existent pour empêcher l'exécution en même temps de quelques actions par différents utilisateurs (modification simultanée des mêmes données).

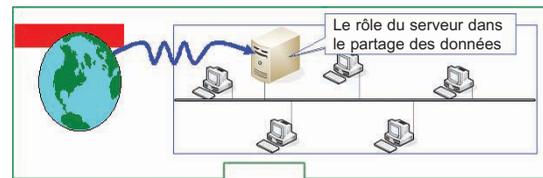


Fig. B-6 : Partage de données

2) Le partage des applications :

Les utilisateurs d'un réseau informatique utilisent un grand nombre d'applications qui coûtent souvent cher. De telles applications peuvent alors être installées sur le serveur et partagées. L'utilisateur aura le droit d'accéder à quelques fichiers seulement et l'administrateur réseau doit gérer les mises à jour de ces fichiers. Le coût sera réduit à cause de la réduction du nombre d'applications achetées.



Fig. B-7 : Partage d'applications

3) Le partage des périphériques :

Un réseau informatique permet de partager des ressources matérielles telles que imprimantes, scanners, modems, lecteur de CD, lecteur de DVD, ... Ce partage permet de diminuer les frais d'acquisition de ces équipements puisqu'un même périphérique peut être utilisé par plusieurs utilisateurs au lieu d'un périphérique par utilisateur. D'autre part, l'administrateur système est capable de mieux contrôler l'utilisation des différents périphériques partagés.



Fig.B-8 : Partage de périphériques

4) La sécurité :

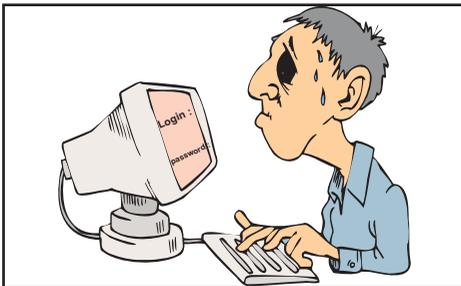


Fig. B-9 : La sécurité

Il existe plusieurs types ou niveaux de sécurité allant des plus simples : mot de passe au plus sophistiqués : utilisation des comptes utilisateurs en passant par l'inhibition de quelques périphériques (lecteur de disquettes, lecteur de CD,...)

L'utilisation du nom d'utilisateur (login) et du mot de passe (password) pour accéder à un réseau informatique est une technique très utilisée dans nos jours. Le système réseau refuse l'accès par l'utilisateur jusqu'à ce qu'il saisisse un nom d'utilisateur et un mot de passe valide.

IV. Les logistiques matérielles et logicielle

1) Introduction :

Les réseaux informatiques ont certains composants en commun quelque soit leur taille ou leur structure. Les principaux composants en commun sont les supports de transmission, les serveurs, les applications, ...

2) Logistique matérielle :

Un réseau informatique est composé matériellement par :

- **un serveur** : fournit les ressources partagées sur le réseau et permet aux clients d'accéder à ces informations
- **un ensemble d'ordinateurs** : appelés aussi stations de travail ou postes utilisateurs, ils permettent aux utilisateurs d'accéder aux ressources partagées.
- **une carte réseau** pour chaque ordinateur
- **des supports de transmission** : câbles, câbles à paires torsadées, câbles coaxiaux, fibres optiques, ...

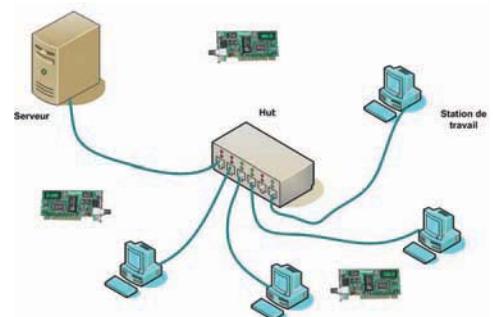


Fig. B-10 : Logistique matérielle

3) Logistique logicielle :

De point de vue logiciels, un réseau informatique comporte :

- Un protocole de communication
- Un système d'exploitation intégrant la fonction réseau

- Des applications d'exploitation réseau
- Les **services** qui appartiennent au système : partage de ressource, ...
- Les **logiciels** : bureautique, base de données, messagerie électronique, ...

V. Les topologies des réseaux

1) Introduction :

La topologie d'un réseau présente sa forme ou sa structure. Elle décrit la façon dont sont interconnectées les différentes entités qui le composent. Il existe deux types de topologies réseau :

- **Logique** : qui décrit le flux de données à travers un réseau.
- **Physique** : qui décrit l'apparence d'un réseau.

Dans ce qui suit, on présentera les trois topologies de réseaux les plus utilisées à savoir la topologie en étoile, en bus et en anneau.

2) Les topologies en étoile :

Activité

a/ Décrivez la topologie présentée dans la figure suivante.

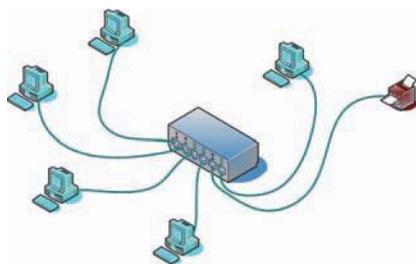


Fig. B-11 : Topologie en étoile

b/ A votre avis, quels sont les avantages et les inconvénients de l'utilisation de cette topologie ?

Dans cette topologie, chaque entité est reliée à un ordinateur central (ou serveur) dans une configuration en étoile. Le serveur reçoit directement les données des postes de travail et se charge de la transmission des signaux aux différents postes.

3) Les topologies en bus :

Activité

a/ Décrivez la topologie présentée dans la figure suivante.

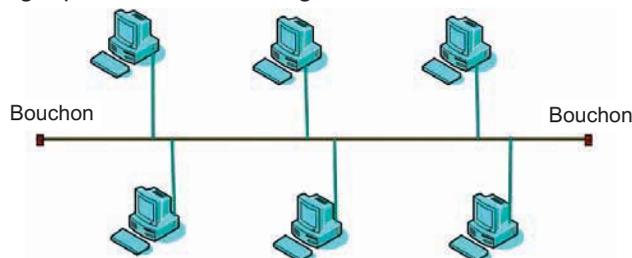


Fig. B-12 : Topologie en bus

b/ A votre avis, quels sont les avantages et les inconvénients de l'utilisation de cette topologie ?

4) Topologie en anneau :

Activité

a/ Décrivez la topologie présentée dans la figure suivante.

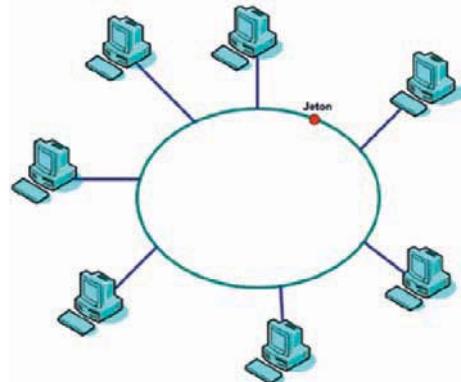


Fig. B-13 : Topologie en anneau

b/ A votre avis, quels sont les avantages et les inconvénients de l'utilisation de cette topologie ?

Dans cette topologie, l'ensemble des entités du réseau est raccordé en boucle fermée. Les micro-ordinateurs sont connectés directement à l'anneau ou indirectement par l'intermédiaire d'un appareil appelé unité d'accès multi station (en anglais, **Multi Station Access Unit, MSAU**). Les informations circulent dans un sens unique, d'une entité à la suivante.

5) Les avantages et les inconvénients :

Chacune des topologies exposées présente des avantages et des inconvénients au niveau de l'utilisation du réseau informatique. Le tableau suivant donne les principaux avantages et inconvénients des trois topologies.

Topologie	Avantages	Inconvénients
Etoile	<ul style="list-style-type: none"> – Facile à configurer – Facile à dépanner – Dès qu'une entité est défectueuse, elle est automatiquement isolée du réseau. 	<ul style="list-style-type: none"> – Une panne au niveau du serveur entraîne une panne du réseau – L'installation nécessite plus de câbles que les autres topologies.
Bus	<ul style="list-style-type: none"> – Facile à installer – L'installation nécessite moins de câbles comparée aux autres topologies. 	<ul style="list-style-type: none"> – Plus difficile à configurer – Difficile à dépanner car la panne peut se trouver n'importe où sur le câble – Dès qu'une entité est défectueuse, elle entraîne automatiquement une panne au niveau du réseau.
Anneau	<ul style="list-style-type: none"> – L'installation nécessite moins de câbles que la topologie en étoile mais plus que la topologie en bus. – Mise en place simple 	<ul style="list-style-type: none"> – Difficile à reconfigurer – La panne d'un support ou d'une entité peut affecter tout le réseau – Dépannage peut être difficile

Activité

Etablissez une comparaison des avantages et des inconvénients des différentes topologies.

VI. Exploitation d'un réseau local

Activité 1

- Activez la fenêtre relative au voisinage réseau (**Fig III.12**).
- Visualisez le réseau entier.
- Complétez cette phrase : *Les postes du laboratoire sont reliés en*

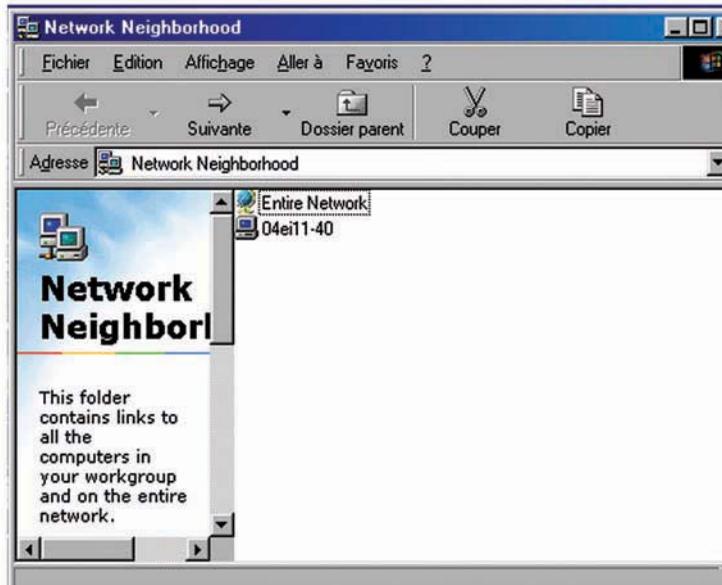


Fig. III.12 : Fenêtre relative au voisinage réseau

Activité 2

- Cliquez sur l'un des postes.
- Consultez son contenu.
- Choisissez un fichier et copiez le sur votre poste.
- Complétez cette phrase : *Sur un réseau, les documents sont*

Activité 3

- Choisissez un fichier texte.
- Demandez son impression.
- Complétez cette phrase : *Sur un réseau, les ressources sont partageables.*

Activité 4

- Activez le programme permettant la messagerie.
- Envoyez des messages à vos camarades.
- Complétez cette phrase : *Sur un réseau, les utilisateurs peuvent des messages.*

Activité 5

- Activez le logiciel du Net meeting ou un logiciel équivalent.
- Utilisez ce logiciel pour montrer à vos camarades le déroulement d'une tâche (création d'un dossier, copie d'un fichier, ...).
- Complétez cette phrase : *Le réseau est un bon outil*

VII - Retenons

- Un système d'exploitation est un ensemble de programmes nécessaires au lancement et à l'utilisation d'un ordinateur.
- Un système d'exploitation admet diverses fonctionnalités telles que : gestion du processeur, gestion de la mémoire vive, gestion des entrées/sorties et gestion des fichiers.
- Un fichier est un ensemble homogène d'informations. Un dossier peut contenir plusieurs fichiers et / ou dossiers.
- Un réseau informatique permet de relier des ordinateurs pour qu'ils puissent s'échanger des données, des applications et des périphériques.
- Il existe plusieurs topologies de réseau dont la topologie en étoile, en bus et en anneau.

EXERCICES

Exercice n° 1

Citez quelques exemples de systèmes d'exploitation et donner leurs caractéristiques. Formulez la réponse sous forme de tableau comparatif.

Exercice n° 2

Soit la liste des mots suivants :

fonctionnalités - démarrage - programmes - système - l'ordinateur

Complétez le texte suivant par les mots de la liste ci-dessus :

Au moment du de l'ordinateur et après avoir choisi le système d'exploitation à utiliser, cherchera l'emplacement de ce dernier pour le mettre en marche. Une partie du sera placée dans la RAM : c'est le noyau du système comportant les fonctionnalités les plus utilisées. D'autres disponibles dans le système d'exploitation existant sous forme de appelés utilitaires seront lancés dans le cas de besoin.

Exercice n° 3

Evaluez chacune des affirmations suivantes en écrivant, dans la case correspondante, «Valide» ou «invalide». En cas d'invalidité, corrigez l'affirmation en question.

Affirmation proposée	Valide / Invalide	Affirmation corrigée
Le système d'exploitation est formé par un programme unique.		
Un système d'exploitation permet de gérer des fichiers.		
Un utilisateur ne peut changer la forme de son Bureau de travail que s'il a activé un logiciel.		
Un réseau permet de partager des ressources matérielles.		
Un réseau permet de partager les ressources matérielles uniquement.		
La topologie réseau utilisée au laboratoire d'informatique du lycée est de type bus.		
Dans un environnement réseau, il est possible de partager l'utilisation du logiciel de traitement de textes.		

Exercice n° 4

- 1/ Lancez l'application " Bloc notes " et créez un fichier intitulé " F1 ".
- 2/ Créez un dossier intitulé " Test ".
- 3/ Enregistrez le fichier "F1" dans ce dossier.
- 4/ Renommez le fichier en " F2 ".
- 5/ Déplacer le fichier dans le dossier " Mes documents ".
- 6/ Supprimez le fichier " F2 " puis le dossier " Test ".

CHAPITRE 4

Éléments de multimédia

Objectifs

- Produire un document en utilisant un logiciel de traitement de texte.
- Réaliser des numérisations d'images, des retouches et des conversions sous différents formats.
- Numériser et réaliser des mixages de séquences sonores.
- Réaliser et manipuler des séquences vidéo.

Plan du chapitre

- I. Introduction
- II. Le traitement de texte
- III. L'image
- IV. Le son
- V. La vidéo
- VI. Retenons

Exercices

CHAPITRE 4

Éléments de multimédia

« Les possibilités se multiplient à mesure qu'on les saisit »

Sun tzu

I. Introduction

Activité 1

Le club d'informatique de votre lycée décide d'envoyer un groupe pour visiter la foire annuelle d'informatique de la région. Puisqu'on vous a chargé de faire un compte rendu de cette visite, quels sont les moyens que vous pensez utiliser pour élaborer un bon rapport faisant profiter aussi ceux qui ne sont pas allés ?

Aide

1. Il est clair que votre rapport peut prendre plusieurs formes. Vous pouvez rédiger un long texte décrivant en détail ce que vous jugez utile d'en faire part à vos amis. Si vous disposez d'un appareil photo et surtout numérique, vous allez pouvoir renforcer votre compte rendu avec de belles images. Et si par chance vous vous procurez une caméra vidéo, vous allez pouvoir réaliser un joli film que vous ferez voir aux adhérents du club. Vous voyez, comment avec quelques outils en plus, vous améliorerez vos supports de communication.
2. Discutez de ces trois formes de communication avec vos amis et définissez chaque support ainsi que l'apport de chacun d'eux. Remarquez que la première forme repose essentiellement sur du texte. La seconde ajoute des images. La troisième est composée de séquences vidéo, de paroles et éventuellement de morceaux de musique.
3. Chacune des ces différentes formes de communication est dite média. Et une forme comportant plusieurs médias est dit multimédia. On déduit que votre première forme composée uniquement d'un rapport texte repose sur un seul média. Quand vous lui ajoutez des images, vous l'avez rendu multimédia. Ce produit ou cette forme va comporter du texte et des images. Un film vidéo, si vous allez en faire un, est un produit multimédia. Bien entendu, avec l'avancée de l'informatique, on peut aller beaucoup plus loin dans ce domaine. En effet, dans un même document, on pourra associer du texte, du son, de la musique, de la parole, de l'image fixe ou animée et de la vidéo. Quant à la navigation dans le document, elle n'est plus forcément linéaire ; des mots ou des objets particuliers peuvent vous envoyer à d'autres documents. Il suffit de cliquer dessus avec la souris. Ces objets ont été associés à des liens. Ces derniers peuvent être internes à un même document ou externes et dans ce cas ils vous brancheront sur d'autres documents. On les appelle liens hypertextes ou hypermédias. Vous avez compris que de tels documents sont obligatoirement numériques et consultables sur ordinateur.

Vous savez que pour réaliser de tels produits, vous aurez besoin d'un ordinateur. Et selon ce que vous voulez utiliser comme média, des performances s'imposent à la machine et ses périphériques.

Activité 2

Citer des logiciels commerciaux et des logiciels libres permettant de traiter :

- du texte
- de l'image
- du son
- de la vidéo

A chaque média, donnez une configuration minimale vous permettant de réaliser votre travail.

Nous allons apprendre comment traiter chacun de ces médias à part.

II. Traitement de texte

II.1. Introduction :

Le concept est apparu en 1964 chez IBM, en Allemagne, avec l'expression *Textverarbeitung*. Il était en fait essentiellement utilisé par cette compagnie. L'expression a été reprise ensuite aux USA sous le terme *Word Processing*. L'expression est devenue **traitement de texte** en français.

Un logiciel de traitement de texte sert à rédiger des documents. Ces documents sont essentiellement formés de textes ; on y trouve des images, des tableaux des graphiques et même des sons. C'est pour cette raison que le mot document convient le mieux. Le logiciel de traitement de texte offre la grande facilité de corriger ou de changer le contenu d'un document en comparaison avec son ancêtre la machine à écrire. Quant aux images et aux graphiques, il n'y a aucune comparaison.

Regardons d'abord de quoi est constitué un texte.

II.2. La structure d'un texte

Un texte est composé de paragraphes. Les paragraphes sont constitués de phrases. Elles commencent dans une nouvelle ligne et se terminent par un retour à la ligne. Les phrases sont constituées de mots. Une phrase commence par une majuscule et se termine par un point. Les mots sont formés de lettres et séparés par des espaces.

Quand on écrit un texte dans une langue donnée, il est impératif de respecter l'orthographe et la grammaire de cette langue. Quelques règles typographiques sont à respecter lors de la saisie d'un texte. Les signes de ponctuation simples tels que la virgule, le point sont collés à la lettre qui les précède et doivent être suivis d'une espace. Quant aux signes de ponctuation double tels le point virgule, les deux points, les points d'exclamation et d'interrogation, ils doivent être précédés d'une espace insécable et suivis d'une espace simple. L'espace insécable empêchera le logiciel de traitement de texte de séparer ce signe de ponctuation du mot qu'il précède dans un retour à la ligne. On évitera ainsi l'erreur typographe qui consiste à commencer une ligne par un signe de ponctuation.

Notion de document

Comme nous allons le découvrir, les logiciels de traitements de texte ne traitent pas uniquement des textes. C'est pour cette raison qu'un produit réalisé avec un logiciel de traitement de texte est plutôt appelé document. Un document peut contenir du texte, des tableaux, des images, des graphiques etc.

Dans toute la suite, on appellera document, tout produit réalisé avec un logiciel de traitement de texte.

II.3. Les fonctionnalités de base d'un logiciel de traitement de texte

Afin de découvrir et de concrétiser les fonctionnalités de base d'un logiciel de traitement de texte, nous allons utiliser un exemple : le logiciel WORD.

1) Présentation du logiciel

Word est un logiciel de traitement de texte WYSIWYG (What you see is what you get). Vous allez donc pouvoir écrire vos textes facilement et leur donner la présentation que vous désirez. La présentation d'un texte imprimé nécessite en effet un minimum de techniques et de méthodes. Il ne faut jamais perdre de vue qu'un texte imprimé est d'abord fait pour être lu, ce qui implique qu'il faudra permettre une lecture aisée.

Activité 3

Exécutez le programme Word et découvrez les différentes composantes de la fenêtre qu'il affiche. Vous remarquerez que cette forme de présentation est pratiquement dans la majorité des logiciels que nous allons utiliser dans ce cours ou ceux que vous allez utiliser librement chez vous.

2) L'interface utilisateur de Word

Vous trouverez à la figure **Fig.IV.1** la fenêtre de WORD telle qu'elle se présente au démarrage.

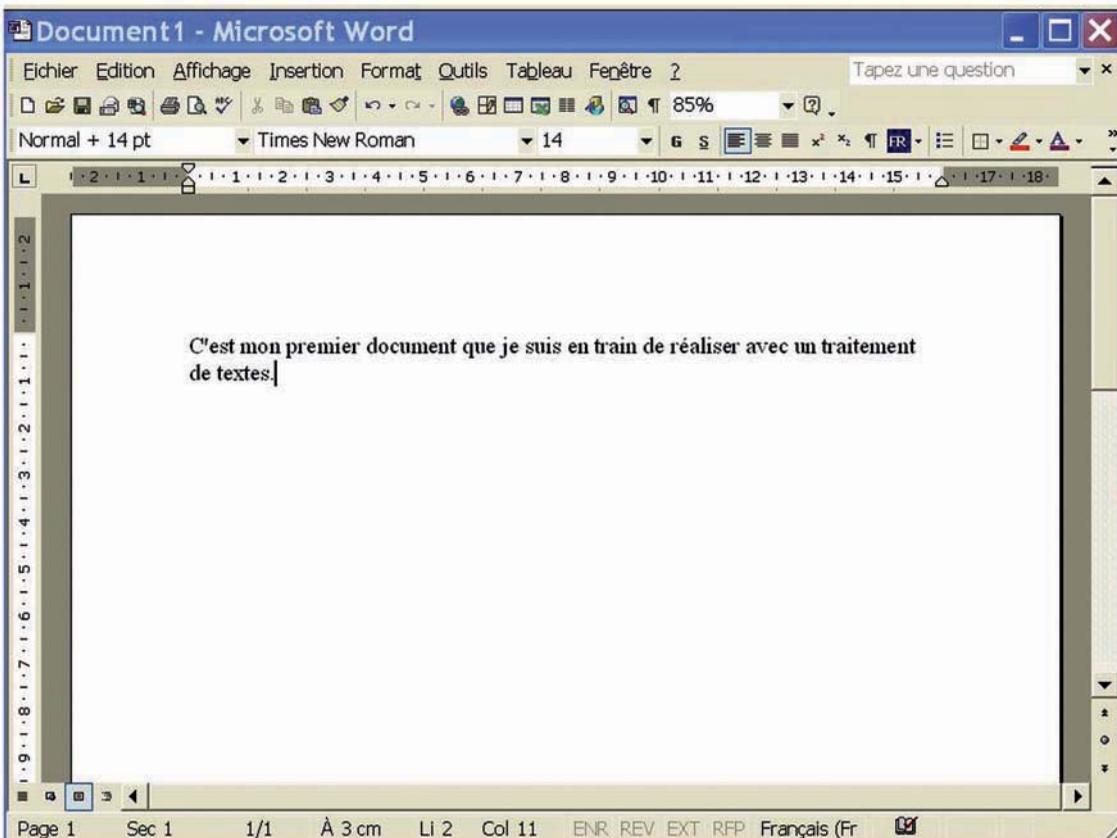


Fig.IV.1 : L'écran du WORD

Dans cette fenêtre vous pouvez distinguer plusieurs zones.

a) La barre de titre

Regardez la figure Fig.IV.2 ou lancez le logiciel Word. La fenêtre affichée contient, comme son nom l'indique, le titre de la fenêtre : Microsoft Word. Elle contient également le nom du document en cours d'édition (le fichier actuellement ouvert s'appelle Document1 ; ce nom est généré par le logiciel lui-même et si vous le changez pas, votre travail sera enregistré sous ce nom. Si vous ouvrez un deuxième document, le logiciel lui donnera comme nom Document 2 et ainsi de suite. Vous avez intérêt donc à commencer par donner un nom significatif à votre document. Nous verrons comment réaliser cette tâche à partir du menu Fichier.

La barre de titre contient également à droite le bouton de fermeture, () et le bouton des niveaux d'affichage de la fenêtre de travail () et le bouton de réduction à la barre de tâches ().

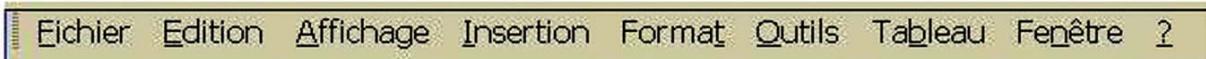
Fig.IV.2 : La barre de titre



b) La barre de menus

Cette barre contient toutes les commandes du logiciel Word. Elles sont classées par thèmes. Pour retrouver les commandes d'un thème il suffit de cliquer sur le nom du menu pour faire dérouler la liste. Vous n'avez qu'à vous déplacer dans cette dernière pour faire votre choix. Parfois le choix d'une commande lance une deuxième liste déroulante. Certaines commandes ouvrent des boîtes de dialogue pour vous permettre de faire vos choix. Vous reconnaîtrez très facilement les commandes directes, celles qui lancent une liste déroulante (suivie d'une flèche) et celles qui ouvrent une boîte de dialogue (suivi de trois points).

Fig.IV.3 : La barre de menus



c) La barre d'outils

Elle est formée d'un ensemble de boutons groupés par thèmes. Leurs apparitions et leurs dispositions sont personnalisables, c'est-à-dire que vous pouvez ajouter, enlever ou déplacer un ou plusieurs boutons. Ils forment des raccourcis aux commandes figurant dans les différents menus. Un bouton porte un dessin 3D signifiant son rôle. Quand vous faites dérouler un menu quelconque, vous allez découvrir les intitulés textuels des commandes et pour certaines d'usage assez fréquent, vous allez trouver un bouton à sa gauche et un raccourci clavier à sa droite. Ce bouton, s'il est placé dans la barre des outils, servira à lancer directement la commande en cliquant dessus par la souris.

Remarque

Pour accéder d'un menu ou lancer une commande, on peut utiliser les touches de raccourcis du clavier.



Fig.IV.4 : La barre d'outils

Word vous permet d'ajouter ou de retirer la barre d'outils d'un thème donné. Il suffit de cocher ou de décocher devant son nom dans la commande Barre d'outils du menu Affichage.

Exemple :

Dans le menu fichier, à la commande *Enregistrer* correspond le raccourci clavier **Ctrl + S**

Le raccourci clavier Ctrl + S signifie qu'on doit appuyer sur la touche Ctrl et la garder enfoncée, ensuite, on appuie normalement sur la touche S puis on relâche la touche Ctrl.



Fig.IV.5 : Exemples de raccourcis pour certaines commandes

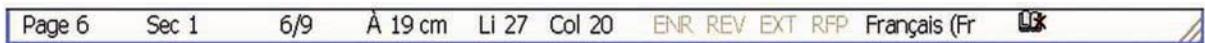
d) La zone d'édition du document à réaliser

C'est dans cette zone que vous allez saisir votre document et y apporter les modifications nécessaires. Vous remarquez que le logiciel provoque une marge en haut, une deuxième à gauche, une troisième à droite et une dernière en bas. Ces retraits sont pris par défaut par le logiciel. Nous verrons comment les modifier selon les besoins.

Deux barres de défilement, l'une verticale et l'autre horizontale, sont placées respectivement à droite et en bas de la zone d'édition. Elles permettent de se déplacer dans le document quand il déborde de la zone d'affichage. Vous pouvez faire apparaître les règles horizontale et verticale. Elles sont graduées à votre convenance (en centimètres par exemple) et vous permettent de faire un positionnement et un choix précis de tous vos objets dans le document. La règle horizontale comporte aussi les taquets de retraits et de tabulation. Ces taquets sont choisis en cliquant sur l'icône d'intersection des deux règles. L'icône visible par défaut est la tabulation gauche.

e) La barre d'état

Dans cette barre vous avez en permanence, l'emplacement du curseur dans le texte (ligne, colonne, page et section) et sa position par rapport au haut de la feuille. Elle sert aussi à afficher des messages et de lancer certaines actions. Par exemple, pour lancer les deux correcteurs orthographique et grammatical, il suffit de cliquer doublement sur le bouton de l'Etat Grammatical et



orthographique.

Activité 4

Fig.IV.6 : La barre d'état

Dans le lecteur D de votre disque dur, créez un dossier intitulé "BUREAUTIQUE". Dans ce dossier, créez deux sous dossiers intitulés respectivement "TEXTES" et "IMAGES". On se propose de mettre respectivement les deux dossiers créés comme dossiers par défaut pour les documents et pour les images. Allez au menu "Outils" puis à la commande "Options" ensuite cherchez l'onglet convenable de la boîte de dialogue affichée. Apportez les modifications souhaitées.

2) Saisie d'un texte

Activité 5

Saisir le texte suivant et l'enregistrer dans le sous dossier "TEXTES" du dossier "BUREAUTIQUE" sous le nom "Reseaux.doc".

Le réseau numérique tunisien

Le réseau national de transmission de données à haut débit tunisien a été mis en place en 1999 suite au démarrage et à la généralisation de l'Internet en 1996-1997.

Il est constitué de nœuds de transit repartis dans toutes les régions du pays. Essentiellement à base de câbles en fibre optique, il intègre le trafic téléphonique et le trafic Internet et multimédia. Depuis quelques années déjà, l'Internet est accessible grâce à 12 fournisseurs de services Internet (FSI). La bande passante est entre-temps passée de 76,5 Mbits/s à 155 Mbits/s pour desservir plus de 700.000 Internauts tunisiens (juin 2004) devant atteindre 850.000 à la fin de l'année 2004 et trois millions dans deux ans. La bande passante à l'internationale quant à elle est passée de 136 Mbits/s en 2003 à 223 Mbits/s en 2004 et portée à 380 Mbits/s à partir de juillet 2004.

En outre, la Tunisie jouit d'un réseau de télécommunications jugé parmi les plus avancés de la Méditerranée. Permettant un débit de 10 à 40 Gbits/s pour le réseau fixe et utilisant 7.000 km en fibres optiques, ce réseau est en train d'être prolongé à l'intérieur des régions par des réseaux régionaux adaptés et conçus pour se rapprocher et rapprocher au maximum les services des zones urbaines et des principaux sites d'activité.

Les statistiques affichées par l'Agence Tunisienne de l'Internet révèlent que la plus grande frange des utilisateurs n'est pas constituée d'entreprises quoique 40% des PME/PMI tunisiennes soient raccordées à l'Internet. L'éducation et l'enseignement supérieur exhibent à eux deux 68% des 102.412 abonnés recensés en juin 2004.

Les FSI privés, où on suppose que les entreprises s'approvisionnent leurs services Internet, n'en sont qu'à 16%, ce qui nous donne à peine plus de 16.000 abonnements. Ceux-ci comprennent aussi les utilisateurs individuels.

Les abonnements des entreprises comprennent bien entendu des prestations à divers services de transmission de données, X.25, ADSL (Asymmetric Digital Subscriber Line), RNIS (Réseau Numérique à Intégration de Services) et lignes spécialisées.

Mohamed LOUADI
ISG - TUNIS

Règles à suivre

- a** Ouvrir un nouveau document Word et l'enregistrer comme indiqué dans l'énoncé de l'activité. Au fur et à mesure que vous avancez dans votre saisie, faites l'enregistrement de votre document pour ne pas perdre les dernières modifications en cas de panne quelconque. Vous pouvez utiliser la commande "Enregistrer" du menu "Fichier" ou l'un de ses deux raccourcis.
- b** Commencer la saisie de votre texte sans se soucier de sa mise en forme. Le retour à la ligne est fait automatiquement par le logiciel lui-même. Dans le cas où vous voulez commencer un nouveau paragraphe, faites un retour à la ligne en appuyant sur la touche "ENTREE". Pour le moment, ne vous souciez pas des lignes ondulées en rouge ou en vert que Word met en dessous de quelques mots ou quelques phrases. Il vous avertit qu'il y a une erreur. Nous verrons plus loin comment s'y prendre.
- c** Les caractères s'insèrent toujours à la position du curseur. Par conséquent si vous avez oublié un mot ou un caractère quelconques, utilisez le clavier ou plus facilement la souris pour positionner le curseur à la place voulue puis procédez à l'insertion des caractères oubliés. Si vous désirez écraser ce qui est déjà écrit vous devez balancer au mode Refrappe en cliquant sur la touche "Inser" du clavier. Un deuxième appui sur la même touche "Inser" vous fait revenir au mode "Insertion".
- d** Parfois les caractères que vous voulez écrire ne se trouvent pas sur les clavier, il faudra donc lancer la commande "Caractères spéciaux" du menu "Insertion". Une fenêtre de dialogue s'affiche et vous pouvez choisir dans les champs qu'elle propose, la police et le caractère désirés puis appuyer sur la bouton "Insérer" avant de fermer cette fenêtre.
- e** L'action de sélection est très importante et il existe plusieurs modalités. On pourra utiliser la souris ou le clavier pour sélectionner des parties du document et si on veut tout sélectionner on pourra utiliser la commande "Tout sélectionner" du menu "Edition".

Activité 6

Lancez de nouveau le logiciel Word (Quittez le s'il est déjà ouvert). Ouvrez le document de l'activité précédente et corriger les éventuelles erreurs d'orthographe et de grammaire.

3) La correction orthographique et grammaticale

Le logiciel Word est multilingue et dispose pour chaque langue les dictionnaires appropriés pour pouvoir assurer les corrections orthographiques et grammaticales. Vous pouvez choisir la langue d'écriture de votre document et autoriser Word à corriger les fautes d'orthographe et de grammaire. En effet, La commande "Langue" du menu "Outils" vous fait accéder à une boîte de dialogue vous permettant de choisir la langue et la correction automatique de l'orthographe et la grammaire.

Il vous est possible d'ajouter beaucoup plus de précision à la façon avec laquelle Word détectera ces erreurs. Ces options sont accessibles dans la boîte de dialogue ouverte par la commande "Options" du menu "Outils". Il suffit d'appuyer sur l'onglet "Grammaire et orthographe".

Une fois cette opération terminée, nous allons passer à la mise en page et la mise en forme. Sachez que si vous comptez imprimer votre document, il vous est utile d'avoir une idée sur les dimensions du papier à utiliser ainsi que la zone de textes à choisir. Il est préférable de commencer par ce choix pour éviter de refaire la mise en forme.

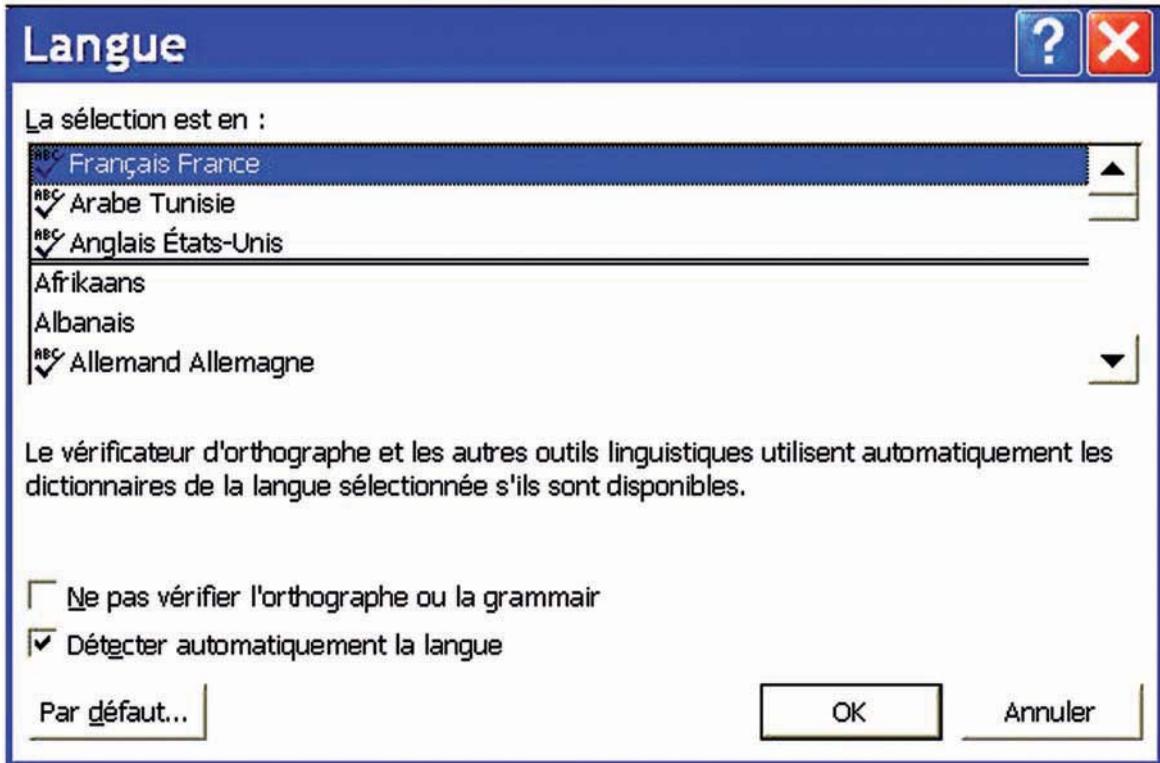


Fig.IV.7 : La boite de dialogue pour le choix de ma langue

4) Mise en page

Activité 7

Ouvrez votre dernier document et lancez la commande "Mise en page" du menu "Fichier". Explorez à travers les trois onglets de la boite, les différents choix de mise en page. Faites en un choix et réalisez un aperçu avant impression pour voir à quoi ressemble votre document une fois imprimé.

Retenir que :

- a A4, A5, B4, B5 US Letter etc. sont des normes standards utilisées dans la fabrication des feuilles d'imprimerie. Le format A4 est le plus utilisé dans notre pays. Les imprimantes sont souvent alimentées par des feuilles A4.
- b Une fois le format choisi, il existe deux manières de disposer la feuille pour l'écriture : verticalement (en portrait) ou horizontalement (en paysage).
- c La zone de texte est déterminée en fonction des quatre marges : de haut, de bas, de gauche et de droite. Si vous allez imprimer en recto verso et réaliser une reliure, les marges de gauche et de droite deviennent marge extérieure et marge intérieure.
- d Si vous comptez ajouter des en-têtes et des pieds de page, indiquez les marges désirées en tenant compte de ce que vous avez choisi comme marge de haut et marge de bas.
- e Il vous est possible à ce stade de définir des bordures à vos pages et des trames de fond.

5) Mise en forme

Activité 8

Reprenez le document "Reseaux.doc" et mettez-le sous la forme suivante :

Le réseau numérique tunisien

Le réseau national de transmission de données à haut débit tunisien a été mis en place en 1999 suite au démarrage et à la généralisation de l'Internet en 1996-1997.

Il est constitué de nœuds de transit repartis dans toutes les régions du pays. Essentiellement à base de câbles en fibre optique, il intègre le trafic téléphonique et le trafic Internet et multimédia. Depuis quelques années déjà, l'Internet est accessible grâce à 12 fournisseurs de services Internet (FSI). La bande passante est entre-temps passée de 76,5 Mbits/s à 155 Mbits/s pour desservir plus de 700.000 Internaute tunisiens (juin 2004) devant atteindre 850.000 à la fin de l'année 2004 et trois millions dans deux ans. La bande passante à l'internationale quant à elle est passée de 136 Mbits/s en 2003 à 223 Mbits/s en 2004 et portée à 380 Mbits/s à partir de juillet 2004.

En outre, la Tunisie jouit d'un réseau de télécommunications jugé parmi les plus avancés de la Méditerranée. Permettant un débit de 10 à 40 Gbits/s pour le réseau fixe et utilisant 7.000 km en fibres optiques, ce réseau est en train d'être prolongé à l'intérieur des régions par des réseaux régionaux adaptés et conçus pour se rapprocher et rapprocher au maximum les services des zones urbaines et des principaux sites d'activité.

Les statistiques affichées par l'Agence Tunisienne de l'Internet révèlent que la plus grande frange des utilisateurs n'est pas constituée d'entreprises quoique 40% des PME/PMI tunisiennes soient raccordées à l'Internet. L'éducation et l'enseignement supérieur exhibent à eux deux 68% des 102.412 abonnés recensés en juin 2004.

Les FSI privés, où on suppose que les entreprises s'approvisionnent leurs services Internet, n'en sont qu'à 16%, ce qui nous donne à peine plus de 16.000 abonnements. Ceux-ci comprennent aussi les utilisateurs individuels.

Les abonnements des entreprises comprennent bien entendu des prestations à divers services de transmission de données, X.25, ADSL (Asymmetric Digital Subscriber Line), RNIS (Réseau Numérique à Intégration de Services) et lignes spécialisées.

Mohamed LOUADI
ISG - TUNIS

Remarquez que :

- ✓ Le titre est centré, gras et de taille plus grande que le reste du texte.
- ✓ Les paragraphes sont justifiés et suffisamment espacés.
- ✓ Chaque paragraphe marque un retrait à droite de sa première ligne.

Regardons de près comment faire quelques mises en forme courantes.

a) La mise en forme des caractères

Word dispose de plusieurs polices de caractères variées. L'utilisateur peut en supprimer et en ajouter d'autres. Cette opération se fait au niveau du système d'exploitation. Certaines polices sont avec empattements et la plus connue est la police Times New Roman tandis que d'autres sont sans empattement comme la police Arial. D'autres polices sont clairement fantaisistes. A chaque police, on peut associer une taille mesurée en points et un style qui peut être gras, italique ou souligné. Il est possible de choisir la couleur des caractères et du fond.

Notons que pour appliquer une mise en forme à une suite de caractères, il faudra d'abord la sélectionner.

Exemples :

Cette ligne est écrite en Times New Roman, une police avec empattement.

Cette ligne est écrite en Arial, une police sans empattement.

Cette ligne est écrite en une police fantaisiste.

Cette phrase est écrite en Tahoma style gras italique.

Ces mots sont en Times New Roman souligné.

Dans cette expression mathématique, 2 est en exposant et n est en indice : $5.u_n + (u_n)^2 = 0$

~~Cette phrase est barrée. Celle-ci est doublement barrée.~~

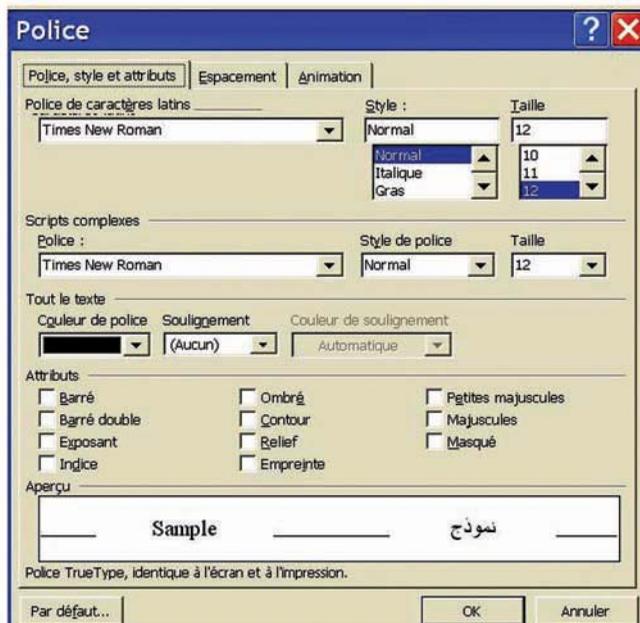


Fig.IV.8 : La boîte de dialogue de la commande Police

Pour changer la police de caractères à une partie donnée du document, commencez d'abord par la sélectionner. Ensuite, vous pouvez utiliser le menu "Format" puis la commande "Police" qui ouvre une boîte de dialogue permettant de choisir beaucoup d'options telles que la police, la taille, la couleur, le style, l'espacement etc.

Vous pouvez accéder à plusieurs de ces options directement de la barre d'outils. Deux listes déroulantes permettent respectivement le choix de la police et de sa taille. Le bouton couleur de police permet de faire un choix parmi une large palette de couleurs.

Autres exemples de mises en forme d'un texte :

Cette phrase est écrite en **Arial** de taille 16 points. Le nom de la police est écrit en rouge et cette deuxième phrase est surlignée en bleu.

Exercice

Reproduisez le document suivant :

La présentation des écrans

Lorsque l'écran ne correspond qu'à une seule fenêtre, on considère que les dessins d'écrans devraient être le plus simple possible, sans informations inutiles, et d'une densité moyenne d'environ les deux tiers de la capacité de l'écran.

On peut les structurer en trois parties :

- l'en-tête qui comporte les éléments de référence du travail en cours et parfois des messages d'aide,
- le corps qui contient l'essentiel de l'objet texte sur lequel travaille l'opérateur,
- et en pied de page comprenant les messages d'erreur et éventuellement le statut du dialogue entre la machine et l'écran

b) La mise en forme des paragraphes

Beaucoup de mises en forme sont associées aux paragraphes. Nous en citons :

1. le retrait de la première ligne ou alinéa,
2. la disposition du texte : aligné à gauche, aligné à droite, centré ou justifié,
3. les espaces avant et après les paragraphes,
4. la façon de couper un paragraphe ou de séparer des paragraphes.

Un exemple de **retrait positif** de la première ligne.

Un texte est composé de paragraphes. Les paragraphes sont constitués de phrases. Elles commencent dans une nouvelle ligne et se terminent par un retour à la ligne. Les phrases sont constituées de mots. Une phrase commence par une majuscule et se termine par un point. Les mots sont formés de lettres et séparés par des espaces.

Un exemple de **retrait négatif** de la première ligne.

Un texte est composé de paragraphes. Les paragraphes sont constitués de phrases. Elles commencent dans une nouvelle ligne et se terminent par un retour à la ligne. Les phrases sont constituées de mots. Une phrase commence par une majuscule et se termine par un point. Les mots sont formés de lettres et séparés par des espaces.

Un exemple de paragraphe **aligné à gauche**.

Un texte est composé de paragraphes. Les paragraphes sont constitués de phrases. Elles commencent dans une nouvelle ligne et se terminent par un retour à la ligne. Les phrases sont constituées de mots. Une phrase commence par une majuscule et se termine par un point. Les mots sont formés de lettres et séparés par des espaces.

Un exemple de paragraphe **aligné à droite**.

Un texte est composé de paragraphes. Les paragraphes sont constitués de phrases. Elles commencent dans une nouvelle ligne et se terminent par un retour à la ligne. Les phrases sont constituées de mots. Une phrase commence par une majuscule et se termine par un point. Les mots sont formés de lettres et séparés par des espaces.

Un exemple de paragraphe **centré**.

Un texte est composé de paragraphes. Les paragraphes sont constitués de phrases. Elles commencent dans une nouvelle ligne et se terminent par un retour à la ligne. Les phrases sont constituées de mots. Une phrase commence par une majuscule et se termine par un point. Les mots sont formés de lettres et séparés par des espaces.

Un exemple de paragraphe **justifié**.

Un texte est composé de paragraphes. Les paragraphes sont constitués de phrases. Elles commencent dans une nouvelle ligne et se terminent par un retour à la ligne. Les phrases sont constituées de mots. Une phrase commence par une majuscule et se termine par un point. Les mots sont formés de lettres et séparés par des espaces.

Activité 9

Ouvrez le document "Reseaux.doc" et mettez en forme ces paragraphes comme indiqués ci-dessous.

Le réseau numérique tunisien

Le réseau national de transmission de données à haut débit tunisien a été mis en place en 1999 suite au démarrage et à la généralisation de l'Internet en 1996-1997.

Il est constitué de nœuds de transit repartis dans toutes les régions du pays. Essentiellement à base de câbles en fibre optique, il intègre le trafic téléphonique et le trafic Internet et multimédia. Depuis quelques années déjà, l'Internet est accessible grâce à 12 fournisseurs de services Internet (FSI). La bande passante est entre-temps passée de 76,5 Mbits/s à 155 Mbits/s pour desservir plus de 700.000 internautes tunisiens (juin 2004) devant atteindre 850.000 à la fin de l'année 2004 et trois millions dans deux ans. La bande passante à l'internationale quant à elle est passée de 136 Mbits/s en 2003 à 223 Mbits/s en 2004 et portée à 380 Mbits/s à partir de juillet 2004.

En outre, la Tunisie jouit d'un réseau de télécommunications jugé parmi les plus avancés de la Méditerranée. Permettant un débit de 10 à 40 Gbits/s pour le réseau fixe et utilisant 7.000 km en fibres optiques, ce réseau est en train d'être prolongé à l'intérieur des régions par des réseaux régionaux adaptés et conçus pour se rapprocher et rapprocher au maximum les services des zones urbaines et des principaux sites d'activité.

Les statistiques affichées par l'Agence Tunisienne de l'Internet révèlent que la plus grande frange des utilisateurs n'est pas constituée d'entreprises quoique 40% des PME/PMI tunisiennes soient raccordées à l'Internet. L'éducation et l'enseignement supérieur exhibent à eux deux 68% des 102.412 abonnés recensés en juin 2004.

Les FSI privés, où on suppose que les entreprises s'approvisionnent leurs services Internet, n'en sont qu'à 16%, ce qui nous donne à peine plus de 16.000 abonnements. Ceux-ci comprennent aussi les utilisateurs individuels.

Les abonnements des entreprises comprennent bien entendu des prestations à divers services de transmission de données, X.25, ADSL (Asymmetric Digital Subscriber Line), RNIS (Réseau Numérique à Intégration de Services) et lignes spécialisées.

Mohamed LOUADI
ISG - TUNIS

Retenez que pour mettre en forme un paragraphe, soit qu'on utilise la boîte de dialogue que fournit la commande "Paragraphe" du menu "Format" soit les boutons appropriés de la barre d'outils.

c) La manipulation de blocs

Il est fréquent d'avoir besoin de faire les actions suivantes :

- Effacer un bloc,
- Copier un bloc,
- Remplacer un bloc,
- Chercher un mot ou un tout un bloc

Toutes ces actions sont accessibles soit du menu "Edition" puis la commande voulue soit à l'aide des boutons de la barre d'outils ou des raccourcis clavier associés.

Par exemple, si on désire déplacer un paragraphe à un autre emplacement, on le sélectionne puis on lance la commande "Couper", soit directement du menu "Edition", soit à l'aide du bouton "Ciseaux", soit à l'aide du raccourci clavier associé. Ensuite, on place le curseur au début de l'emplacement désiré et on lance la commande "Coller" du menu "Edition" ou l'un de ses raccourcis.

Activité 10

- Reprenez le document "Reseaux.doc" et réalisez les opérations suivantes :
- Permutez les deux derniers paragraphes du document.
 - Remplacez chaque mot "FSI" du document par le mot anglais "providers".

Aide

Pour déplacer un bloc,

1. on le sélectionne,
2. on lance la commande "Couper" en utilisant le menu "Edition" ou l'un de ses raccourcis,
3. on met le curseur à la position d'insertion désirée,
4. on lance la commande "Coller" du menu "Edition" en utilisant le menu "Edition" ou l'un de ses raccourcis.

Pour remplacer un bloc par un autre,

1. on lance la commande "Remplacer" du menu "Edition"
2. dans la boîte de dialogue, on met le bloc à remplacer dans le champ "Rechercher" et le nouveau bloc dans le champ "Remplacer par"
3. on appuie sur le bouton "Remplacer" si on veut voir tous les cas de remplacement ou sur le bouton "Remplacer tout" si on veut laisser faire tous les remplacements d'un seul coup.

d) L'insertion d'objets

Tous les logiciels de traitement de texte vont au-delà de la manipulation des textes. En effet, ils permettent des actions beaucoup plus importantes pour réaliser des documents attrayants. On peut ajouter des tableaux, des graphiques, des images et même des sons. Ces objets peuvent être créés par le logiciel lui-même ou réalisés par d'autres applications et importés. En plus de l'ajout, l'application de traitement de texte permet de faire quelques mises en forme sur ces objets.

Activité 11

Cherchez dans les dossiers de votre ordinateur les fichiers images. Sélectionnez-en quelques unes et prenez-en une copie que vous placerez dans le dossier "IMAGES"
Dans un nouveauté document intitulé "Objets_divers" que vous enregistrerez dans le dossier "TEXTES", copiez le document suivant :

Le multimédia et l'enseignement

Il est essentiel de comprendre que le professeur sera toujours partie, intégrante d'une formation multimédia. On a claironné la mise à mort prochaine de l'enseignement par le multimédia. En fait ce dernier se contente de renforcer les techniques pédagogiques usuelles. Un bon professeur a plus de valeur que n'importe quelle quantité d'équipements, si perfectionnés soient-ils.



Fig.IV.9 : Enseignement et multimédia

1) L'insertion d'une image

Pour insérer une image dans un document

1. Lancer la commande "Image" du menu "Insérer"
2. Dans la liste déroulante correspondant à la commande "Image", choisir la source de cette image. Choisir parmi les sources proposées. L'insertion de l'image choisie se fera à la place du curseur.
3. Pour modifier la position du texte aux alentours de l'image, lancer la commande "Habillage". L'accès à cette commande se fera en cliquant sur l'image avec le bouton droit de la souris ou en utilisant la barre d'outils "Image". On pourra afficher cette barre en lançant la commande "Barre d'outils" du menu "Affichage". Il suffit de cocher devant "Image". On vous laisse le soin d'explorer les actions associées aux différents boutons de cette barre.

N.B. : Il est possible de créer des dessins à l'aide du logiciel et de les insérer dans votre document.

2) Insertion d'un tableau

Pour insérer un tableau dans un document, il suffit de lancer la commande "Insérer" puis "Tableau" du menu "Tableau".

Plusieurs autres commandes sont proposées dans ce menu : le dessin direct d'un tableau, l'ajout ou la suppression de ligne ou de colonnes pour un tableau existant, la modification de la largeur d'une colonne ou de la hauteur d'une ligne, la mise en forme à l'intérieur des cellules d'un tableau, l'habillage d'un tableau par le texte l'entourant, etc.

Exercice

Reproduisez le document suivant et l'enregistrer dans le dossier "BUREAUTIQUE".

Le multimédia et l'enseignement

Il est essentiel de comprendre que le professeur sera toujours partie, intégrante d'une formation multimédia. On a claironné la mise à mort prochaine de l'enseignement par le multimédia. En fait ce dernier se contente de renforcer les techniques pédagogiques usuelles. Un bon professeur a plus de valeur que n'importe quelle quantité d'équipements, si perfectionnés soient-ils.

Equipement	Caractéristiques	Rôle
Haut parleur	Courbe de réponse	Sortie du son
Microphone	Direction	Entrée du son
Logiciels	Spécialisé audio	Traitement du son

Remarques:

1. La couleur de fond des cellules se fait à l'aide de la commande "Bordure et trame " du menu "Format"
2. Pour modifier l'habillage du tableau, il suffit de cliquer dessus à l'aide du bouton droit de la souris. Vous retrouverez la commande "Propriétés" en fin de liste.

III. L'image

III.1. L'image fixe

Nous avons vu dans la partie précédente comment ajouter une image à un texte. Nous avons vu aussi qu'au moment de l'insertion d'une image, le logiciel de traitement de texte nous offre beaucoup de ressources. L'une des ressources est la numérisation d'une image en utilisant un scanner ou un appareil photo numérique. Parlons un peu de la numérisation.

La numérisation d'une image consiste à transformer son aspect analogique en une forme numérique.

Les appareils qui font la numérisation d'images sont les scanners, les appareils photos numériques, les webcams, les caméscopes.



Fig.IV.10 : L'appareil photos numériques

Le principe de scanner est le suivant :

Le noir absorbe la lumière, le blanc la reflète et les autres couleurs font ce travail à des degrés variés. Ces actions sont transformées en valeurs numériques pour chaque point de l'image. Le nombre de points par unités de mesure et la grandeur de l'information rapportée sur ce point définissent la résolution de la numérisation. Vous pouvez agir sur ses paramètres et les modifier selon les performances de votre scanner ou tout autre appareil numériseur.

L'image obtenue est maintenant un fichier numérique. Elle peut être traitée par des logiciels de traitement d'images. Il est possible de lui apporter plusieurs transformations et d'en modifier la façon de numérisation ou encore de compression. Chaque format a ses particularités et influe donc sur la qualité de l'image et la taille du fichier. La transformation d'une image dépend de ce qu'on veut avoir et de ce qu'on veut en faire. Au moment de son affichage, une image peut être soit représentée soit calculée. Dans le premier cas on dit qu'elle est de type **bitmap** et dans le second cas on dit qu'elle est **vectorielle**. Il existe plusieurs formats de compression, nous en citons les plus fréquemment utilisées.

Format	Acronyme de :	Caractéristiques
BMP	Windows Bitmap	Pas de compression et donne un fichier assez volumineux
JPEG	Joint Photographic Experts Group	Compression basée sur un algorithme de compression éliminant les parties redondantes de l'image. Le taux est personnalisable. La compression fait perdre un peu de la qualité de l'image mais fait gagner énormément sur la taille du fichier associé.
GIF	Graphics Interchange Format	Compression réduisant l'image à moins de 256 couleurs mais rendant faible la taille de l'image.
TIFF	Tagged Image File Format	Format conservant l'image avec ses niveaux de gris. Il donne un fichier relativement volumineux

Activité 12

Lancez le logiciel de numérisation disponible dans le laboratoire et faites votre choix pour scanner une image. Faites des numérisations sous différentes options et comparez les rendus obtenus.

Aide

1. Choisir une résolution faible puis moyenne puis élevée.
2. Choisir un mode noir et blanc, nuances de gris ou couleur.
3. Manipulez les réglages des couleurs.

Activité 13

Lancez un logiciel de traitement d'images.

- 1/ Insérez dans le nouveau document, l'image que vous avez scannée.
- 2/ Modifiez la teinte des couleurs,
- 3/ Insérez dessus d'autres images,
- 4/ Ajoutez un titre à cette image,
- 5/ Enregistrez ce document image successivement sous les formes : BMP, JPG, GIF et TIFF.
- 6/ Comparer les différents formats.

III.2. Les images animées

Activité 14

- 1/ Réalisez un cadran d'une montre analogique sans les aiguilles et l'enregistrez sous le nom "Cadran"
- 2/ Faites en 12 copies et enregistrez-les sous les noms **Cadran_1**, **Cadran_2**, ..., **Cadran_12**.
- 3/ Dans l'image **Cadran_i**, ajoutez l'aiguille des minutes en face de l'heure i.
- 4/ Utilisez un logiciel d'animation d'images pour générer avec ces cadrans une animation de l'aiguille.

Exemples :

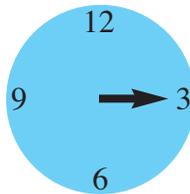


Fig.IV.11 : Cadran_3

Activité 15

- 1/ Dessinez deux ordinateurs
- 2/ Réalisez à l'aide d'un logiciel d'animation un dessin animé montrant le transfert de données entre les deux appareils.

IV. Le son

D'après la définition donnée dans Le Petit Larousse, le son est "la sensation auditive engendrée par une onde acoustique". En effet, tout corps animé d'un mouvement vibratoire émet un son. Le son se propage sous forme d'ondes avec une vitesse dépendante du milieu de propagation. Dans l'air, la vitesse du son est de l'ordre de 340 mètres par seconde. Elle est plus rapide dans l'eau (1425 mètres par seconde) et encore plus rapide dans l'acier (5000 mètres par seconde). La sensation auditive d'un son dépend de la fréquence de l'onde associée et de son amplitude ou volume.

IV.1. Les caractéristiques d'un son

Un son est caractérisé par sa fréquence, son volume et son timbre.

IV.1.1. La fréquence ou la hauteur

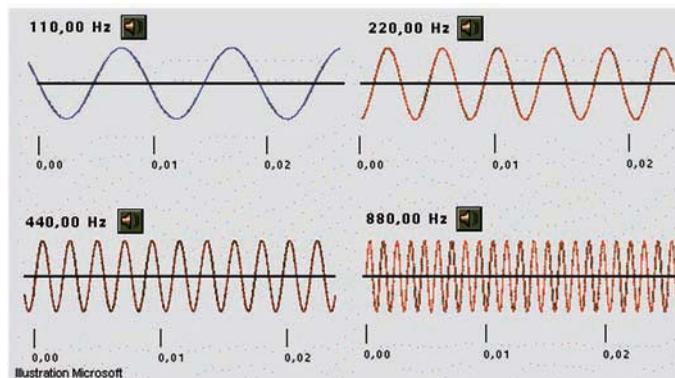


Fig.IV. 12 : Etalonnage de quatre sons

La fréquence traduit le nombre de vibrations par seconde. Elle se mesure en hertz (Hz). Un son de petite fréquence est dit grave ; quand sa fréquence est élevée il est dit aigu. Entre les deux, le son est dit médium.

Les égaliseurs que vous manipulez dans vos magnétophones ou dans les logiciels de lecture multimédia sont des filtres qui selon vos désirs vous permettent d'accentuer ou d'éliminer certaines bandes : larges, médium ou aiguës. Parfois, la partition en bandes est beaucoup plus élaborée et permet de faire la manipulation de plusieurs intervalles de la bande audible. La figure suivante montre l'interface graphique d'un égaliseur fourni par un logiciel de lecture multimédia.



Fig.IV. 13 : Equaliseur graphique d'une application de lecture multimédia

Sachez que l'être humain n'entend que les sons ayant des fréquences comprises entre 20 Hz et 20 000 Hz. Cette bande de fréquence est dite bande **audibles**.

IV.1.2. Le volume ou l'amplitude

Le volume traduit l'intensité avec laquelle est émise l'onde correspondante à ce son. Elle est mesurée en décibels. Nous vous présentons dans la figure ci-dessous une grille traduisant cette mesure pour des sons ou des bruits que vous connaissez.

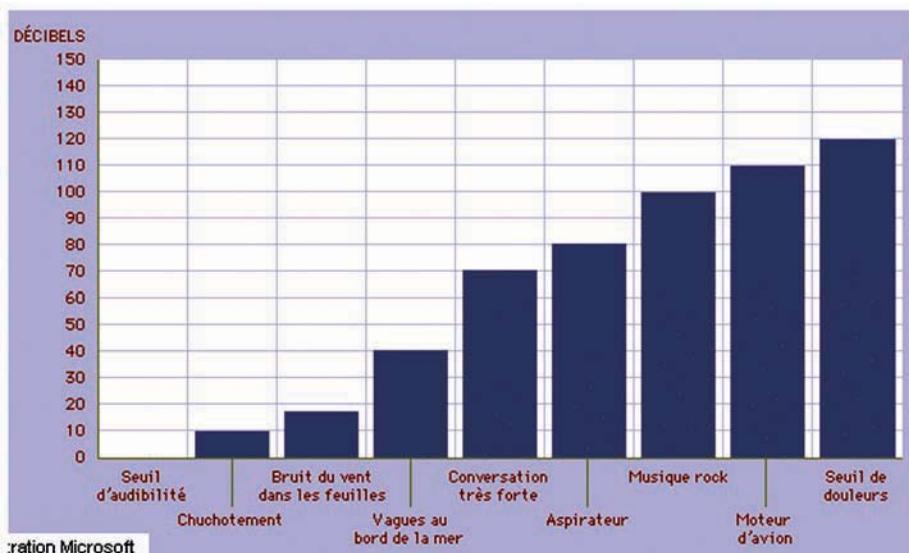


Fig.IV. 14 : Valeur en décibels de l'amplitude ou le volume de quelques sons et bruits. (Source : Microsoft)

Pour mieux comprendre la notion de volume, nous présentons ci-dessous un graphique comportant trois amplitudes différentes pour un même son ou onde.

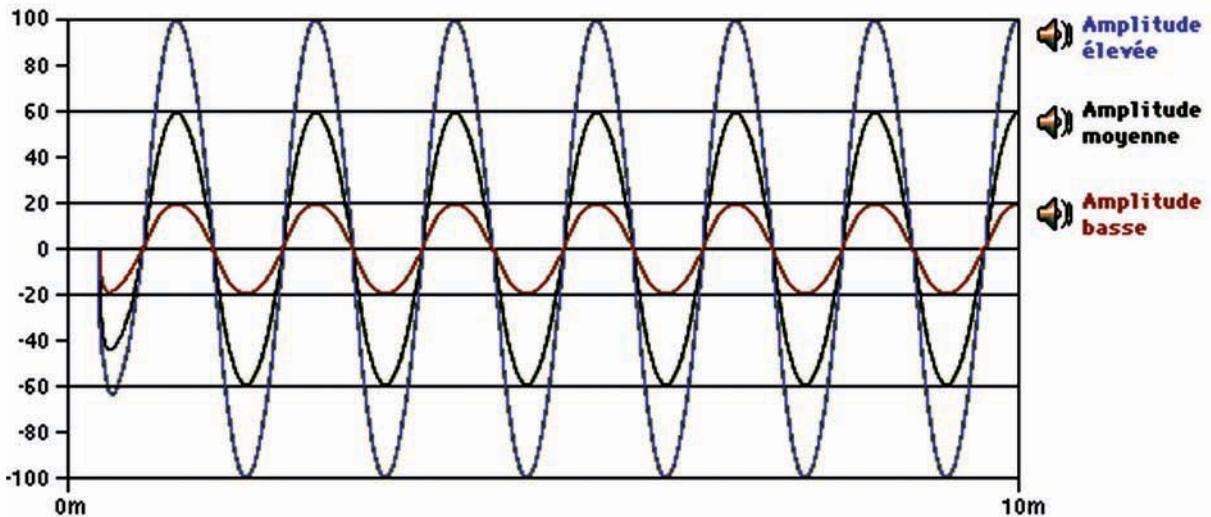


Fig.IV. 15 : Différentes amplitudes pour un même son

IV.1.3. Le timbre

Le timbre est une autre qualité importante d'un son qui est indépendante de la hauteur de son amplitude et qui donne des sensations auditives différentes suivant l'instrument ou la voix qui est à l'origine de ce son. En effet, essayez de prononcer la voyelle "a" par quelques membres de votre groupe. Vous allez remarquer qu'il y a une différence entre les sensations émises par chacun d'eux. Ces différences traduisent les timbres distincts dans chaque prononciation.

Un autre cas où vous pouvez facilement remarquer la différence entre les timbres d'un son est d'essayer de jouer une même note sur des instruments musicaux différents allant du diapason au piano en passant par le violon.



Fig.IV. 16 : Diapason, violon et piano

La voix, la musique et la chanson sont des mixages de plusieurs sons. Vous savez certainement manipuler ces éléments avec un magnétophone ou un lecteur de cassettes ou de disques. L'enregistrement et la lecture sont faits sur des supports magnétiques ou optiques. Le son est donc analogique.

IV.2. Comment numériser un son ?

La numérisation du son est basée sur la notion d'échantillonnage. Elle consiste à réaliser des mesures de ce son à intervalles de temps égaux pendant une unité de temps. Il est clair que plus le nombre est grand, meilleure est la qualité du son rendu. Cet échantillonnage est mesuré en kHz. La plage de mesure qu'on adopte est, elle aussi, importante et se traduit par le nombre maximum de bits significatifs. Cette caractéristique s'appelle la résolution du son. De la même manière, une résolution plus grande permet d'apporter plus de fidélité et de précision au son reproduit.

Pour une résolution de 8 bits, nous disposons de 28 ou de 256 valeurs possibles. (de 0 à 255).

Pour une résolution de 16 bits nous disposons de 65536 valeurs possibles.

Vous verrez que ce paramètre est très important dans le calcul du poids des fichiers sons. Si le son est stéréophonique c'est-à-dire composé de deux pistes, le poids de son fichier double. Un son est dit mono s'il est composé d'une seule piste.

Attention, ces opérations sur ordinateur nécessitent un matériel supplémentaire, dont tous les micro-ordinateurs d'aujourd'hui sont heureusement équipés. Il s'agit de la carte son, des haut-parleurs et du microphone.

IV.3. Les formats d'enregistrement

IV.3.1. Le format WAV

Le format WAV le plus connu est celui associé au système d'exploitation Windows. Les fichiers sons enregistrés sous ce format portent l'extension WAV (Waveform Audio File). Ce format est caractérisé par des fichiers très lourds. En effet, une minute de musique stéréo échantillonnée à 44kHz occupera environ 10Mo d'espace disque. L'idée est de compresser ces fichiers sons sans trop perdre sur la qualité.

IV.3.2. Le format MP3

Actuellement, le format MP3 (MPEG Audio Layer 3) est d'une large utilisation. Ce fût une révolution dans le domaine de la musique.

IV.3.3. Le format MIDI

Le format MIDI (Musical Instrument Digital Interface) est généré par des applications permettant à l'ordinateur de créer la musique voulue sur l'instrument auquel il est branché. Retenons que les fichiers MIDI ont un poids relativement faible.

N.B. Il existe d'autres formats du son dont le format VOC, le format AUD, etc.

Activité 16

Cherchez et repérez les fichiers WAV existant des les disques durs de votre machine. Répétez la même opération pour les fichiers MP3.

Activité 17

Lancez une application de lecture multimédia et écoutez quelques fichiers WAV et quelques fichiers MP3. A chaque fois, lisez les propriétés relatives à l'échantillonnage et à la résolution du morceau joué.

Activité 18

Lancez une application de lecture et d'enregistrement de son. Ouvrez en lecture un des fichiers WAV que vous avez repérés dans l'activité 16 et enregistrez-le sous d'autres noms en modifiant les caractéristiques d'échantillonnage et de résolution. Comparez les qualités et les poids des nouveaux enregistrements.

IV.4. Quelques effets

IV.4.1. L'effet panoramique

On dispose d'un son stéréo. L'effet panoramique consiste à manipuler les deux voix gauche et droite en modifiant leur amplitude respective de telle sorte qu'on perçoive clairement le son provenant alternativement du côté gauche du côté droit.

IV.4.2. L'écho

Rappelons que le son se propage dans les milieux naturels tels que l'air. Le son est réfléchi quand il rencontre un obstacle. La réflexion répétée d'un son génère un écho de ce son.

La majorité des applications de traitement de son disposent de la fonction écho. Souvent, un bon paramétrage permet d'avoir l'effet désiré.

IV.4.3. La réverbération

C'est un autre aspect de la réflexion du son souvent dans un milieu clos. Il se traduit par une persistance du son quand la source de ce dernier a cessé d'émettre.

IV.4.4. Le mixage

C'est l'action de superposer deux sons. On peut agir sur l'amplitude de chacun d'eux pour avoir le mixage désiré.

Activité 19

Utilisez une application de lecture et d'enregistrement de son pour enregistrer votre voix. Vous pouvez lire un texte ou chanter. Enregistrer ce morceau sous le format WAV stéréo avec différents échantillonnages et différentes résolutions.

- Donnez un effet panoramique à l'un des enregistrements.
- Ajoutez un écho à l'un des enregistrements
- Choisissez un morceau de musique et mettez-le comme musique de fond à l'un des enregistrements.

V. La vidéo

V.1. Définitions et matériels

La vidéo est l'ensemble des techniques permettant de traiter des images avec éventuellement du son pour obtenir une projection ou une transmission animée comme au cinéma ou à la télévision.

Une séquence vidéo est composée d'un ensemble d'images successives auquel est accolé un son. La projection ou la lecture d'une séquence vidéo avec une certaine vitesse, produit un effet de continuité. Cette vitesse est de 24 images par seconde. Dès que la vitesse devient inférieure à cette valeur, l'aspect *saccadé* se fait sentir.

La vidéo passe aussi au numérique à travers l'exploitation des outils informatiques utilisés dans ce domaine.

Comme pour la production audio, la vidéo nécessite des périphériques adéquats. Outre le nécessaire audio, il est impératif de disposer d'un moyen de capturer la vidéo. Cette capture peut se faire par l'intermédiaire d'une carte d'acquisition numérique et/ou analogique sur laquelle on peut brancher la source vidéo. Elle pourra aussi se faire via un port rapide sur lequel on branche des sources numériques. Nous citons par exemple les ports USB 2 ou les ports IEEE 1394 appelés aussi Firewire ou iLink. Le débit de transfert de données via ces ports est relativement élevé et avoisine les 400 Mb par seconde. Quant aux sources vidéo, citons par exemple les caméscopes analogiques ou numériques, les tuners des téléviseurs ou des cartes satellites etc.

Les fichiers vidéo sont plus lourds en taille que les fichiers son. Vous avez besoin d'une machine performante avec assez de mémoire vive pour les traitements et suffisamment de mémoires auxiliaires pour les sauvegardes.

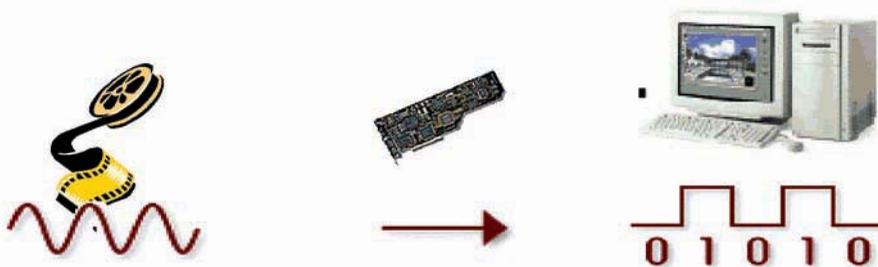


Fig.IV. 17 : La vidéo : de l'analogique au numérique

V.2. Les formats vidéos et les codecs

IV.2.1. Les codecs

Comme les séquences vidéo consomment énormément de mémoire. Il est souvent nécessaire de trouver des moyens de réduire cette mémoire sans compromettre la qualité du film. Ces moyens sont appelés les codecs.

Codec est l'acronyme de codeur/décodeur. Il est basé sur un algorithme de compression et de décompression. Cet algorithme compresse la séquence au moment de son enregistrement pour occuper moins d'espace et décompresse le fichier associé au moment de sa lecture pour retrouver la séquence. Il existe un grand nombre de codecs.

Dans la pratique, tout dépend de ce qu'on veut faire de la séquence et sur quel dispositif on va la diffuser.

Les techniques de compression du flux de données numériques sont principalement basées sur le fait qu'une image contient beaucoup d'informations redondantes.

Cette redondance est de deux types :

a) La redondance spatiale : lorsque des informations sont similaires ou se répètent dans des zones contiguës de l'image

b) La redondance temporelle : lorsque des informations sont similaires ou se répètent dans le temps, même si leurs positions dans l'image ont changé (deux images successives sont souvent relativement proches).

La compression consiste à identifier ces redondances et à les éliminer. Selon les critères de choix de la redondance et la précision avec laquelle elle est déterminée, la compression produira une image plus ou moins proche de l'originale.

IV.2.1. Les formats vidéo

Il existe plusieurs normes de compression, nous citons :

Les extensions les plus fréquentes des fichiers vidéo sont : AVI (plusieurs codecs sont associés à cette extensions, on y retrouve les fichiers vidéo de Windows ainsi que les fichiers compressés en DivX), MPEG, MOV etc.

- **La norme M-JPEG** (Motion Joint Photographic Experts Group) : permet de traiter la vidéo comme une succession d'images fixes. Chaque image étant compressée séparément en utilisant le standard JPEG. Le Motion JPEG permet ensuite le montage à l'image près.
- **La norme MPEG-1** (Moving Pictures Experts Group) : Devant l'impossibilité d'obtenir des facteurs élevés de compression avec le Motion JPEG (max. 4:1 sans perte), les experts se sont basés sur le principe de similitude temporelle, c'est à dire sur la redondance existante entre les images successives.
Le même groupe donnera ensuite les deux normes MPEG 2 et MPEG 4
- **QUICKTIME** : Parallèlement aux travaux de MPEG, le format QUICKTIME est apparu en 1991, et a été intégré au Système 7 des Macintosh. Il s'agit d'un environnement de développement et d'exécution qui permet d'associer à des données classiques des fichiers représentant des séquences sonores ou vidéo.
- **DivX** : Apparu pour la première fois en décembre 2000 sur Internet, le format DivX est un format de compression d'images et de sons qui permet de copier le contenu d'un DVD (soit 90 à 120 minutes de film) sur un ou deux Cd-rom de 700 Mo chacun. Soit une occupation d'espace dix fois moindre que celle d'un DVD.

Ces différents formats demandent le plus souvent le système que vous allez utiliser pour la projection. Nous savons qu'il y a des standards bien connus dans le domaine de diffusion télévisée. Chaque système a ses caractéristiques. Vous trouverez dans le tableau suivant les systèmes les plus connus.

Système	Caractéristiques
NTSC utilisé en Amérique du nord et en Asie	<ul style="list-style-type: none"> - Taille de l'image : 720x480 - Nombre d'images par seconde : 29,97 - Couleurs : millions, qualité : 100% - Son : 32kHz, 16 bits, stéréo
PAL utilisé en Afrique, en Europe et Amérique Latine et en Asie	<ul style="list-style-type: none"> - Taille de l'image : 720x576 - Nombre d'images par seconde : 25 - Couleurs : millions, qualité : 100% - Son : 32kHz, 16 bits, stéréo
MVW Multimedia Video for Windows	<ul style="list-style-type: none"> - Taille de l'image : 320x240 - Nombre d'images par seconde : 15 - Couleurs : millions, qualité : 100% - Son : 22kHz, 16 bits, mono
Multimedia Quick Time	<ul style="list-style-type: none"> - Taille de l'image : 320x240 - Nombre d'images par seconde : 15 - Couleurs : millions, qualité : 100% - Son : 22kHz, 16 bits, mono

Ces formats et ces codecs ont permis une grande évolution dans le domaine de la vidéo. Les lecteurs de toutes sortes deviennent à la portée du grand public. La possibilité de mettre un film complet sur des supports tels le CD ou le DVD a permis une grande diffusion de la vidéo. Vous pouvez également les visionner en famille à l'aide d'un lecteur DVD de salon.

Ci-dessous les interfaces de deux logiciels de montage vidéo.

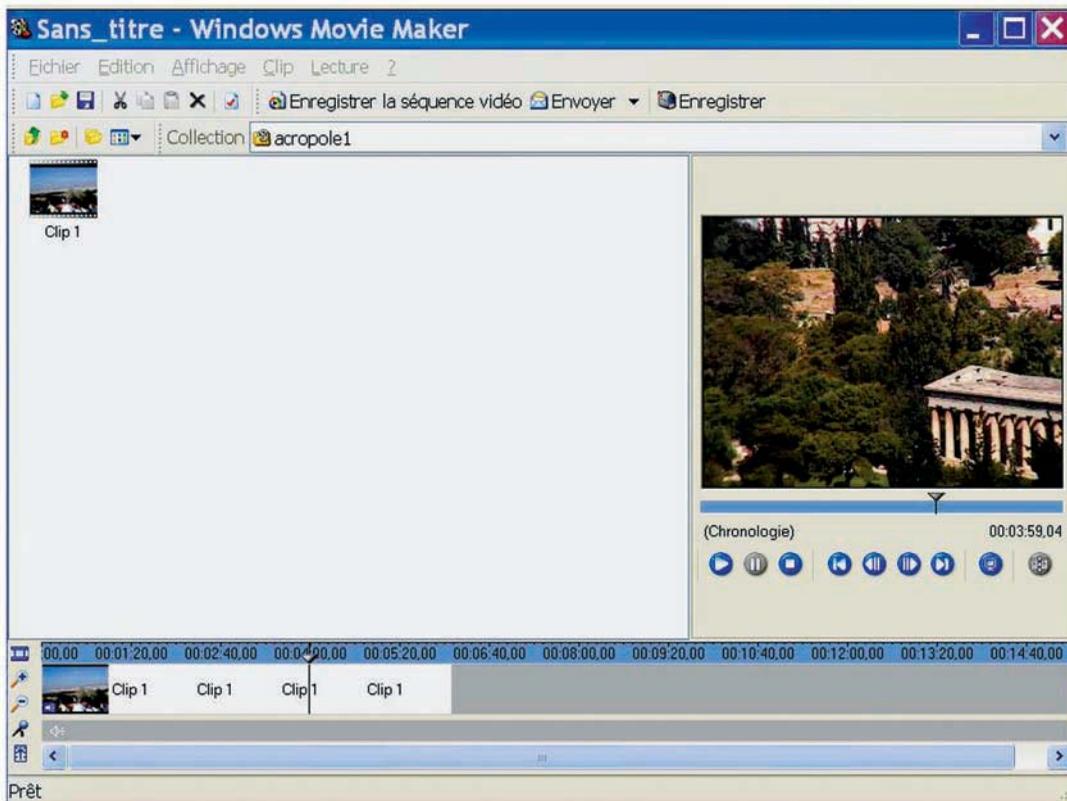


Fig.IV. 18 : Interface de l'application Movie Maker de Microsoft

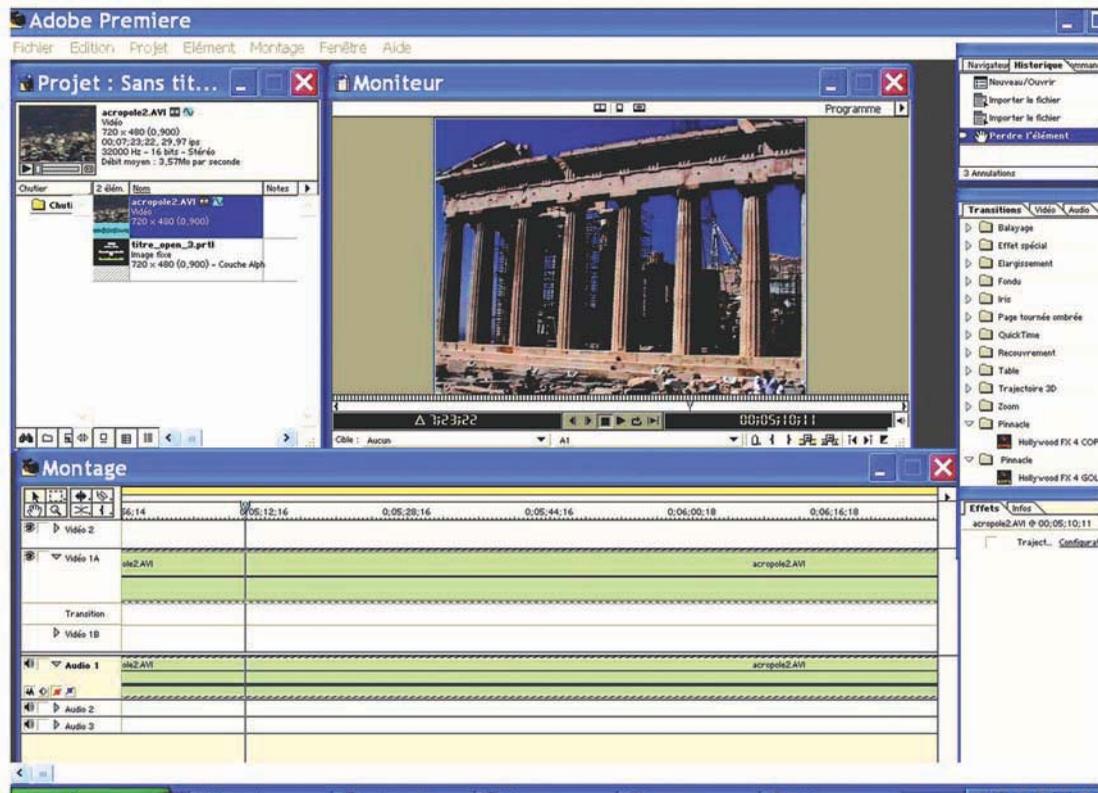


Fig.IV. 19 : Interface de l'application Premiere d'Adobe

VI - Retenons

- 1 Un logiciel de traitement de texte vous aidera à écrire vos textes facilement et à leur donner la présentation que vous désirez. La saisie, la mise en page, la mise en forme, l'insertion d'objets sont les actions de base d'un tel logiciel.
- 2 Pour avoir des images, il est possible d'opter pour la technique de numérisation d'une image en utilisant un scanner ou un appareil photo numérique.
- 3 Le son est la sensation auditive engendrée par une onde acoustique, il est caractérisé par sa fréquence, son volume et son timbre.
- 4 La vidéo est l'ensemble des techniques permettant de traiter des images avec éventuellement du son pour obtenir une projection ou une transmission animée comme au cinéma ou à la télévision.

EXERCICES

Exercice n° 1

- 1/ Dessinez deux ordinateurs
- 2/ Réalisez à l'aide d'un logiciel d'animation un dessin animé montrant le transfert de données entre les deux appareils.

Exercice n° 2

Lancez une application de lecture multimédia et écoutez quelques fichiers WAV et quelques fichiers MP3. A chaque fois, lisez les propriétés relatives à l'échantillonnage et à la résolution du morceau joué.

Exercice n° 3

Lancez une application de lecture et d'enregistrement de son. Ouvrez en lecture un des fichiers WAV que vous avez repérés dans l'activité 16 et enregistrez-le sous d'autres noms en modifiant les caractéristiques d'échantillonnage et de résolution. Comparez les qualités et les poids des nouveaux enregistrements.

Exercice n° 4

Utilisez un graticiel (un logiciel gratuit) pour réaliser une séquence vidéo comportant une démonstration de l'utilisation d'une commande de mise en forme dans un logiciel de traitement de texte.

Exercice n° 5

Lancez une application de réalisation d'une séquence vidéo et repérez les éléments suivants : le moniteur, la table de montage, les pistes vidéo, les pistes audio, les effets audio, les effets vidéos et les transitions.

Importez une séquence existante et la mettez dans la table de montage. Décrivez ces caractéristiques.

CHAPITRE 5

Internet

Objectifs

- Exploiter les services de recherche d'informations, de téléchargement et de communication
- Mettre à profit les services d'Internet pour réaliser un projet

Plan du chapitre

- I. Présentation
- II. Les services d'Internet
- III. Projet
- IV. Retenons

Exercices

Lecture

CHAPITRE 5

Internet

Dans le chapitre précédent, vous avez manipulé différentes formes de média à savoir texte, images, son et vidéo. Ce chapitre vous aide à utiliser Internet à travers ses différents services. A la fin de ce chapitre, vous serez appelé à élaborer un projet dans lequel vous intégrerez les connaissances présentées dans ce chapitre.

I. Présentation

Activité 1

- 1/ Qu'est ce qu'un réseau informatique ?
- 2/ Peut-on réaliser une connexion entre plusieurs réseaux ?
- 3/ D'après vous qu'est ce qu'Internet et quelles sont ses origines ?

I.1. Qu'est ce qu'Internet ?

Le mot **Internet** vient de l'anglais **Inter**connected **Net**works, ce qui lui a valu d'être appelé le "réseau des réseaux". Ces réseaux se communiquent entre eux grâce à des règles d'échanges de données standard. Ces règles sont connues sous le nom de protocoles et permettent de faire communiquer plusieurs ordinateurs et plusieurs réseaux entre eux. Internet est donc un réseau international qui relie des millions d'ordinateurs. De plus amples détails sur le principal **protocole** de communication seront donnés dans le paragraphe I-6.

Internet est de loin le plus grand réseau informatique du monde, un réseau de réseaux tous pouvant échanger des informations en toute liberté. Pour l'utilisateur de l'Internet, les différents ordinateurs branchés au réseau Internet peuvent communiquer entre eux de façon transparente et indépendamment des types d'ordinateurs utilisés (Mac, PC, ou autres). Des logiciels appropriés existent et permettent la connexion à Internet de n'importe quel pays du monde.

I.2. Quelles sont les origines d'Internet ?

Activité 1

Sur votre cahier, copiez puis commentez le tableau suivant et dégagez les dates clés de l'Internet.

Décennie	Evénements
1960	Différents projets de réseaux informatiques ont vu le jour, dont le réseau ARPANET (de la Defense Advanced Research Project Agency) aux Etats Unis. C'est ce dernier qui a donné naissance à l'Internet.
1970	Le réseau s'est agrandi pour accueillir des centres du Département de la défense américain, des sites universitaires et des centres de recherche.
1980	La famille de protocoles de l'Internet fut finalisée et la topologie du réseau qu'on connaît aujourd'hui fut mise en place.
1990	La déréglementation quasi générale subie par les opérateurs de télécommunications a ouvert la voie à de nouvelles structures professionnelles offrant une connexion au réseau Internet à quiconque voulait adhérer au réseau.
2000	Internet est à portée du public et des produits multimédia (intégrant surtout le son et la vidéo) forment les services majeurs offerts par Internet.

À l'origine, Internet reliait entre eux les sites informatiques militaires des États-Unis. Internet s'étend aujourd'hui à la grandeur de la planète et à des sites diversifiés : gouvernementaux, de recherches, d'enseignement, etc. Les services offerts par Internet garantissent son expansion dans le monde et au grand public.

Les services offerts par Internet sont très variés et axés sur l'utilisation des différents médias et surtout le son et la vidéo. Le réseau continue de croître et de se diversifier chaque jour.

I.3. Que trouve-t-on dans Internet ?

Activité 3

- 1/ D'après vous, que peut offrir Internet aux élèves de votre classe ?
- 2/ Décrivez quelques documents tirés d'Internet et montrez le profit qu'un élève peut tirer en utilisant ce réseau.

Dans Internet, on trouve les fichiers informatiques de tout type : textes, images, sons, vidéos et autres. Ce sont les fichiers que les participants au réseau ont bien voulu mettre à la disposition de la communauté Internet. On peut également y trouver des livres, des périodiques, des journaux, des logiciels, des films, tous en format électronique. Cependant, tout ce qui se trouve dans les bibliothèques ou les entreprises n'est pas dans Internet car tout n'est pas disponible en format électronique et, aussi, tout n'est pas libre de droits d'auteur et de droits de diffusion. Malgré les milliards de fichiers qui se trouvent dans Internet, tout n'y est pas et ce qui s'y trouve n'est pas toujours gratuit.

I.4. Que faut-il avoir pour se connecter à Internet ?

Activité 4

Complétez la colonne 2 du tableau suivant par V(rai) si vous jugez que l'élément proposé est indispensable pour pouvoir se connecter à Internet et F(aux) dans le cas contraire.

Élément	Mettre V ou F
Micro-ordinateur	
Imprimante	
Scanner	
Modem	
Ligne téléphonique	
Abonnement chez un FSI	

Vous brûlez d'impatience d'utiliser Internet... pour cela il faut vous assurer de posséder un certain nombre d'outils sans lesquels toute connexion serait impossible.

Pour pouvoir vous connecter à Internet il vous faut :

1- Un ordinateur personnel (PC, un Mac, ...) qui possède au moins 64 Mo de RAM et 5 Mo sur son disque dur.



2- Un logiciel de navigation (appelé navigateur ou browser, le plus répandu étant Explorer).



3- Un modem pour assurer la modulation et la démodulation des informations utilisées. Ayant une vitesse de connexion de l'ordre de 56k, il permet d'exploiter des documents de plus en plus entourés d'images, de son et de vidéo.

4- Un abonnement à un **F**ournisseur de **S**ervices **I**nternet (FSI ou provider) à qui vous payerez un droit d'accès sous forme d'abonnements, généralement mensuels.

I.5. Qu'est ce qu'une adresse électronique ?

Activité 5

- 1/ Donnez des exemples d'adresses Internet que vous connaissez ?
- 2/ Pouvez vous les classer par catégories ou types ?
- 3/ Demandez l'adresse courrier électronique de votre collègue ?

a) Qu'est ce que l'adresse électronique d'un ordinateur ?

Chaque ordinateur connecté à Internet se voit attribuer une adresse électronique appelé aussi adresse IP. C'est le protocole TCP/IP, qui est à la base d'Internet, qui inclut un système d'adresses universel qui permet à chaque noeud d'être localisé sans équivoque. Cette adresse est entièrement numérique et est constituée d'une série de quatre chiffres, entre 0 et 255, séparés par des points.

Si cette adresse IP demeure bien comprise par les machines, elle reste obscure, ou du moins sans signification pour nous. On a donc pensé à faire correspondre à l'adresse IP une adresse écrite en caractères alphanumériques : le **nom Internet**. Cette adresse établit la correspondance entre les deux dénominations.

Application :

Cherchez l'adresse IP de votre machine relativement à votre réseau local.

b) Qu'est ce que l'adresse électronique d'une personne ?

Les adresses de courrier électronique des personnes prennent la forme : utilisateur@adresse_ordinateur. Par exemple, un utilisateur s'appelant *Flen* aurait un compte e-mail sur le serveur *edunet* du ministère de l'Education et de la formation. Son adresse électronique pourrait être : *Flen@edunet.tn*

c) Qu'est ce que l'adresse URL ?

L'adresse URL (*Uniform Resource Locator*) est d'introduction plus récente dans le développement d'Internet. Elle se présente sous la forme générale:

protocole://adresse_du_serveur/répertoire/fichier.

Voici quelques exemples:

<http://www.edunet.tn>

<http://www.planet.tn>

<http://www.tunisia.com>

Les adresses URL sont utilisées par les logiciels de navigation. Ainsi, l'adresse URL spécifie le type de service Internet, en plus de la localisation du serveur en question sur le réseau.

1.6. Qu'est ce que le protocole TCP/IP ?

Dans le paragraphe I-1. Nous avons défini un protocole comme étant une convention de formatage et de transmission de données sur un réseau d'une façon générale. En effet, le protocole essentiel dans une communication Internet se nomme **TCP/IP**, qui signifie Transmission Control Protocol / Internet Protocol. Cela semble très technique, mais ce n'est pas difficile à comprendre.

- **TCP (Transmission Control Protocol)** concerne la gestion des paquets (ou ensembles) de données. C'est un programme qui subdivise les longs messages ou les longs fichiers de données en petits paquets qui peuvent ainsi être envoyés dans le réseau plus facilement et indépendamment les uns des autres.

- **IP (Internet Protocol)** sert à créer l'enveloppe qui sert à la fois d'emballage et d'étiquette à chaque paquet. Chaque enveloppe comporte, entre autres, l'adresse de l'expéditeur et celle du destinataire.

Activité 6 II. Les services d'Internet

Citez quelques services d'Internet et donner les caractéristiques de chaque service cité.

Internet offre plusieurs services à ses utilisateurs. Tous fonctionnent dans l'environnement client-serveur. Pour y avoir accès, l'utilisateur doit disposer des logiciels adéquats.

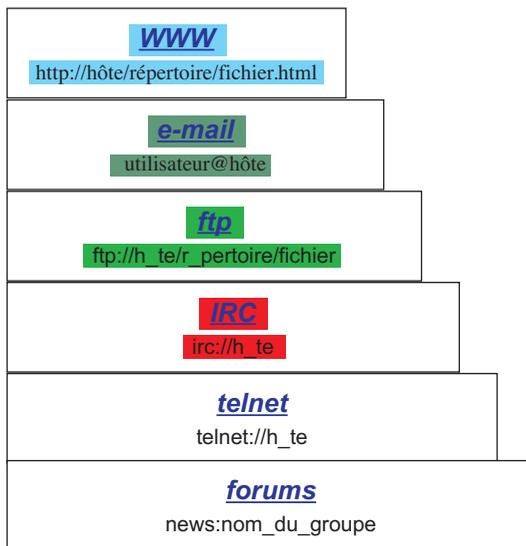
Certains apprécient Internet pour le courrier électronique, d'autres pour ses forums de discussion. Le World Wide Web (le Web ou tout simplement le 3W) est de plus en plus utilisé et ne cesse de se développer. Internet c'est tout cela et encore plus. C'est un ensemble de plusieurs services ou fonctions : une boîte à outils en quelque sorte. Dans Internet, les services sont développés pour gérer des catégories de transfert de données.

- **E-mail** (ou messagerie) est un système de courrier électronique permettant de transmettre des messages d'un ordinateur à un autre.

- **Web** est un outil polyvalent pour automatiser la récupération des documents, d'images, de sons, de vidéos, de données et de toute autre ressource informatique ou objet stocké dans des fichiers sur des serveurs distants.

- **FTP** (file transfert protocol) est un outil pour connecter à un serveur distant et télécharger des fichiers.
- **IRC** est un outil qui permet de discuter interactivement avec des personnes connectées à distance.
- **Telnet** est un outil permettant de se connecter à un serveur distant et de lancer des programmes sur le serveur.
- **News** est un outil permettant la création de groupes ou forums de discussion.

Les services les plus utilisés par l'Internet sont les suivants :



II.1. Le web

Activité 7

Connectez vous à Internet et cherchez des informations concernant :

- 1/ la Tunisie
- 2/ votre ville natale

a) Qu'est ce que le Web ?

Le 3W, le WWW, le World Wide Web ou en un mot le Web est à l'origine du grand succès d'Internet. En anglais, le mot Web désigne une toile d'araignée. Dans le langage de l'Internet, l'expression World Wide Web désigne la toile d'araignée couvrant le monde entier. Développé en Suisse au CERN (Centre Européen de Recherche Nucléaire) en 1990, le Web devient au début de l'année 1995, le moyen le plus populaire d'accéder à cette somme d'informations incommensurable qu'est Internet. Le Web est constitué d'un ensemble de pages contenant des textes et des images reliés par des liens hypertexte. Les pages sont accessibles par un simple clic de souris sur le lien qui y correspond. Pour accéder au Web, il suffit d'avoir un ordinateur connecté à l'Internet et d'utiliser un logiciel de navigation. On peut alors accéder à n'importe quelle page dans le monde entier grâce à son adresse ou URL (Uniform Ressource Locator).

Un bon logiciel de navigation est indispensable pour naviguer sur le Web. Parmi les logiciels les plus répandus et disponibles en version freeware ou shareware, on peut citer :

- NETSCAPE NAVIGATOR : <http://www.netscape.com>
- INTERNET EXPLORER : <http://www.microsoft.com>
- ENHANCED MOSAIC DE NSCA : <http://www.twg.com>

Le Web est souvent utilisé pour la recherche d'informations que vous pouvez effectuer grâce à des moteurs de recherche, par exemple :

- Yahoo <http://www.yahoo.fr>
- Lycos <http://www.lycos.com>
- Google <http://www.google.com>
- Altavista <http://www.altavista.com>

Pour effectuer une recherche, connectez-vous sur l'un des sites mentionnés ci-dessus et définissez un ou plusieurs critères de recherche, appelés mots clés. Le moteur de recherche vous renverra une liste d'adresses correspondant aux critères définis. Il vous suffira, alors de cliquer sur les liens pour accéder aux sites correspondants.

b) Comment rechercher sur le Web ?

La recherche des informations est une action très pratiquée par les utilisateurs d'Internet. Effectuer une bonne recherche ne demande pas seulement un bon équipement, des logiciels appropriés et un accès aux services Internet, mais avant tout d'avoir une idée claire de ce que l'on cherche et de ce que l'on peut s'attendre à trouver.

En fonction de sa requête, l'utilisateur doit choisir le meilleur outil, puis apprendre son mode d'interrogation afin de l'utiliser correctement. La méthode la plus utilisée dans la recherche d'informations fait appel à la technique d'utilisation des mots clefs.

Il est possible d'utiliser les opérateurs logiques pour optimiser la recherche tel que :

- L'opérateur **AND** (intersection) précise la recherche. Par exemple, la requête "Tunisie AND Internet" permettra de récupérer seulement les documents contenant à la fois les mots "Tunisie" et "Internet";
- L'opérateur **OR** (réunion) élargit le champ de recherche. La requête "Tunisie OR Internet" permettra de récupérer tous les documents contenant le mot "Tunisie" ou le mot "Internet";
- L'opérateur **NOT** (exclusion), comme l'opérateur **AND**, sert à préciser la recherche en excluant l'un des termes. Ainsi, la requête "Tunisie **NOT** Internet" permettrait de récupérer tous les documents comprenant le mot "Tunisie" mais non pas le mot "Internet".

c) Application :

Connectez vous à Internet et cherchez des sites évoquant les résultats de la coupe du monde de handball 2005. Utilisez les opérateurs logiques.

Retenons

<http://serveur/repertoire/fichier.html>

Ce type d'URL permet d'accéder à un serveur Web, généralement présent sur une autre machine.

II.2. Le courrier électronique

Activité 8

Avez vous envoyé un courrier électronique à un ami ? Avez vous reçu un courrier électronique d'un ami ? Si oui décrivez les actions que vous avez faites pour saisir ou lire le message.

a) Qu'est ce que le courrier électronique ?

Le courrier électronique permet aux utilisateurs d'Internet de s'échanger des messages écrits de manière efficace et rapide. Pour s'adonner à cette activité, l'utilisateur doit disposer d'une adresse de courrier électronique.

Le courrier électronique, ou e-mail (contraction de Electronic Mail) est sans aucun doute le service le plus utilisé d'Internet. Pourquoi ? Parce que :

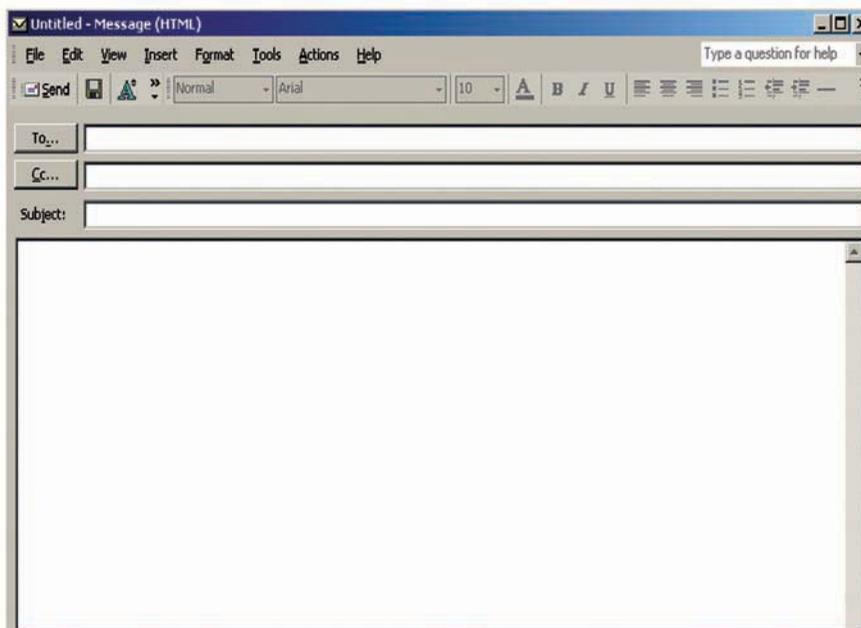
- il est simple (un peu d'inspiration et un bouton suffisent),
- il est rapide (un message peut arriver à l'autre bout de la terre en quelques minutes)
- et il n'est pas cher (au prix d'une communication locale).

Le mot anglais e-mail est resté dans le langage et les utilisateurs parlent de leur adresse e-mail. Le courrier électronique permet aux utilisateurs d'Internet de s'échanger des messages écrits de manière efficace et rapide. En effet, la réception des messages se fait généralement rapidement, et ce à peu près n'importe où dans le monde. Pour s'adonner à cette activité, l'utilisateur doit disposer d'une adresse de courrier électronique. Celle-ci est de la forme utilisateur@hôte où hôte est l'adresse Internet du serveur sur lequel l'utilisateur possède un compte de courrier électronique. Voici un exemple d'adresse de courrier électronique valide : f1en@edunet.tn

Une fois que vous disposez de votre compte Internet, que vous avez installé votre logiciel de messagerie et mémorisé votre adresse électronique, vous pouvez envoyer vos premiers messages. Les logiciels de messagerie sont en général très simples à utiliser. Ils comportent généralement un carnet d'adresses vous permettant de noter les adresses de vos correspondants et éviter ainsi de les ressaisir à chaque fois.

b) Comment utiliser le courrier électronique ?

Décrivez le contenu de l'écran suivant.



En-tête
du
message

C
O
R
P
S
D
U
M
E
S
S
A
G
E

Un message est divisé en deux parties distinctes. La première partie est l'en-tête où l'on trouve les renseignements concernant la logistique du message. La deuxième partie est le corps du message: c'est là que se situe le texte rédigé par l'auteur du message.

L'entête est de la forme :

Le diagramme illustre la structure d'un en-tête de message électronique. Il est composé de trois champs distincts, chacun avec une étiquette à gauche et un champ de saisie à droite. Le premier champ est étiqueté 'To...' (à destination de), le deuxième 'Cc...' (en copie) et le troisième 'Subject:' (sujet).

- Dans le champ *To...*, on saisit le destinataire du courrier.
- Dans le champs *Cc...* (Carbon Copy) qui est réservé à une liste d'utilisateurs qui recevront le courrier en copie. Dans cette liste de destinataires, chaque adresse est séparée par une virgule (ou un point virgule).
 - Dans le champs *Subject* : on saisit le sujet du message qui apparaît dans la liste courrier lorsque vous les récupérer. Le sujet doit être concis et le plus explicite possible.
 - On peut trouver aussi un champs *Bcc...* (Blind Carbon Copy) qui permet de donner une liste de destinataires, mais contrairement au champ *Cc...* chacun des utilisateurs n'aura pas connaissance de la liste des autres lecteurs de ce même courrier.
 - On trouve également un champ *Attachment* dans lequel l'expéditeur peut donner les noms des fichiers qui seront expédiés en pièces jointes en même temps que le courrier.

c) Application :

Rédigez puis envoyez un message à votre collègue pour lui souhaiter bonne année scolaire.

Retenons

nom@organisation.domaine

Ce type d'URL permet d'écrire un courrier électronique à l'utilisateur dont l'adresse figure dans l'URL.

II.3. Le transfert de fichiers

Activité 9

Vous voulez vous connecter à Internet pour chercher des informations sur un virus qui a contaminé les fichiers de votre ordinateur. Le résultat de la recherche est fort intéressant puisqu'on fournit l'adresse d'un site à partir duquel il sera possible de télécharger un antivirus. Comment allez-vous procéder pour réaliser l'action de téléchargement ?

a) Qu'est ce que le transfert de fichiers ?

FTP signifie File Transfer Protocol, en français : "protocole de transfert de fichiers". Il permet à un utilisateur de charger un fichier depuis un serveur ou d'y déposer un fichier (pour autant que le serveur l'y autorise).

FTP est, donc, un protocole de transferts de fichiers qui reste le mode de transfert privilégié, notamment pour les fichiers volumineux. Grâce à FTP, il est possible de télécharger les dernières mises à jour des logiciels ou gérer son site Web à distance.

Pour charger un fichier par FTP, il est nécessaire :

- de se procurer un logiciel FTP (WS-FTP, Cute-FTP, FTP-Expert...).
- de connaître le nom du serveur où se connecter.
- d'avoir les codes d'accès nécessaires : le nom du compte ("anonymous" est le compte public par défaut de la plupart des serveurs) et un mot de passe ("guest" dans le cas des connexions anonymes).

Utiliser FTP, c'est un peu comme lister des fichiers sur le répertoire d'un disque dur. La différence, c'est que la recherche ne s'effectue pas dans l'arborescence de votre disque, mais dans celle du disque d'un autre ordinateur sur Internet. Certains sites imposent une autorisation préalable pour ouvrir l'accès à leurs machines. D'autres jouent la carte de la liberté totale. Dans ce cas lorsque la machine vous demande votre nom d'utilisateur, tapez anonymous (anonyme).

FTP est un outil réservé aux initiés et reste assez peu convivial pour le grand public qui ne connaît généralement pas les références exactes de ce qu'il cherche, alors que les initiés connaissent le nom des fichiers à transférer et les serveurs FTP sur lesquels ils se trouvent.

FTP peut présenter des risques lorsqu'il est utilisé sans précaution sur des serveurs douteux. En effet, il n'y a pas de meilleur moyen pour véhiculer les virus informatiques.

b) Application

Télécharger l'antivirus résultat de la requête de l'activité.

Retenons

<ftp://serveur/repertoire/fichier>

Ce type d'URL permet d'accéder à un serveur ftp et :

- de visualiser l'ensemble des fichiers d'un répertoire si aucun fichier n'est spécifié,
- de rapatrier le fichier sur votre disque local si un nom de fichier est donné.

II.4. IRC ou "clavardage"

IRC (Internet Relay Chat), vous permet de discuter "en direct" avec des personnes connectées à distance en utilisant le clavier. Ce que vous saisissez apparaît sur les écrans de ceux qui utilisent le même canal que vous. Pour se distraire, il n'y a pas mieux.

Cette forme de conversation s'effectue textuellement, plutôt qu'oralement, par l'intermédiaire du clavier (pour cette raison, certains proposent le terme "clavardage"). En général, ces conversations se déroulent de façon anonyme, car les utilisateurs s'identifient par des pseudonymes. De plus, les conversations sont ouvertes, c'est-à-dire que tous ceux qui ont rejoint un canal de conversation peuvent voir les interventions des différents participants sur leur écran.

Une dizaine, si ce n'est une centaine, de personnes peuvent utiliser le même canal, et rien ne vous interdit d'être dans deux endroits à la fois pour discuter avec deux personnes différentes, ubiquité oblige !

Lorsqu'un utilisateur se joint à un site de cyberbavardage, il doit s'identifier en utilisant son pseudonyme, préalablement défini sur ce site. Une fois connecté, il choisira l'un des canaux offerts, selon le centre d'intérêt qui l'intéresse. Chaque canal actif a un opérateur en ligne qui veille à ce que les principes éthiques définis pour ce canal soient respectés.

IRC est fondé sur la technique " client-serveur ". Les serveurs sont des machines disséminées sur la planète et qui transmettent les discussions à travers le monde. Le client, c'est votre logiciel qui interprète tous les codes curieux qui circulent et vous les présente sous une forme... humainement compréhensible.

Le chat est un type d'activité plutôt qu'une technologie ou un service Internet précis. Il peut s'effectuer sur des services généraux, tels que le Web ou telnet, ou des services spécialisés, tels que IRC et ICQ.

II.5. Telnet

L'une des fonctions importantes à l'origine d'Internet était de permettre aux chercheurs des centres de recherche et universités d'accéder à distance aux ordinateurs mis à leur disposition. Le protocole telnet sert à cette fonction. L'adresse URL pour telnet est de la forme suivante : `telnet://h_te`. Certains catalogues de bibliothèques universitaires peuvent être consultés par le biais de telnet. Pour que vous puissiez effectuer la communication, il vous faut un logiciel-client telnet installé sur votre ordinateur et adéquatement configuré pour votre logiciel-navigateur.

C'est un système de connexion primaire, basé sur des commandes, entre deux ordinateurs distants. Avec Telnet, tous les ordinateurs deviennent un peu le vôtre. Telnet vous permet de prendre le contrôle d'un ordinateur situé à l'autre bout du monde. En fait, votre micro devient un terminal de la machine distante. L'accès par Telnet n'est autorisé que sur les ordinateurs pour lesquels vous disposez d'une autorisation ou pour ceux " ouverts " au public.

Certains catalogues sont encore en version telnet, mais progressivement ils vont migrer vers des versions Web.

Retenons

Telnet://Nom:Password@serveur:port

Ce type d'URL permet d'ouvrir une session telnet, c'est à dire une fenêtre représentant la console d'une machine distante présente sur Internet.

II.6. Forums

Que ce soit pour résoudre leurs problèmes ou faire avancer leur science ou tout simplement demander de l'aide, les utilisateurs d'Internet ont besoin d'entrer en contact avec d'autres spécialistes comme eux. Le courrier électronique s'est vite répandu comme un moyen efficace d'établir ce contact. Cependant le courrier électronique présente au moins une limitation importante: il faut savoir a priori vers qui envoyer nos messages. C'est ainsi qu'ont vu le jour les forums électroniques. Le phénomène est vite sorti du domaine exclusif des universitaires et tout sujet est devenu propice pour la création de groupes ou forums de discussion.

Dans Internet, on trouve principalement deux types de forums :

- Le premier type de forum est en fait basé sur le courrier électronique. Ces forums fonctionnent par la création et le maintien d'une liste d'abonnés. L'intéressé peut s'y inscrire en envoyant un message bien défini à l'adresse du forum (ou **listserv**). Une fois que son adresse est incluse dans la liste du groupe, tous les messages postés au forum lui seront acheminés dans sa boîte postale électronique.
- Le second type de forum se nomme Usenet. Essentiellement, tous les messages envoyés à un groupe de discussion de Usenet sont envoyés à tous les serveurs **Usenet** du monde. Au lieu de s'accumuler dans votre boîte de courrier personnelle, les messages des groupes Usenet sont consultables sur les serveurs Usenet.

Certains groupes de discussion sont disponibles selon les deux formules : on peut s'y abonner personnellement ou aller les consulter sur un serveur Usenet, au choix.

Retenons

[news:nom_du_groupe](#)

Permet d'obtenir un groupe de discussion de Usenet.

III. Projet

Activité 10

Utilisez Internet pour réaliser un dossier présentant les définitions des composants d'un micro-ordinateur que vous avez étudié dans le chapitre II.

Le dossier, qui sera présenté sous forme d'un ensemble de pages, comportera :

- une première page où on trouvera votre nom et votre classe ainsi que le thème du projet
- une page pour chaque composant dans laquelle on trouvera son nom, son image et une définition
- une dernière page qui comportera les adresses électroniques des différents sites Internet que vous avez utilisés pour réaliser ce projet.

III.1. Quelles sont les étapes à suivre pour réaliser un projet ?

Pour réaliser un projet, vous suivrez les étapes suivantes :

- 1-bien lire le thème du projet
- 2-préparer un dossier électronique (sous un logiciel de traitement de texte par exemple) pour mettre les documents choisis d'Internet
- 3-faire une recherche dans Internet et copier dans le dossier créé tous les documents (textes, images, ...) utiles au projet
- 4-faire un tri des documents copiés et ne laisser que les documents que vous considérez bon pour le projet
- 5-classer les documents par critère que vous choisissez (par composant dans le cas de ce projet)
- 6-ajouter vos commentaires
- 7-lire le projet réalisé et assurez-vous que son contenu st bien structuré
- 8-présenter le projet à votre professeur.

IV - Retenons

- Le mot "Internet" vient de l'anglais "*Inter*connected *Net*works" et signifie un réseau international d'ordinateurs communiquant entre eux grâce à des règles d'échanges de données standard.
- Dans Internet, on trouve les fichiers informatiques, textes, images, sons, vidéos et autres que chaque participant au réseau a bien voulu mettre à la disposition de la communauté Internet.
- Pour pouvoir vous connecter à Internet il vous faut :
 - 1- Un ordinateur personnel
 - 2- Un logiciel pour le Web
 - 3- Un modem
 - 4- Un abonnement à un fournisseur d'accès
- Internet offre plusieurs services à ses utilisateurs : le web, le courrier électronique, le téléchargement de fichiers, ...

EXERCICES

Exercice n° 1

Mettez V (Vrai) dans la case qui correspond à chaque proposition si vous jugez qu'elle est vraie sinon mettez la lettre F (Faux).

1/ Internet est un(e)

énorme ordinateur réseau de réseaux autoroute

2/ Comment sont reliés les réseaux Internet ?

Avec des pinces croco Avec un protocole TCP/IP Avec des TAM-TAM

3/ Internet a été créé en 69 79 89

4/ Comment appelle-t-on le courrier électronique d'Internet ?

E-Smail E-Mail E-Pail

5/ Pour se connecter à Internet, on a besoin

d'un modem d'un scanner d'un abonnement

6/ TCP/IP veut dire Protocol de contrôle de

réseau paiement transmissions de données

7/ Que veut dire WWW ?

Word Wild Wait World Wide Web Word Wide West

8/ A quoi servent les Newsgroups ?

A s'abonner à une revue A lire les articles de journaux A consulter des groupes

9/ Qu'est ce que FTP ?

Un outil de transfert de fichiers Un service de presse Une agence d'Internet

10/ En quelle langue communiquez-vous sur Internet ?

Uniquement en français Uniquement en anglais Dans toutes les langues

11/ Que pouvez-vous faire avec l'IRC ?

Faire de la recherche Passer un scanner Communiquer avec d'autres personnes

12/ A quelle heure pouvez-vous vous connecter à Internet ?

le matin le soir 24 heures sur 24

13/ Pouvez-vous télécharger des images ? Oui Non Peut-être

14/ Comment appelle-t-on la première page de Navigator ?

La home page Le fichier d'accueil L'annuaire

15/ Que veut dire " NewsGroup " ?

Groupe de discussion Atelier d'édition Kiosque à journaux

16/ Comment préciser un pays dans une recherche ?

Par un numéro de code Par deux lettres ex : tn Par le nom complet du pays

Exercice n° 2

Pour définir les éléments indispensables pour la connexion à Internet, mettre V(Vrai) dans la case qui correspond à chaque élément si vous jugez qu'il est indispensable sinon mettre la lettre F(Faux). Justifier votre choix.

Elément	V / F	Justification
Micro-ordinateur		
Imprimante		
Logiciel de traitement de texte		
Scanner		
Logiciel de traitement d'images		
Modem		
Ligne téléphonique		
Abonnement chez un FSI		
Navigateur		
Logiciel de traitement de son		

Exercice n° 3

- 1/ Qu'est ce qu'une adresse électronique d'une personne ? donnez un exemple.
- 2/ Qu'est ce qu'une adresse URL ? donnez un exemple
- 3/ Pour chacune des adresses données dans le tableau suivant, mettez **AEP** s'il s'agit d'une adresse électronique d'une personne ou **URL** lorsqu'il s'agit d'une adresse URL. Dans chaque cas, donnez le nom du protocole utilisé.

Adresse	Type	Protocole utilisé
Ali.hamdi@edunet.tn		
http://www.edunet.tn		
http://www.yahoo.fr		
http://www.google.com		
Med_04@yahoo.fr		
http://www.planet.tn		
http://www.tunisia.com		

Exercice n° 4

Connectez vous à Internet et cherchez des définitions aux termes suivants :

- navigateur
- moteur de recherche
- browser
- annuaire
- répertoire thématique

Comparez les définitions données par différentes sources.

Exercice n° 5

- 1/ Quel est l'acronyme du protocole TCP/IP ?
- 2/ Quels sont les rôles des protocoles TCP et IP ?

Exercice n° 6

Connectez vous à Internet et réalisez les actions suivantes :

- recherchez des images de personnalités, de compositeurs, écrivains, ...
- recherchez une donnée de type son
- effectuez une recherche dans les Newsgroups
- recherchez un logiciel à télécharger.

Lecture

Quelques statistiques sur l'utilisation du réseau Internet

N.B. Ces chiffres sont à prendre avec précaution : d'une part, ils sont établis par des instituts différents, avec des méthodes sans doute différentes, de l'autre, ils sont susceptibles d'avoir beaucoup changé selon la date à laquelle ils ont été établis. Par exemple, le nombre d'internautes en Chine tourne, fin 2003, entre 70 et 100 millions selon les statistiques...

- En **France**, fin 2003, 40 % des foyers disposaient d'un ordinateur, 29 % d'une connexion internet et 13,6 % d'une connexion haut débit, selon les derniers chiffres publiés par ZDNet en janvier 2004. En septembre 2003, 19 681 000 personnes de 11 ans ou plus se sont connectées au moins une fois à Internet (soient 39 %, contre 42 % en juin 2003 et 32,5 % en juin 2002) – en novembre 2003, le nombre d'internautes était estimé à 21,7 millions. [Source : Les Baromètres Multimédia – Médiamétrie – Septembre 2003]

Selon une enquête du CREDOC [Centre de recherche pour l'étude et l'observation des conditions de vie] de juin 2003, 46 % des plus de 18 ans disposent d'un ordinateur à domicile (et 69 % des 12-17 ans), 31 % disposent à domicile d'une connexion Internet (40 % des 12-17 ans) et 13 % d'une connexion haut débit (19 % des 12-17 ans). Fin 2004, on estimait à 6 millions le nombre de foyers disposant d'une ligne haut débit, soit près d'un foyer sur deux.

- **Statistiques pour quelques pays** (pris – presque – au hasard) : population (en millions – 2004), nombre d'internautes (en valeur absolue, en millions, et en pourcentage de la population – 2002) et produit intérieur brut par habitant (en dollars US en parité de pouvoir d'achat – 2003) [Source : The World Factbook publié en ligne par la CIA...]. On notera les fortes disparités d'un pays à l'autre (p.ex., 68,5 % des Suédois utilisent Internet contre 0,5 % des Vietnamiens), mais surtout le pourcentage d'internautes n'est pas proportionnel au PIB par tête : à PIB par tête semblable, l'Allemagne compte 25 % d'internautes (par rapport à la population) de plus que la France, la Chine presque trois fois plus que l'Ukraine. Et bien que le PIB/tête de la Grèce soit très nettement supérieur à celui de la Pologne, le pourcentage d'internautes y est inférieur. Disparités liées à des cultures différentes, à une histoire récente différente et à des politiques économiques différentes: il faudrait tenir compte aussi des taux de croissance du PIB et du nombre d'internautes.

Pays	Population (en millions)	Internautes (en millions)	%	PIB/tête (PPA)	Pays	Population (en millions)	Internautes (en millions)	%	PIB/tête (PPA)
Algérie	32,1	0,5	1,6	5 900	Roumanie	22,4	1,8	8,0	6 900
Allemagne	82,4	34,0	41,3	27 600	Royaume Uni	60,3	25,0	41,5	27 700
Chine	1 300,3	59,1	4,5	5 000	Russie	143,8	6,0	4,1	8 900
France	60,4	18,7	31,0	27 500	Slovaquie	5,4	0,9	16,5	13 300
Grèce	10,7	1,6	15,0	19 900	Suède	9,0	5,1	56,7	26 800
Irlande	4,0	1,0	25,0	29 800	Tchèque	10,2	1,6	15,7	15 700
Japon	127,3	53,2	41,8	28 000	Tunisie	10,0	0,5	5,0	6 900
Maroc	30,6	0,22	1,3	3 767	Turquie	68,9	4,9	7,1	6 700
Mauritanie	3,0	0,01	0,3	1 800	Ukraine	47,3	0,9	1,9	5 300
Niger	11,4	0,015	0,1	800	USA	293,0	159,0	54,3	37 800
Pologne	38,6	8,9	8,9	11 000	Vietnam	82,7	1,5	0,5	2 500

Internet et diversité linguistique

Selon les données publiées en janvier 2003 par l'Union internationale des télécommunications, les langues utilisées sur Internet se répartissent ainsi :

Langue	%
Anglais	68,3
Japonais	5,9
Allemand	5,8
Chinois	3,9
Français	3,0
Espagnol	2,4
Russe	1,9
Italien	1,6
Portugais	1,4
Coréen	1,3
Autres	4,6

Selon d'autres chiffres, pour l'année 1997, l'anglais représentait 84 %.

Une belle série de statistiques sur Internet

Une série impressionnante de statistiques en tous genres, des plus généralistes aux plus précises, concernant l'utilisation d'Internet à pas mal de sauces. 934 millions de personnes connectées en 2004, 1.07 milliards prévus en 2005, 1.21 en 2006, 1.35 en 2007 : Diantre.

Aux Etats-Unis, on compte pas moins de 188.5 millions, les autres pays leaders du nombre de connectés étant:

Royaume-Uni	(33.11 millions),
Corée du Sud	(31.67 millions),
Russie	(21.23 millions),
Japon	(78.05 millions),
Inde	(36.97 millions),
Italie	(25.53 millions),
Allemagne	(41.88 millions),
France	(25.47 millions),
Chine	(99.8 millions),
Brésil	(22.32 millions).

Voici en outre un extrait (traduit) de la liste :

- 10% des très jeune ont leur propre site
- 13 millions d'internautes espagnols en cette fin 2004
- 14% des américains veulent surveiller leur maison avec des webcam
- 15.6 millions de personnes regarderont la télé sur ip en 2007
- 17% ont utilisé Internet par le Wifi
- 23% des américains connectés ont participé à une vente aux enchères
- 65% des internautes ont acheté au moins un produit en ligne
- 22 millions d'américains se servent d'un service de banque en ligne
- 22 millions de jeunes américains attendus pour 2008
- 22% des américains âgés de plus de 65 ans sont connectés
- 23% des Bulgares ne connaissent pas Internet
- 24.7% des américains de la partie rurale ont accès à Internet
- 25% des utilisateurs américains pensent changer de FAI
- 28 millions d'utilisateurs attendus en Europe Centrale et de l'Est pour 2006
- 3.8 millions d'Ukrainiens en ligne
- 30 millions de Coréens (Sud) en ligne

CHAPITRE 6

Éléments de présentation

Objectifs

- Produire une présentation
- Produire des pages web.

Plan du chapitre

I. La production de présentation

- I-1. Qu'est ce qu'une présentation ?
- I-2. Comment créer une présentation ?
- I-3. Quels sont les éléments de base d'une présentation ?
- I-4. Comment réaliser une présentation ?

II. La production de pages Web

- II-1. Introduction
- II-2. Quelles sont les étapes de création d'un site Web ?

III. Retenons

Exercices

Lecture

CHAPITRE 6

Éléments de présentation

Une présentation est un moyen de transmettre et de communiquer des informations à autrui. L'informatique a aidé à mieux réaliser les présentations et ceci en fournissant des outils logiciels puissants et une technologie adaptée qui permettent à l'utilisateur de réaliser des présentations comportant diverses options. Dans ce chapitre, vous allez apprendre à utiliser deux logiciels qui vont vous permettre de réaliser des présentations sous forme de diaporamas ou sous forme de pages web.

I. La production de présentation

I.1. Qu'est ce qu'une présentation ?

Conçus à l'origine pour aider un conférencier à illustrer un exposé, les logiciels de présentation permettent la préparation de documents d'accompagnement. Les présentations sont pilotées à partir d'un ordinateur et projetées sur un grand écran ou sur un téléviseur. Le conférencier contrôle leur déroulement à l'aide d'une souris ou d'une télécommande. Il est ainsi libéré de la recherche et de la manipulation de ses documents.

Avec l'intégration du multimédia aux outils informatiques, il est maintenant aisé d'associer au texte de nombreuses illustrations (graphiques, schémas, dessins, images, photos, animations...) et de l'accompagner de messages sonores (commentaires, musique, bruitage...). Aujourd'hui, une présentation n'est autre qu'un instrument d'assemblage de séquences imagées et sonorisées, utilisée pour la communication dans l'entreprise, dans les expositions et les forums, dans l'enseignement et la formation, etc.

L'introduction d'interactions, lorsqu'elle est prévue, oriente la présentation vers une utilisation plus individuelle, comme la consultation de séquences rapides de formation ou de soutien, de diaporamas documentés et commentés, de bornes interactives, etc.

Il existe divers logiciels qui permettent de réaliser des présentations. Le tableau suivant donne quelques exemples de logiciels de PréAO (**Pr**ésentation **A**ssistée par **O**rdinateur).

Éditeur	OpenOffice	Microsoft	Lotus	Asymétrix	Micrografx	Macromédia
Logiciel	Impress	Powerpoint	Freelance Graphics	Asymétrix Présentation	Charisma 4.0	Action !

L'utilisation de la PréAO nécessite de prendre en compte deux types de préoccupations. Il faut :

a) D'une part, apporter à l'auteur une aide efficace dans la préparation de son exposé : simplicité, rapidité, efficacité... de création, qualité et unité visuelle, préparation et impression de documents d'accompagnement.

b) D'autre part, il est indispensable d'assurer le confort du conférencier et celui du public lors du déroulement de la prestation.

En effet, une utilisation efficace de la PréAO repose autant sur la conception d'une présentation soignée que sur la mise en œuvre de dispositifs technologiques adaptés.

I.2. Comment créer une présentation ?

Il est possible de créer une présentation de plusieurs façons. On peut utiliser l'Assistant accompagnant les logiciels de présentation qui suggère un contenu et un modèle de présentation. Généralement, cet assistant contient plusieurs types de présentation, par exemple pour une réunion ou la planification d'un événement, ainsi que pour des présentations que vous pouvez utiliser sur Internet.

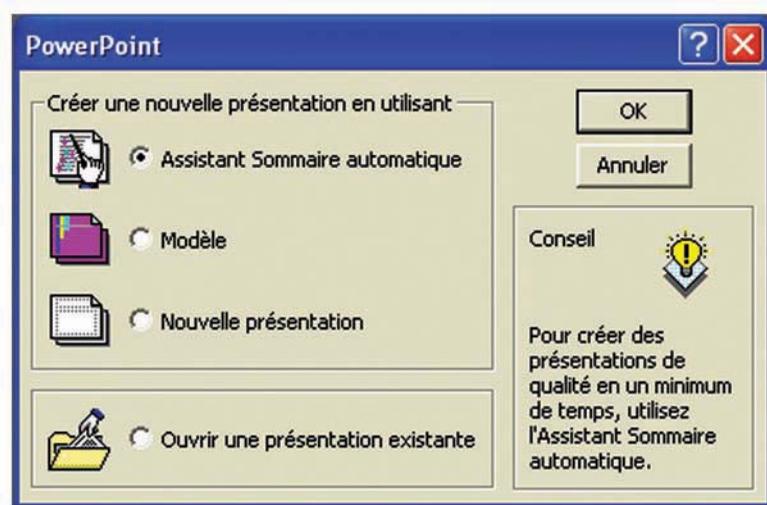


Fig.VI. 1 : Création d'une présentation

On peut également démarrer une présentation en sélectionnant un modèle de conception dont dépend le modèle de la présentation, mais qui ne comprend pas le contenu. On peut aussi commencer avec un plan qu'on importe d'une autre application, tel qu'un logiciel de traitement de textes, ou avec une présentation vide qui ne comprend ni texte ni modèle.

I.3. Quels sont les éléments de base d'une présentation ?

a) Diapositives

Lorsqu'on démarre un logiciel de présentation, on trouve immédiatement des outils qui servent à créer les écrans successifs de la présentation, souvent appelés " **diapositives** ".

Ces écrans sont composés d'un ensemble d'objets de nature diverses (textes, graphiques, diagrammes, éléments multimédias, etc.). Leur création est assez intuitive, et assistée par des boîtes de dialogue bien documentées et faciles à remplir.

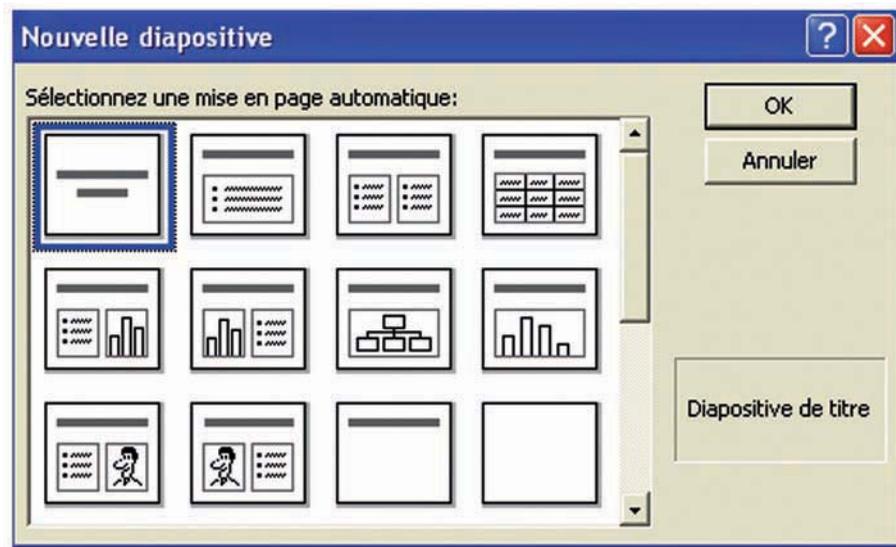


Fig.VI. 2 : Création d'une nouvelle diapositive

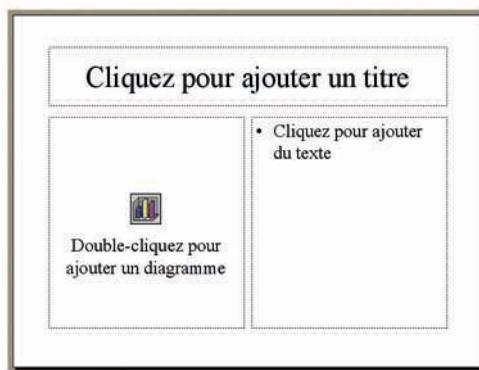


Fig.VI. 3 : Diapositive comportant un titre, un diagramme et du texte

b) Comment animer du texte ou des objets ?

L'animation est une fonction permettant d'ajouter un effet visuel ou sonore spécial à du texte ou à un objet. Par exemple, vous pouvez faire en sorte que les puces de votre texte arrivent par la gauche, un mot à la fois, ou faire entendre du son lorsque vous dévoilez une image. Vous pouvez animer un nombre quelconque d'objets sur une diapositive et même animer les éléments d'un graphique.

Vous pouvez animer du texte, des diagrammes, des sons, des films et d'autres objets dans vos diapositives, de manière à mettre en évidence les points importants, contrôler le flux d'informations et rendre votre présentation plus attrayante. Vous avez la possibilité de faire apparaître chaque puce principale indépendamment des autres ou d'afficher les objets progressivement, les uns après les autres.

Vous pouvez définir les modalités d'affichage de chaque puce ou objet sur votre diapositive (par exemple, avec un balayage gauche) et décider d'estomper ou de changer la couleur des autres puces ou objets lorsque vous ajoutez un nouvel élément. Vous pouvez aussi animer les éléments d'un diagramme.

Vous pouvez également modifier l'ordre et le minutage de vos animations et les définir de sorte qu'elles s'affichent automatiquement sans cliquer sur la souris.

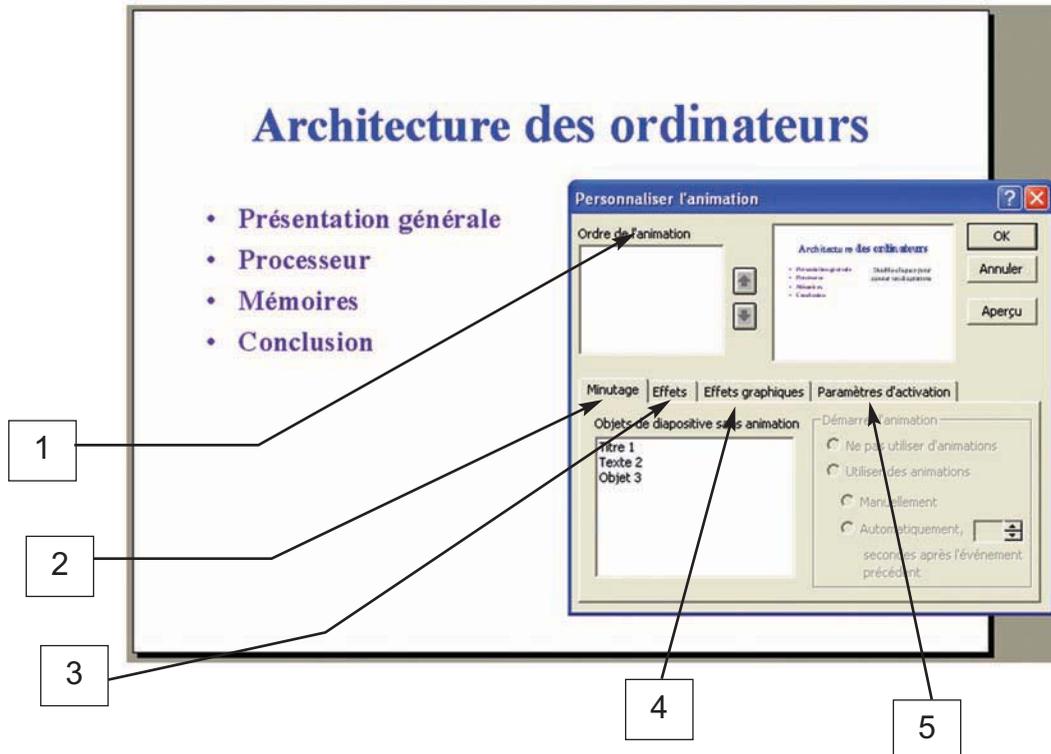


Fig.VI. 4 : Personnalisation d'une présentation avec Power point

N°	Type d'animation	Commentaires
1	Ordre de l'animation	Définit l'ordre d'apparition des éléments de la diapositive
2	Minutage	Type d'utilisation de l'animation (manuelle ou automatique après combien de secondes)
3	Effet	Effets sur le texte objet de l'animation (type d'entrée, son d'accompagnement, ...)
4	Effets graphiques	Effet sur le graphique utilisé (type d'introduction, animation de la légende et du quadrillage, ...)
5	Paramètres d'activation	Pendant l'activation, interruption de l'activation, ...

c) Comment ajouter un graphique dans une présentation ?

Une présentation peut comporter des graphiques que l'utilisateur peut construire ou insérer à partir d'autres logiciels. D'une façon générale, les logiciels de présentation prévoient un espace réservé à la création ou à l'insertion de graphique.

Lorsque vous voulez créer un graphique, double-cliquer sur l'espace réservé au diagramme et le logiciel affiche un diagramme et les données qui lui sont associées. Il est également possible d'insérer un graphique à partir d'un logiciel tableur dans votre présentation.

d) Comment ajouter des commentaires à une présentation ?

Lorsque vous visualisez une présentation, les logiciels de présentation vous permettent d'insérer vos commentaires directement sur les diapositives. Les commentaires apparaissent dans des zones de commentaires ; vous pouvez déplacer, redimensionner et remettre en forme le texte et les zones de commentaires, exactement comme vous le feriez avec n'importe quel objet. Vous pouvez également distribuer votre présentation et demander à d'autres personnes d'y ajouter leurs commentaires.

Il est aussi possible de masquer les commentaires si vous voulez projeter un diaporama sans que l'on les voit, ou si vos diapositives deviennent trop encombrées.

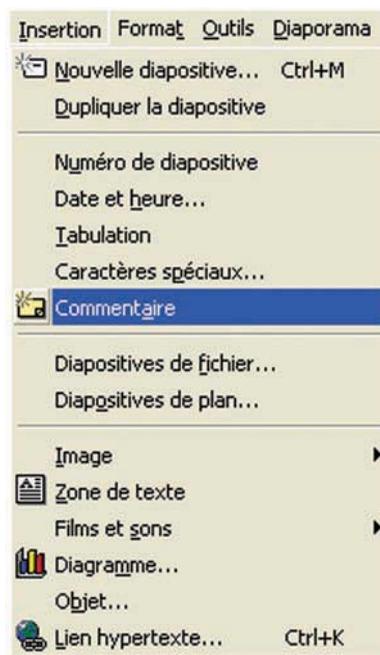


Fig.VI.5 : Ajout d'un commentaire

e) Comment ajouter une transition à une présentation ?

Une transition est un effet spécial utilisé pour présenter une diapositive. Vous pouvez utiliser plus d'une transition dans une même présentation. Il est possible, par exemple, de choisir le type d'ouverture de la diapositive, le type de fermeture, le type de balayage, etc. Il est aussi possible de choisir les transitions souhaitées et modifier la vitesse de chacune d'elles.

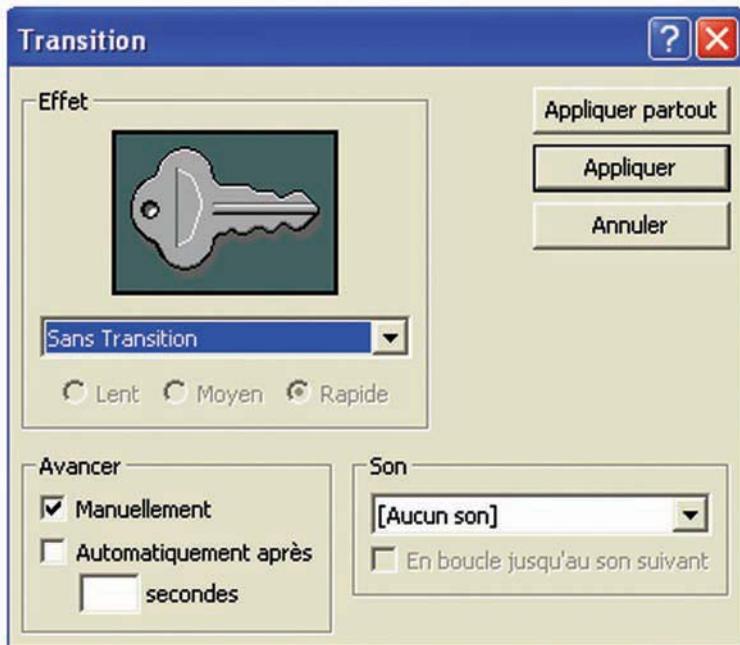


Fig.VI. 6 : Ajout d'une transition

I.3. Réalisation de présentations

a) Activité 1

- 1/ Lancez le logiciel de présentation disponible au laboratoire.
- 2/ Avec l'aide de votre enseignant, créez une présentation comportant quelques diapositives sur le thème " Architecture des ordinateurs ". Les diapositives comporteront du texte, des images, des commentaires, etc.

b) Réalisation

1. Lancez le logiciel de présentation.
2. Choisissez la forme de la diapositive (dans le cas de l'activité, choisir une forme comportant une zone titre, une zone texte et une zone d'insertion d'image).
3. Remplissez les différentes zones de la diapositive.
4. Enregistrez votre travail dans le dossier de travail.
5. Choisissez une nouvelle forme.
6. Remplissez les différentes zones
7. Enregistrez votre travail.
8. Refaites les étapes 5, 6 et 7.

c) Quelles sont les étapes de réalisation d'une présentation ?

1) Activité 2

Avec l'aide de votre enseignant, dégagez les principales étapes de la réalisation d'une présentation.

2) Les étapes

1. Lancez le logiciel de présentation.
2. Choisissez la forme de la diapositive
3. Remplissez les différentes zones de la diapositive.
4. Intégrez les animations, les insertions des objets, les commentaires, etc.
5. Enregistrez la présentation.

Refaire les étapes 2 à 5 jusqu'à avoir présenté tous les thèmes de votre présentation.

3) Conseils pour la conception d'une présentation

1. Le contenu est l'élément le plus important lors de la présentation d'un diaporama. Les autres outils tels que les animations et les transitions servent à mettre en valeur les éléments de la présentation.
2. L'utilisation occasionnelle de musique ou d'un effet sonore lors d'une transition ou d'une animation incite l'assistance à se concentrer sur la présentation. Toutefois, une grande fréquence des effets sonores risque de détourner l'attention de l'assistance.
3. Il est conseillé de tester le rythme de la présentation avant de l'utiliser. Le rythme aura des répercussions sur l'attention de l'assistance. Un rythme rapide fatiguera les auditeurs, un rythme lent les endormira.
4. Penser à évaluer l'impact visuel de vos diapositives. L'abondance de mots ou d'images sur une diapositive peut susciter une certaine distraction. Éviter d'utiliser des textes trop lents.

II. La production de pages Web

II.1. Introduction

Activité 3

- Utilisez un explorateur d'Internet.
- Tapez dans la barre des adresses l'adresse URL suivante : WWW.edunet.tn
- Activez le menu Affichage de votre explorateur.
- Choisissez la commande Source.

```

<html>

<head>
<meta http-equiv="Content-Language" content="fr">
<meta http-equiv="Content-Type" content="text/html; charset-windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Nouvelle page 1</title>
<meta name="Microsoft Theme" content="sumipntg 011">
</head>

<body>

<p align="center"><font size="6">Exemple d'une page WEB</font></p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
</u1>
    <li><b><i>Page 1</i></b></li>
    <li><b><i>Page 2</i></b></li>
    <li><b><i>Page 3</i></b></li>
<p>&nbsp;</p>
<p align="center">Microsoft Word</b>         | Les nouvelles versions du Word peuvent servir à créer des pages HTML. Pour cela il faut utiliser l'option enregistrer en HTML du menu fichier                                                   |
| <b>Microsoft Front Page</b>   | C'est l'éditeur spécialisé de Microsoft pour la conception de pages WEB et la gestion des sites web.                                                                                            |
| <b>Macromedia Dreamweaver</b> | Cet éditeur permet de créer des pages WEB mais aussi de créer des pages dynamiques. Il possède aussi un module qui permet de vérifier la compatibilité du code avec les différents navigateurs. |
| <b>Filemaker Homepage 3.0</b> | Créer des pages avec cet éditeur est très simple.                                                                                                                                               |
| <b>Composer de Mozilla</b>    | C'est un logiciel libre, complet, puissant et simple à utiliser.                                                                                                                                |

- Opérations de base d'un éditeur de création de pages WEB

– *Création d'un dossier et de la première page d'un site (Fig.VI. 9)* : Il s'agit de créer un dossier racine du site. Au lancement, le logiciel ouvre une page vierge à enregistrer sous le nom index.htm: c'est la page d'accueil du site.

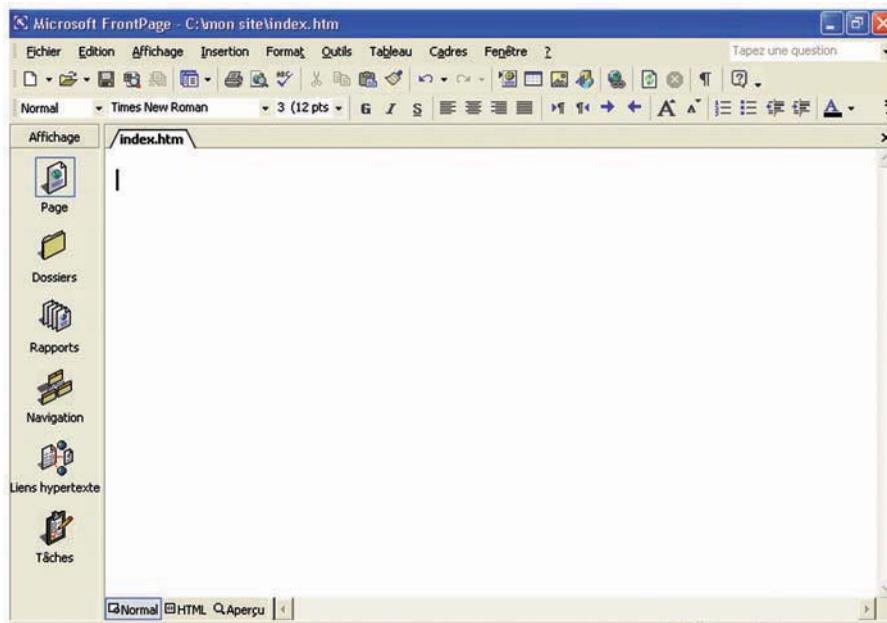


Fig.VI. 9 : Première page d'un site

– *Insertion de fichiers* : il suffit de placer le curseur à l'endroit de l'insertion du fichier et d'activer la commande d'insertion de fichiers en spécifiant le format voulu.

– *Insertion des images* : Il s'agit d'activer la commande d'insertion d'images et de spécifier la source à partir de la fenêtre qui sera active, exemple (Fig.VI. 10) :

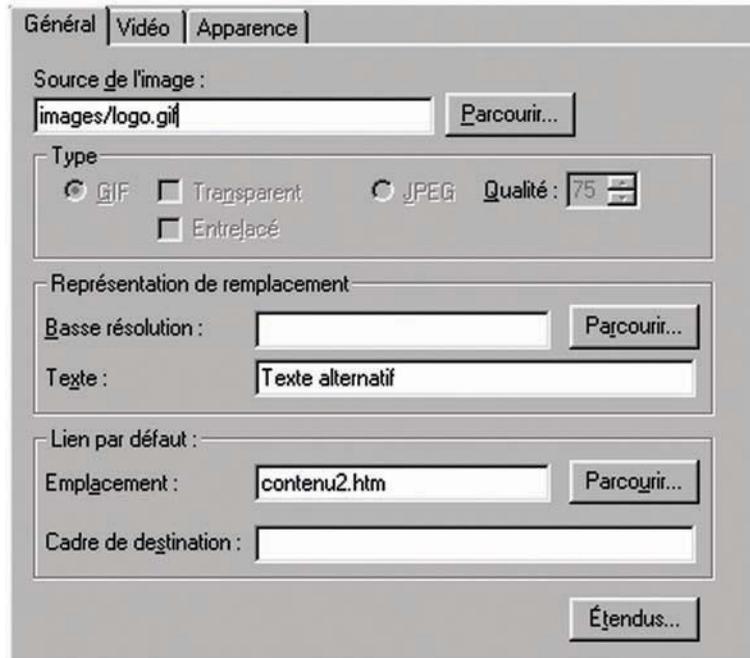


Fig.VI. 10 : Fenêtre permettant de définir la source de l'image à insérer

– *Insertion de tableaux* (Fig.VI. 11) : Il faut activer la commande d'insertion de tableaux puis définir le nombre de lignes et de colonnes nécessaires en précisant :

- le type d'alignement du tableau dans la page
- l'absence ou la présence d'une bordure
- l'espacement entre les cellules

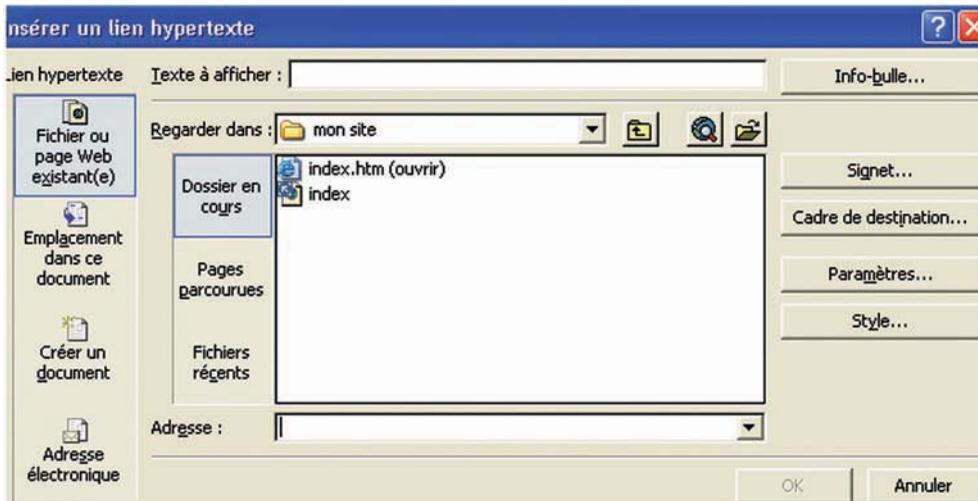


Fig.VI. 11 : Fenêtre permettant de définir la source de l'image à insérer

– *Insertion de liens* : une expression, un bouton ou une image peut être l'objet d'un lien. Il existe deux types de liens :

- Les liens internes : ils permettent à un utilisateur, par un simple click sur le lien, de se retrouver à un endroit prédéfini du document. Afin d'insérer un lien interne, il est nécessaire d'indiquer au navigateur deux endroit : le départ et la destination.

- Les liens externes : ils permettent de renvoyer l'utilisateur vers une autre page WEB; il suffit alors de spécifier l'adresse de la page destination (**Fig.VI. 12**).



**Fig.VI. 12** : Fenêtre permettant de définir un lien externe

– *Insertion de formulaires* : Activez la commande d'insertion de formulaires et choisissez le modèle qui vous convient.

### Etape 3 : Hébergement

Une fois le site est réalisé, il ne sera utile que s'il devient accessible. Pour cela il faut qu'il soit placé sur un serveur qui est connecté en permanence à Internet : c'est l'étape de l'hébergement (en anglais *Hosting*). On distingue l'hébergement gratuit et l'hébergement payant.

### Etape 4 : Se faire connaître

Il existe différents moyens afin que les internautes sachent que votre site a été hébergé et puissent le visiter :

- 1) Les moteurs de recherche / annuaires** : Le meilleur moyen de s'attirer les surfeurs, est d'être connu des moteurs de recherche/annuaires. Pour indexer un site, les moteurs de recherche se basent sur le contenu du site en lui-même : il faut choisir des titres attrayants.

- 2) Par mail** : La plupart des clients mails actuels vous permettent d'ajouter une signature à la fin de vos mails. Cette signature est en fait une suite de lignes dans lesquelles il vous suffit d'insérer l'adresse de votre site WEB, de façon à ce que vos correspondants puissent aller le visiter.

- 3) Dans les newsgroups** : Il existe des groupes de discussion spécialement conçus pour faire de la publicité.

### III - Retenons

- Une présentation est un moyen de communication basé sur l'assemblage de séquences de types texte, image, sonore,... utilisée essentiellement dans les domaines de la formation, de l'enseignement, ...
- Un logiciel de présentation permet à son utilisateur de créer des documents comportant des animations et dont les paramètres d'exploitation sont personnalisables.
- Pour créer une présentation, il est important de définir son thème et les contenus avant de commencer la réalisation des diverses diapositives.
- Un site Web est un ensemble de pages reliées par des liens de diverses catégories.
- Pour créer un site Web, il est possible d'utiliser le langage HTML ou l'un des éditeurs existants.
- Une page Web peut comporter différents éléments : texte, image, son, animations, etc.

## EXERCICES

### Exercice n° 1

Donnez les principales étapes de réalisation d'une présentation.

### Exercice n° 2

Utilisez un logiciel de présentation pour réaliser des diapositives décrivant vos sports préférés.

### Exercice n° 3

Votre enseignant d'informatique vous a chargé de réaliser un dossier dans lequel vous devez présenter les différentes matières étudiées en 2ème année de la filière Technologie de l'informatique.

Utilisez un logiciel de présentation pour réaliser ce dossier.

### Exercice n° 4

Dans le club de santé de votre établissement, l'animateur du club vous propose de collecter puis de présenter des données relatives au thème " Protection de l'environnement ".

1/ Utilisez Internet pour collecter ces informations.

2/ Réalisez une présentation à l'aide d'un logiciel de présentation.

### Exercice n° 5

Reprenez l'exercice n°2 et créez un site Web de vos sports préférés.

### Exercice n° 6

Reprenez l'exercice n°3 et créez un site Web présentant les différentes matières étudiées à raison d'une matière par page.

### Exercice n° 7

Reprenez l'exercice n°4 et créez un site Web présentant le thème "Protection de l'environnement".

### Exercice n° 8

Créez un site Web présentant votre établissement et comportant des images et des statistiques de l'évolution du nombre des enseignants et des élèves.

## CHAPITRE 7

# Introduction à la résolution de problèmes et à la programmation

### Objectifs

- Comprendre les différentes étapes de résolution d'un problème par ordinateur.
- S'approprier une méthodologie de résolution de problèmes à l'aide de l'ordinateur.

### Plan du chapitre

- I. Introduction
- II. Spécification et analyse d'un problème.
- III. Ecriture d'un algorithme
- IV. Traduction en un programme exécutable par ordinateur
- V. Exécutions et tests
- VI. Retenons

### Exercices

### Lectures

## CHAPITRE 7

# Introduction à la résolution de problèmes et à la programmation

## I. Introduction

Nous avons vu dans les chapitres précédents, plusieurs applications informatiques qui nous ont servis à réaliser différentes tâches. Pour pouvoir exploiter le micro-ordinateur, nous avons toujours utilisé un ensemble de programmes réunis dans une application appelée système d'exploitation. Dans un tel environnement, nous avons utilisé le logiciel MS Word pour produire un texte ou un document, MS PowerPoint pour réaliser une présentation, MS Internet Explorer pour naviguer sur Internet, une panoplie de logiciels pour traiter les images, les sons, les séquences vidéos, etc. Toutes ces applications sont conçues et réalisées pour des fins bien déterminées. Les auteurs de ces applications sont des particuliers ou des sociétés. Certaines applications sont gratuites, d'autres sont payantes.

Dans tous les cas, ces applications sont des solutions pour résoudre des problèmes donnés. Elles sont assez bien développées et ont demandé beaucoup d'efforts et de travail. Dans ce chapitre, nous allons comprendre comment fait-on pour réaliser de petites applications servant à résoudre de petits problèmes.

Partant d'un problème donné, on se propose de lui trouver une solution informatique. Cette solution peut être trouvée grâce à une application déjà conçue et essayée. Ce n'est pas le but de ce cours. L'objectif est de produire un programme permettant de résoudre ce problème. D'emblée, plusieurs questions vont se poser. Comment va-t-on rédiger une solution au problème posé. Comment va-t-on traduire la solution trouvée à l'ordinateur ? Quel langage choisir pour cette traduction ?

Ces questions vont dicter une certaine conduite à suivre dans le développement d'une solution. D'abord, il est impératif de bien reposer le problème dans tous ses détails afin de lever toute ambiguïté et toute ambiguïté. Une fois l'énoncé devenu clair, on commencera à développer une solution en passant par plusieurs étapes. Nous allons traiter ces étapes une à une dans les paragraphes qui vont suivre.

## II. Spécification et analyse du problème

C'est l'étape la plus importante dans la résolution d'un problème. Il existe plusieurs approches pour développer une solution à un problème donné. Les plus connues sont l'approche ascendante et l'approche descendante. Chacune d'elles pourra être améliorée en utilisant les modules qu'ils soient fonctions ou procédures. Mais regardons de près ces deux approches et essayons de comprendre l'état de pensée de chacune. Commençons par une activité.

### Activité 1

On se propose d'écrire un programme qui détermine et affiche la décision que le conseil d'une classe prendra pour chacun des élèves. La classe compte  $n$  élèves.

Nous constatons clairement que le conseil va entreprendre la répétition d'une action. L'action en question est d'évaluer un élève donné et décider de son cas. En effet, les membres vont traiter un à un tous les élèves de la classe et décider du passage et de la mention à attribuer.

Pour cette première approche, nous allons nous intéresser d'abord au résultat. Ce dernier apparaît comme une suite de décisions que nous allons appeler "Décisions".

Pour formuler ce résultat, nous allons utiliser des termes assez proches du langage naturel et traduisant des actions répétitives et sélectives. Nous obtenons la définition suivante du résultat "Décisions"

```
Décisions : Initialisation1,
 Pour i de 1 à n répéter
 {Décision concernant le ième élève}
```

Maintenant, regardons comment formuler ou écrire la décision du  $i$ ème élève. On nommera cette action "Décision".

```
Décision : Initialisation2
 si (moyenne_i <10) alors {decision = " redouble "}
 sinon si (moyenne_i <12) alors {decision = " admis – passable "}
 sinon si (moyenne_i <14) alors {decision = " admis – assez bien "}
 sinon si (moyenne_i <16) alors {decision = " admis – bien "}
 sinon {decision = " admis – très bien "}
```

Nous remarquons maintenant que nous avons besoin d'avoir la moyenne de l'élève  $i$  pour pouvoir décider de son passage. Cette moyenne peut être fournie avec toutes les autres moyennes au niveau de la première initialisation ou encore au moment où on commence le traitement du  $i$ ème élève donc au niveau de l'initialisation2. Vous déduisez par vous-même que Initialisation1 et Initialisation2 comporteront les actions nécessaires pour le bon déroulement des répétitions ou des sélections qui suivent. Nous allons convenir que toutes les moyennes ont été préparées au niveau du premier module d'initialisation. Ainsi, la définition de l'objet Décisions devient :

```
Décisions : Initialisation1,
 Pour i de 1 à n répéter
 Décision : Initialisation2
 si moyenne_i <10 alors {decision = " redouble "}
 sinon si moyenne_i <12 alors {decision = " admis – passable "}
 sinon si moyenne_i <14 alors {decision = " admis – assez bien "}
 sinon si moyenne_i <16 alors {decision = " admis – bien "}
 sinon {decision = " admis – très bien "}
 Fin si
 Fin Pour
```

Remarquons que les mots FinPour et FinSi servent respectivement à délimiter les actions de la répétition ou de la sélection.

Nous continuerons la définition de tous les éléments que nous découvrirons au fur et à mesure dans le développement et la spécification du problème. Nous arrêterons cette opération quand tous les objets évoqués seront entièrement définis.

Dans cette approche, vous avez vu que le développement d'une solution se fait d'une façon **systematique**. Nous commençons par le résultat et nous descendons pas à pas vers les actions élémentaires nécessaires. Cette approche va nous permettre d'aborder un à un tous les éléments qui entrent dans l'analyse du problème. Cette approche est appelée **analyse descendante**.

A l'encontre de cette approche, il existe une autre approche dite **ascendante**. Le principe de cette dernière consiste à préparer les objets et les actions jugés utiles pour arriver au résultat. Ensuite et pour développer la solution, nous devons trouver les liaisons adéquates pour assembler ces outils. Essayons d'appliquer cette méthode au même problème précédemment posé.

Avant de commencer, précisons que l'analyse à adopter dans la résolution d'un problème s'appuie sur un mode de pensée bien précis. Les deux approches descendante et ascendante sont deux modes de pensée totalement différents.

Oublions la précédente approche et abordons le problème d'une façon ascendante.

Nous allons d'abord préparer les notes et faire le calcul de la moyenne Moy. L'action pourra être réduite à une simple saisie de celle-ci. Nous allons traduire ceci par les termes :

**Lire (Moy)**

Ayant cette moyenne, comment allons-nous décider du sort d'un élève ?

Traduisons cette action par :

```

si Moy <10 alors {decision = " redouble "}
sinon si Moy <12 alors {decision = " admis – passable "}
sinon si Moy <14 alors {decision = " admis – assez bien "}
sinon si Moy <16 alors {decision = " admis – bien "}
sinon {decision = " admis – très bien "}
Fin si

```

Cette action étant prête, essayons d'en faire une répétition.

```

Pour i de 1 à n répéter
 si (Moy <10) alors {decision = " redouble "}
 sinon si (Moy <12) alors {decision = " admis – passable "}
 sinon si (Moy <14) alors {decision = " admis – assez bien "}
 sinon si (Moy <16) alors {decision = " admis – bien "}
 sinon {decision = " admis – très bien "}
 Fin si
Fin Pour

```

On continuera l'ajout des instructions traduisant la saisie des données et éventuellement une autre action d'affichage du résultat pour aboutir à la solution.

Nous avons vu deux approches différentes du point de vue raisonnement. Bien entendu, il est clair que le but est de trouver une solution au problème posé.

### La question maintenant est laquelle choisir des deux ?

En partant du fait que le but de cet apprentissage est le développement chez l'apprenant d'un mode de pensée structurée et d'une systématisation dans l'élaboration d'une solution à un problème donné, nous pencherons vers la méthode descendante. Cette méthode a l'avantage de rendre systématique la recherche d'une solution. Nous ne ferons pas une exclusion de l'autre approche, bien au contraire, nous verrons que dans certains problèmes, l'approche ascendante est beaucoup plus facile à appliquer. A ces deux approches, nous ajouterons à la fin de ce cours, l'approche modulaire qui consiste à diviser la solution en modules paramétrables, en l'occurrence, les fonctions et les procédures.

### III. Ecriture d'un algorithme

Une fois la spécification achevée, nous allons réécrire dans l'ordre chronologique d'exécution toutes les actions évoquées. Nous obtiendrons une suite finie d'actions ou d'instructions. Une telle suite est appelée **algorithme**.

Le mot algorithme provient du nom du mathématicien arabe : **Mohamed Ibn Hassen El Khawarizmi**. Il est né en 750 à Khawarism, un village perse et est mort en 840. On lui doit aussi le mot "algèbre" de son livre "Al Jabr wal Moukabalah".

Que ce soit dans la spécification ou encore plus dans l'algorithme, nous avons besoin de faire des conventions d'écriture et de notation pour rédiger la solution cherchée. Généralement, ce genre de notation est assez proche du langage naturel mais assez restrictif et précis.

#### III.1. Notation algorithmique

Comme indiqué précédemment, il est très difficile d'utiliser le langage naturel pour développer un algorithme compte tenu de la diversité des sens et la multitude des synonymes et la pluralité sémantique des expressions. On convient alors de définir un vocabulaire réduit et précis servant à noter les différentes actions pour écrire un algorithme. Comme pour tout langage ou toute notation, on définira une sémantique claire et une grammaire précise et stricte.

#### Exemples de notation

Pour traduire la saisie d'une note, nous donnons un nom à cette note, N1 par exemple, et nous traduisons la saisie par :

**Lire (N1)**

Cette notation algorithmique signifie que le programme, lors de son exécution, demande à l'utilisateur de saisir un nombre qu'il mettra dans le contenant **N1**. Retenez pour l'instant que **N1** est un case mémoire ou un endroit permettant de garder ce réel.

Au cours de l'analyse, on pourra se trouver contraint de définir une inconnue V qui a figuré dans une expression précédente. Quand on sait que V est une donnée que l'utilisateur devra fournir au programme, on traduira ceci par :

**V = DONNEE**

Cette écriture a la même signification que : **Lire (V)**. C'est uniquement le contexte au cours de l'analyse qui permettra le choix entre la notation **Lire** et la notation **DONNEE**.

Regardons maintenant, la notation servant à afficher quelque chose à l'écran. Le verbe couramment utilisé est :

### Ecrire

Ce verbe convient bien avec d'autres supports de sortie tels que l'imprimante et les fichiers.

### Activité 2

On se propose de calculer et d'afficher la surface d'un disque de rayon donné.

Nous allons voir comment écrire la spécification de ce problème et comment à partir de cette dernière, nous déduisons l'algorithme solution. Nous allons utiliser pour l'instant les deux notations de saisie et d'affichage vues précédemment.

### Spécification du problème

Le résultat de ce problème est l'affichage de la surface que nous allons nommer **Surface**.

#### Ecrire (Surface)

Dans cette action, **Surface** est un objet à définir.

$$\text{Surface} = \pi \cdot \text{Rayon} \cdot \text{Rayon}$$

Dans cette dernière expression, l'objet Rayon est à définir.

$$\text{Rayon} = \text{DONNEE}$$

Maintenant, tous les objets sont entièrement définis, nous allons écrire dans un ordre logique d'exécution toutes les actions et les définitions citées dans la spécification. Ceci donnera d'une façon **systématique** l'algorithme solution du problème posé. Appelons cet algorithme Calcul\_surface.

#### Algorithme :

- 0) Début Calcul\_surface
- 1) Lire (Rayon)
- 2) Surface  $\leftarrow \pi \cdot \text{Rayon} \cdot \text{Rayon}$
- 3) Ecrire (Surface)
- 4) Fin Calcul\_surface

Comme vous l'avez remarqué, nous allons convenir de donner un nom à chaque algorithme. Chaque algorithme commence par le mot Début et se termine par le mot Fin.

### Activité 3

Ecrivez un programme qui calcule votre moyenne trimestrielle en informatique. Les données sont la note du devoir de contrôle et celle du devoir de synthèse. On suppose qu'elles ont respectivement un coefficient 1 et 2.

### Spécification du problème

Résultat : Ecrire (Moy)  
 $Moy = (DC + 2.DS) / 3$   
 DC = DONNEE  
 DS = DONNEE

Nous nous sommes abstenus d'expliquer littéralement les objets comme nous l'avons fait dans l'activité précédente vu la clarté du problème. Nous ajouterons des explications quand nous en verrons l'utilité.

#### Algorithme :

- 0) Début Moyenne
- 1) Lire (DC)
- 2) Lire (DS)
- 3)  $Moy \leftarrow (DC + 2. DS)/3$
- 4) Ecrire (Moy)
- 5) Fin Moyenne

### Activité 4

□ Ecrire un programme qui conjugue au futur, un verbe régulier du premier groupe.

### Spécification

Un verbe régulier du premier groupe se termine par "er". Pour le conjuguer au futur, nous allons ajouter la terminaison adéquate suivant la personne.

Résultat : Conjugaison

conjugaison =      Ecrire ("Je ", verbe, "ai")  
                          Ecrire ("Tu ", verbe, "as")  
                          Ecrire ("Il/elle ", verbe, "a")  
                          Ecrire ("Nous ", verbe, "ons")  
                          Ecrire ("Vous ", verbe, "ez")  
                          Ecrire ("Ils ", verbe, "ont")

verbe = DONNEE

Fin

Nous savons que le résultat est composé de plusieurs affichages, nous lui avons donné le nom **Conjugaison** puis nous l'avons développé juste après. Les textes mis entre guillemets sont des messages et ils seront affichés tels quels. L'objet **verbe** contiendra le verbe régulier du premier groupe, saisi par l'utilisateur.

#### Algorithme :

- 0) Début Futur
- 1) Lire (verbe)
- 2) Ecrire ("Je ", verbe, "ai")
- 3) Ecrire ("Tu ", verbe, "as")
- 4) Ecrire ("Il/elle ", verbe, "a")
- 5) Ecrire ("Nous ", verbe, "ons")
- 6) Ecrire ("Vous ", verbe, "ez")
- 7) Ecrire ("Ils/elles ", verbe, "ont")
- 8) Fin Futur

## IV. Traduction en un programme exécutable par ordinateur

Les deux étapes précédentes sont les plus importantes dans la résolution d'un problème. Il reste cependant les dernières étapes liées étroitement à l'ordinateur. Il faudra trouver un moyen de rendre l'algorithme solution compréhensible par la machine. L'opération consiste à écrire cet algorithme dans un langage de programmation. Il en existe des centaines, mais quelques uns, de part leurs structures et leurs vocations sont plus connues que d'autres. Nous en citons quelques-uns :

**COBOL** (COmmun Business-Oriented Language)

**FORTRAN** (FORmula TRANslation)

**BASIC** (Beginners All-purpose Symbolic Instruction Code)

**PASCAL** (c'est un hommage au savant Blaise Pascal)

**C**

**ADA** (C'est un hommage à Ada Lovelace, la 1ère femme programmeur)

Le langage **Pascal** est assez structuré, très pédagogique et se prête bien pour l'enseignement. Il permet une traduction simple des algorithmes que nous allons développer. C'est pour cette raison que nous allons l'utiliser pour écrire nos programmes. De temps à autres, nous donnerons des traductions en C pour le besoin de la section Informatique industrielle.

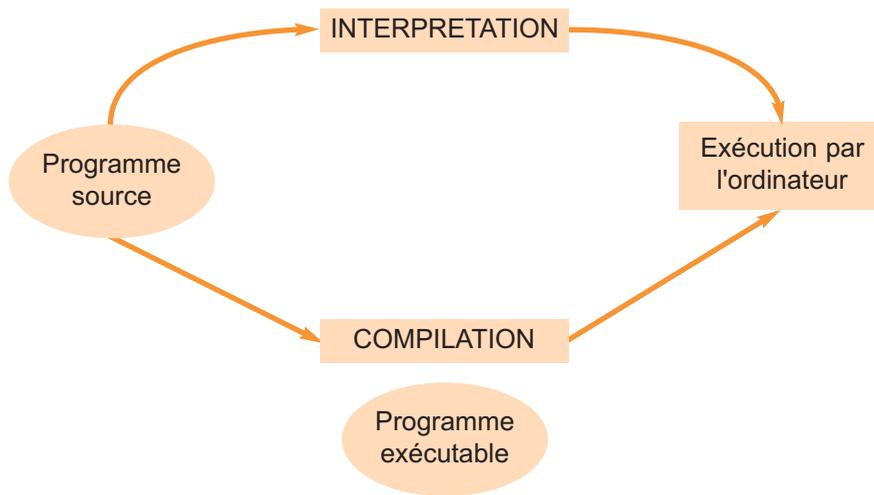
### IV.1. Comment écrire un programme ?

Une fois l'algorithme établi et testé, on commence à penser traduction et codage en un langage de programmation. Pour ce dernier, il faudra connaître le vocabulaire, la syntaxe, les instructions, les représentations des données et leurs contraintes, les différents modules ou unités constituantes, etc. Ces connaissances vont permettre de traduire un algorithme pour avoir le programme source. Généralement, chaque langage est muni d'un éditeur de textes qu'on pourra utiliser pour développer les programmes sources. Le programme développé est sous la forme d'un fichier.

### IV.2. L'interprétation ou la compilation

Pour exécuter le programme écrit, il faut encore le traduire en langage machine. C'est ce dernier qui est exécutable par la machine. Il existe deux manières de traduire en langage machine. Soit on traduit et on exécute au fur et à mesure les différentes instructions du programme source. On dit qu'on fait de l'interprétation ou encore que le langage est interprété. Cette méthode a l'inconvénient de la lenteur et du plantage à la première erreur.

Soit on traduit le programme source en un autre programme dit exécutable et il s'agit dans ce cas de la compilation ou encore que le langage de programmation est compilé. Lors de la compilation, le traducteur révèle toutes les erreurs de vocabulaire et de syntaxe et permet donc de les corriger.



Nous citons comme exemple de langages interprétés le BASIC et comme exemples de langages compilés le Pascal et le C.

Autre élément important à prendre en compte et qui dépend essentiellement du langage est la représentation des données. Nous savons que pour exécuter un programme, il faudra l'installer dans la mémoire vive ou la RAM. Et quelque soit la taille de la RAM, elle restera d'une grandeur finie. Nous comprenons par là que les données numériques avec leurs divers types ainsi que les données alphanumériques sont représentées par des grandeurs finies ou limitées. C'est d'ailleurs pour cette raison que la majorité des langages exigent à priori une identification de la structure et du type des données. Nous reviendrons plus en détail sur ces points dans le prochain chapitre.

Suite à ce que nous venons d'avancer, il est donc nécessaire de dresser un tableau de codification des objets. Il comporte pour un objet, le nom, le code ou encore le nom qu'on va lui attribuer dans le programme, nous sommes parfois obligés de coder autrement l'objet pour ne pas interférer avec des mots réservés au programme, le type, et éventuellement le rôle. Le type va permettre au langage de préparer l'espace mémoire adéquat à cet objet.

Voici un exemple de tableau de codification des variables.

| NOM | CODE | TYPE | ROLE    |
|-----|------|------|---------|
| MO  | MO   | réel | Moyenne |
| TO  | TOT  | réel | Total   |

### Activité 5

Nous allons reprendre l'algorithme de l'activité et le traduire en Pascal, en BASIC et en C.

Commençons par le tableau de codification des variables.

| NOM | CODE | TYPE | ROLE                       |
|-----|------|------|----------------------------|
| DC  | DC   | réel | note du devoir de contrôle |
| DS  | DS   | réel | note du devoir de synthèse |
| Moy | DC   | réel | moyenne trimestrielle      |

### Traduction en BASIC

```
10 REM Moyenne
20 INPUT DC
30 INPUT DS
40 Moy=(DC + 2*DS)/3
50 PRINT Moy
60 END
```

### Traduction en PASCAL

```
PROGRAM Moyenne;
USES WINCRT;
VAR DC, DS, Moy : REAL;
BEGIN
 CLRSCR;
 READLN(DC);
 READLN(DS);
 Moy := (DC + 2*DS)/3;
 WRITELN(Moy);
END.
```

### Traduction en C

```
#include <stdio.h>
void main();
{
 float DC, DS;
 float Moy;
 scanf("%f",DC);
 scanf("%f",DS);
 moy = (DC + 2*DS)/3;
 printf("%f",Moy);
}
```

### Activité 6

Reprenez l'algorithme de la conjugaison d'un verbe régulier du premier groupe vu dans l'activité 4 et traduisez-le en Pascal et C.

### Traduction en Pascal

```
PROGRAM Futur;
uses WINCRT;
VAR verbe : STRING;
BEGIN
 READLN(verbe);
 WRITELN('Je ', verbe, 'ai');
 WRITELN('Tu ', verbe, 'as');
 WRITELN('Il/elle ', verbe, 'a');
 WRITELN('Nous ', verbe, 'ons');
 WRITELN('Vous ', verbe, 'ez');
 WRITELN('Ils/elles ', verbe, 'ont');
END.
```

## Traduction en C

```
#include <stdio.h>
void main();
{
char verbe[20];

scanf ("%s",verbe);
printf ("Je ",verbe,"ai");
printf ("Tu ",verbe,"as");
printf ("Il/elle ",verbe,"a");
printf ("Nous ",verbe,"ons");
printf ("Vous ",verbe,"ez");
printf ("Ils/elles ",verbe,"ont");
}
```

## V. Exécutions et tests

Une fois l'écriture du programme source achevée, nous allons procéder aux exécutions. Ces opérations consistent à tester le programme obtenu et s'assurer qu'il répond bien à la question et fait effectivement ce que nous lui demandons de faire. Pour cela, il est nécessaire de préparer un jeu de tests bien choisis et de les essayer à ce programme. Bien entendu, cette étape devra permettre de corriger les éventuelles erreurs et de s'assurer de la justesse de ce programme.

### Activité 7

Saisir le programme Pascal de l'activité 6 relatif à la conjugaison au futur. Sauvegarder le code source sous le nom futur.pas. Formez le programme exécutable futur.exe.

Tester votre programme pour les verbes : marcher, chanter, manger, nager, aller.

## VI - Retenons

- 1** - La spécification d'un problème est par définition l'interprétation exacte et non ambiguë de ce que le programme devra réaliser. Elle comporte les structures de tous les objets évoqués en commençant par le résultat du problème et en adoptant une approche descendante. Ces structures concernent les données et les actions.
- 2** - Un algorithme est une suite ordonnée et finie d'actions ou d'instructions servant à résoudre un problème donné.
- 3** - Les programmes sources écrits à l'aide d'un langage compilé sont des textes non exécutables. Il faut les compiler pour obtenir les programmes exécutables correspondants.

## EXERCICES

### Exercice n° 1

Définir la démarche descendante.

### Exercice n° 2

Qu'est ce qu'un algorithme ?

### Exercice n° 3

Quelle est la notation algorithmique qui traduit la saisie d'une donnée via le clavier ?

### Exercice n° 4

Quelle est la notation traduisant une sortie de données à l'écran ?

### Exercice n° 5

Quelles sont les étapes à faire pour résoudre un problème à l'aide de l'ordinateur ?

### Exercice n° 6

Citer au moins trois langages de programmation ?

### Exercice n° 7

Qu'appelle-t-on compilation et interprétation ?

**Donner une spécification et déduire un algorithme solution à chacun des problèmes suivants :**

### Exercice n° 8

On se propose de convertir en heures, minutes et secondes, un temps donné en secondes.

### Exercice n° 9

Ecrivez un programme qui conjugue un verbe régulier du premier groupe à l'imparfait.

### Exercice n° 10

Décrivez avec exactitude le chemin pour aller à pied du lycée à la municipalité ou à un autre lieu connu de la ville.

### Exercice n° 11

Déterminez est afficher le plus petit de deux nombres donnés.

### Exercice n° 12

Déterminez si trois points donnés sont sur une même ligne.

### Exercice n° 13

On dispose de deux récipients contenant respectivement de l'huile et du lait. On se propose de mettre le lait dans le récipient qui contenait l'huile et réciproquement. Décrivez en détails les étapes à faire et les éventuels objets nécessaires.

### Exercice n° 14

Résolvez une équation du 1<sup>er</sup> degré dans l'ensemble des réels.

## Lecture

---

*"La programmation est l'art d'utiliser la puissance des ordinateurs pour déguiser en intelligence leur extrême stupidité". Cette formule due à B. Meyer et C. Baudoin est très significative.*

*Il est bien vrai en effet que l'ordinateur n'est rien d'autre qu'un automate capable d'exécuter des actions (opérations) dites élémentaires. Il ne possède d'ailleurs qu'un répertoire limité de telles actions. La puissance de l'ordinateur et l'étude de ses domaines d'applications reposent essentiellement sur la possibilité qu'il a d'exécuter automatiquement des suites d'opérations extrêmement longues. Un programme est un ensemble d'instructions pour commander l'exécution de telles suites.*

*Lorsque la suite d'opérations élémentaires permettant de résoudre un problème atteint une certaine longueur et une certaine complexité (plusieurs milliers, voire plusieurs millions d'opérations), la conception du programme devient extrêmement difficile. Grâce à la puissance des ordinateurs, il est en effet possible aujourd'hui de chercher à résoudre des problèmes d'une taille bien supérieure à tout ce qui était simplement imaginable auparavant. L'esprit humain est mal préparé à maîtriser leur complexité. L'obligation de spécifier d'une manière extrêmement précise et rigoureuse le déroulement pas à pas d'un "traitement de l'information" entraîne la difficulté de la programmation. **Beaucoup plus que l'apprentissage de tel ou tel langage particulier, la programmation nécessite l'acquisition d'un mode de pensée nouveau.***

*La programmation, un art plutôt qu'une science ? Oui sans doute, dans la mesure où à l'heure actuelle, il n'est guère possible de dégager des règles absolues pour écrire de "bons programmes".*

*S'il nous faut définir la programmation, nous dirons qu'elle est l'art de concevoir et d'exprimer des algorithmes. C'est un exercice difficile. Il existe cependant des méthodes pour systématiser certains aspects, méthodes que tout programmeur devrait connaître....*

*L'apparition des premiers langages de programmation fut un tournant décisif dans le développement de l'informatique. Leur rapide succès eut cependant une conséquence néfaste : la programmation fut pendant un temps (trop long, hélas) considérée comme la codification d'algorithmes dans un langage donné, certains langages étant conçus pour la gestion, d'autres pour les calculs scientifiques, d'autres enfin pour des domaines encore plus spécialisés. En fait, quel que soit le domaine d'application, les objets élémentaires à traiter et les idées fondamentales pour la conception des programmes sont les mêmes ; ce sont ces idées qu'il est important d'enseigner. **De même que l'utilisation d'un langage de programmation permet de s'affranchir de telle ou telle machine particulière, il nous semble fondamental de présenter les concepts élémentaires de la programmation indépendamment de tout langage particulier.***

J.-P. LAURENT

Initiation à l'analyse et à la programmation

DUNOD

## CHAPITRE 8

# Les structures de données

### Objectifs

- Définir les notions de constante et de variable
- Déclarer des variables de type simple, de type chaîne de caractères et de type tableau
- Distinguer les types simples du type tableau

### Plan du chapitre

- I. Les constantes
- II. Les variables
- III. Les types de données
- IV. Retenons

### Exercices

## CHAPITRE 8

# Les structures de données

Dans ce chapitre, vous allez apprendre à manipuler les données à utiliser pour résoudre un problème. En plus des notions de variable et de constante, vous allez découvrir les types des données les plus utilisés.

## I. Les constantes

### Activité 1

Dans le tableau ci-dessous, vous trouverez des constantes connues et utilisées dans plusieurs disciplines, donnez des valeurs approchées de celles-ci :

| Constantes                  | Valeurs approchées |
|-----------------------------|--------------------|
| La constante <b>PI</b>      |                    |
| Charge élémentaire <b>q</b> |                    |
| Masse de l'électron         |                    |
| Masse du proton             |                    |
| Masse du neutron            |                    |
| Nombre d'or                 |                    |

### I.1. Définition

Une constante est une donnée dont on décide de garder la valeur inchangée tout le long d'un algorithme.

### I.2. Déclaration d'une constante

- Déclaration algorithmique

| Objet                       | Type / Nature                             | Rôle |
|-----------------------------|-------------------------------------------|------|
| Identifiant de la constante | <b>Constante</b> = valeur de la constante | Rôle |

#### Exemple

| Objet | Type / Nature           | Rôle |
|-------|-------------------------|------|
| PI    | <b>Constante</b> = 3,14 |      |

• **Déclaration en Turbo Pascal**

```
CONST
 Identificateur_constante = valeur_constante ;
```

**Exemple**

```
CONST
 PI=3.14 ;
```

La constante est caractérisée par un nom et un type. Son type est celui de la valeur qui lui est attribuée.

**I.3. Application**

Soit à calculer et à afficher le périmètre d'un cercle de rayon R. Présentez la spécification de ce problème.

**II. Les variables**

**Activité 2**

Pour chaque rayon donné, calculez le périmètre du cercle correspondant :

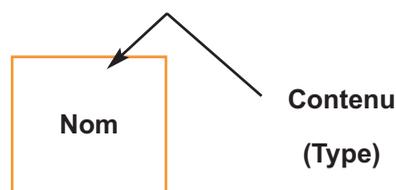
| PI   | R    | P |
|------|------|---|
| 3.14 | 2.5  |   |
| 3.14 | 1.25 |   |
| 3.14 | 4.5  |   |

On constate que PI est une constante tandis que R et P prennent des valeurs différentes. P et R sont appelés des **variables**.

**II.1. Définition**

Une variable est un objet pouvant prendre différentes valeurs lors de l'exécution d'un algorithme. Une variable est caractérisée par :

- son identificateur (nom)
- son type
- son contenu



**Exemple**

| Identificateur | Contenu |
|----------------|---------|
| Rayon          | 5.2     |
| Moyenne        | 13.45   |
| Num_mois       | 5       |
| Nom            | “Salah” |
| Voyelle        | “a”     |

**II.2. Déclaration d’une variable**

• **Déclaration algorithmique**

| Objet                      | Type / Nature       | Rôle                                  |
|----------------------------|---------------------|---------------------------------------|
| Identifiant de la variable | Type de la variable | Rôle de la variable dans l’algorithme |

**Exemple**

| Objet | Type / Nature | Rôle                |
|-------|---------------|---------------------|
| P     | Réel          | Périmètre du cercle |
| R     | Réel          | Rayon du cercle     |

• **Déclaration en Turbo Pascal**

VAR

Identificateur\_variable : Type\_variable ;

**Exemple**

VAR

P,R :REAL ;

**III. Les types de données**

Une variable est caractérisée par son nom, son type et son contenu. Le type permet de déterminer :

- l’ensemble des valeurs qui peuvent être affectées à la variable en question.
- l’ensemble des opérations qu’on peut appliquer sur cette variable.

Un type est désigné par un identificateur (nom).

**Exemple**

| Identificateur | Contenu | Type                 |
|----------------|---------|----------------------|
| Rayon          | 5.2     | Réel                 |
| Moyenne        | 13.45   | Réel                 |
| Num_mois       | 5       | Entier               |
| Nom            | “Salah” | Chaîne de caractères |
| Voyelle        | “a”     | Caractère            |

### III.1. Le type entier

#### a) Le domaine des valeurs du type entier

L'ensemble des valeurs du type entier dépend des types employés dans le langage de programmation. En effet, le langage Turbo Pascal fournit cinq types entiers prédéfinis. Chacun d'eux concerne un sous-ensemble particulier des nombres entiers. Le type entier est exprimé sous différentes formes :

| Type relatif à l'entier | Domaine des valeurs             |
|-------------------------|---------------------------------|
| Byte                    | 0..255                          |
| Word                    | 0..65 535                       |
| Longint                 | -2 147 483 648 .. 2 147 483 647 |
| Shortint                | -128..127                       |
| Integer                 | -32768 .. 32767                 |

#### Remarque

Afin de faciliter le travail et éviter toute incompatibilité entre la solution algorithmique et sa traduction en Turbo Pascal, on supposera qu'au type **entier**, employé en algorithmique, lui correspondra le type **Integer** en Turbo pascal. Néanmoins, il est conseillé d'employer lors de la traduction de l'algorithme le type entier convenable.

#### Exemple

Si on demande à la machine de calculer la factorielle de 8 (8!).

Par définition,  $8! = 1*2*3*4*5*6*7*8 = 40320$

D'après la valeur trouvée, est-il judicieux de choisir le type **Integer** comme type du résultat trouvé ?

On sait que le domaine des valeurs du type **Integer** est : -32768 .. 32767. On voit bien que le résultat 40320 dépasse la borne supérieure de l'intervalle. Prendre le résultat dans une variable entière provoque **un problème de débordement** qui engendre des calculs erronés. Le type convenable, dans ce cas, est **Longint**.

#### b) Les opérateurs arithmétiques sur les entiers

Sur des opérands de type entier, on peut appliquer les opérateurs arithmétiques suivants :

| Opérateur arithmétique | Type du résultat |
|------------------------|------------------|
| +                      | ENTIER           |
| -                      | ENTIER           |
| *                      | ENTIER           |
| /                      | REEL             |

A part ces opérateurs arithmétiques usuels, il existe deux autres opérateurs assez intéressants intitulés **MOD** et **DIV** définis comme suit :

**MOD** : donne le reste de la division entière du premier opérande par le deuxième opérande.

**Exemple** : (15 MOD 7 = 1).

**DIV** : donne le quotient de la division entière du premier opérande par le deuxième opérande.

**Exemple** : (15 DIV 7 = 2).

Ces deux opérateurs sont binaires.

|            |               |
|------------|---------------|
| <b>MOD</b> | <b>ENTIER</b> |
| <b>DIV</b> | <b>ENTIER</b> |

### c) Les opérateurs relationnels sur les entiers

On peut appliquer les opérateurs relationnels usuels en adoptant les notations suivantes :

| Notation algorithmique | Notation Turbo Pascal |
|------------------------|-----------------------|
| =                      | =                     |
| ≠                      | <>                    |
| ≤                      | <=                    |
| <                      | <                     |
| ≥                      | <=                    |
| >                      | >                     |

### d) La déclaration d'une variable entière

**Tableau de déclaration des objets**

| Objet        | Type / Nature | Rôle |
|--------------|---------------|------|
| Nom-variable | ENTIER        | Rôle |

**Déclaration en Turbo Pascal**

**VAR**

Nom-variable : INTEGER ;

### Activité 3

Présentez le tableau de déclaration des objets pour y déclarer des variables de type entier en spécifiant leurs rôles. Traduisez ces déclarations en Turbo Pascal.

## III.2. Le type réel

### a) Le domaine des valeurs du type réel

Un type réel possède un ensemble de valeurs qui engendrent un sous ensemble des nombres réels. Turbo Pascal en fournit cinq types différents. Chacun d'eux a un domaine de valeurs spécifique :

| Type relatif au réel | Domaine des valeurs |
|----------------------|---------------------|
| Real                 | 2.9E-39..1.7E38     |
| Single               | 1.5E-45..3.4E38     |
| Double               | 5.0E-324..1.7E308   |
| Extended             | 3.4E-4932..1.1E4932 |
| Comp                 | -9.2E18..9.2E18     |

**Remarques**

- Afin de faciliter le travail et d'éviter toute incompatibilité entre la solution algorithmique et sa traduction en Turbo pascal, on supposera qu'au type **Réel**, employé en algorithmique, lui correspondra le type **Real** en Turbo Pascal.
- La borne inférieure du type REAL 2.9E-39 n'est rien d'autre que  $2.9 \times 10^{-39}$ . En effet, la lettre E se lit "dix puissance" ou "dix exposant", 2.9 constitue la **mantisse** et -39 représente l'**exposant**.

**b) Les opérateurs applicables sur les réels**

- Les opérateurs arithmétiques sur les réels sont les mêmes que sur les entiers exceptés MOD ET DIV. Ces derniers ne sont appliqués qu'avec des entiers..
- Les opérateurs relationnels sont applicables sur des réels.

**c) Déclaration d'une variable réelle**

Tableau de déclaration des objets

| Objet        | Type / Nature | Rôle |
|--------------|---------------|------|
| Nom-variable | REEL          | Rôle |

Déclaration en Turbo Pascal

VAR

Nom-variable : REAL;

**Activité 4**

Présentez le tableau de déclaration des objets pour y déclarer des variables de type réel en spécifiant leurs rôles. Traduisez ces déclarations en Turbo Pascal.

**d) Les fonctions arithmétiques standard**

La bibliothèque Turbo Pascal, comme celle de la plupart des langages de programmation, est riche en fonctions arithmétiques dont voici les plus utilisées :

| Fonction | Rôle                                                                           | Type du paramètre x | Type du résultat  | Exemples                                                      |
|----------|--------------------------------------------------------------------------------|---------------------|-------------------|---------------------------------------------------------------|
| TRUNC(x) | Tronque une valeur réelle en valeur entière.                                   | Réel                | Entier            | TRUNC(5.27) vaut 5<br>TRUNC(-5.27) vaut -5                    |
| ROUND(x) | Arrondit une valeur réelle à l'entier le plus proche.                          | Réel                | Entier            | ROUND(7.49) vaut 7<br>ROUND(7.5) vaut 8<br>ROUND(7.99) vaut 8 |
| ABS(x)   | Renvoie la valeur absolue de x.                                                | Réel/Entier         | Type du paramètre | ABS(-10) vaut 10                                              |
| SQR(x)   | Renvoie le carré de x.                                                         | Réel ou Entier      | Type du paramètre | SQR(5) vaut 25                                                |
| SQRT(x)  | Renvoie la racine carrée de x s'il est positif sinon elle provoque une erreur. | Réel ou Entier      | Réel              | SQRT(5) vaut 2.236...                                         |
| INT(x)   | Renvoie la partie entière de x.                                                | Réel                | Réel              | INT(5.26) vaut 5.00                                           |
| FRAC(x)  | Renvoie la partie fractionnaire de x.                                          | Réel                | Réel              | FRAC(5.26) vaut 0.26                                          |
| COS(x)   | Renvoie le cosinus de x (x est en radians).                                    | Réel                | Réel              | COS(PI) vaut -1.00                                            |
| SIN(x)   | Renvoie le sinus de x (x est en radians).                                      | Réel                | Réel              | SIN(PI) vaut 0.00                                             |

**Activité 5**

Ecrivez les expressions suivantes en Turbo Pascal

| Expressions                             |
|-----------------------------------------|
| $7x^2+2x-1$                             |
| $-(b+\sqrt{\Delta})/(2a)$               |
| $-11x^3/7y^2$                           |
| $\text{Cos}(\pi - 1)+\text{Sin}(\pi/2)$ |

**III.3. Le type booléen**

**Définition**

Une proposition est une phrase qui est soit vraie soit fausse. On l'appelle aussi expression booléenne ou logique.

**a) Domaine des valeurs du type booléen**

Le type booléen comporte deux valeurs VRAI (TRUE en Turbo Pascal) et FAUX (FALSE en Turbo Pascal).

On définit sur l'ensemble des propositions divers opérateurs booléens.

**b) Opérateurs logiques sur les booléens**

Ces opérateurs sont les suivants :

| Notation algorithmique | Notation en Turbo Pascal | Rôle        |
|------------------------|--------------------------|-------------|
| <b>NON</b>             | <b>NOT</b>               | Négation    |
| <b>ET</b>              | <b>AND</b>               | Conjonction |
| <b>OU</b>              | <b>OR</b>                | Disjonction |
| <b>OUex</b>            | <b>XOR</b>               | Ou exclusif |

Les tables de vérités de ces opérateurs sont récapitulées dans le tableau ci-dessous :

| Expression logique A | Expression logique B | NON(A) | A ET B | A OU B | A OUex B |
|----------------------|----------------------|--------|--------|--------|----------|
| VRAI                 | VRAI                 | FAUX   | VRAI   | VRAI   | FAUX     |
| VRAI                 | FAUX                 | FAUX   | FAUX   | VRAI   | VRAI     |
| FAUX                 | VRAI                 | VRAI   | FAUX   | VRAI   | VRAI     |
| FAUX                 | FAUX                 | VRAI   | FAUX   | FAUX   | FAUX     |

**NB :** Remarquons que l'opérateur NON nécessite un seul opérande, on dit qu'il est unaire par contre, tous les autres utilisent deux opérandes, on dit qu'ils sont binaires.

**Activité 6**

Evaluez les propositions suivantes :

| Propositions                                                               |
|----------------------------------------------------------------------------|
| $(10 < 5) \text{ ET } (8 > 0)$                                             |
| $(\text{ROUND}(7.5)=7) \text{ OU } (\text{SQRT}(2)>2)$                     |
| $(\text{TRUNC}(5.26)=5) \text{ ET } (\text{NON}(\text{SIN}(\text{PI})>1))$ |

c) Déclaration d'une variable booléenne

Tableau de déclaration des objets

| Objet        | Type / Nature | Rôle |
|--------------|---------------|------|
| Nom-variable | BOOLEEN       | Rôle |

Déclaration en Turbo Pascal

VAR

Nom-variable : BOOLEAN;

Exemple

Tableau de déclaration des objets

| Objet | Type / Nature | Rôle                                                        |
|-------|---------------|-------------------------------------------------------------|
| nulle | BOOLEEN       | Elle est à vraie si un entier est égal à 0 et à faux sinon. |

Déclaration en Turbo Pascal

VAR

Nulle : BOOLEAN;

Priorité des opérateurs

| Opérateurs                | Priorité  | Catégorie                 |
|---------------------------|-----------|---------------------------|
| - , NOT                   | Haute     | Opérateurs unaires        |
| *, / , DIV, MOD, AND      | Deuxième  | Opérateurs multiplicatifs |
| +, -, OR, XOR             | Troisième | Opérateurs additifs       |
| = , <> , < , ><br><= , >= | Quatrième | Opérateurs relationnels   |

Règles de priorité

- 1) Un opérande entre deux opérateurs de priorité différente est lié à l'opérateur de plus haute priorité.
- 2) Un opérande entre deux opérateurs de même priorité est lié à celui de gauche.
- 3) Les expressions entre parenthèses ont la priorité quel que soit le niveau interne.

III.4. Le type caractère

a) Domaine des valeurs du type caractère

Un caractère est représenté par le caractère lui-même placé entre deux apostrophes. On distingue plusieurs types de caractères :

- les lettres alphabétiques:
  - les lettres en majuscules ["A".."Z"]
  - les lettres en minuscules ["a".."z"]
- les chiffres ["0".."9"]
- les symboles comme la touche Echap (Escape) "!", "\$", ">", "ç", "é", ...
- les caractères non imprimables comme le retour chariot, la touche Echap (Esc), ...

Tous ces caractères sont ordonnés selon leurs codes ASCII variant de 0 à 255. (Voir tableau des codes ASCII à l'annexe 1).

Exemples

| Caractère             | Code ASCII |
|-----------------------|------------|
| "A"                   | 65         |
| "B"                   | 66         |
| "Z"                   | 90         |
| "a"                   | 97         |
| "b"                   | 98         |
| "2"                   | 50         |
| "3"                   | 51         |
| "?"                   | 63         |
| " " (caractère blanc) | 32         |

Remarques

D'après ces exemples, on peut constater que :

- les chiffres, les lettres en majuscule et les lettres en minuscule ont des codes ordonnés et contigus
- les lettres en majuscule précèdent les lettres en minuscule.
- les lettres en minuscule ne suivent pas les lettres en majuscule donc l'intervalle ["A".."z"] n'est pas formé uniquement de lettres.

b) La concaténation des caractères

On peut concaténer deux caractères ou plusieurs caractères en employant l'opérateur "+".

Exemples

| Caractère c1 | Caractère c2 | c1+c2 |
|--------------|--------------|-------|
| "a"          | "b"          | "ab"  |
| "1"          | "6"          | "16"  |
| "5"          | " "          | "5"   |

Remarque

Les valeurs obtenues dans la colonne c1+c2 ne sont pas du type caractère.

**Une variable de type caractère contient impérativement un seul caractère.**

c) Les opérateurs relationnels sur les caractères

Du moment que les valeurs du type caractère sont ordonnées grâce à leurs codes ASCII, on peut alors les comparer en utilisant les opérateurs relationnels déjà définis sur les entiers.

Exemple

En se basant sur le tableau des exemples de codes, on peut en déduire les affirmations suivantes :

- "A"<"B" est VRAI puisque 65<66
- "a" <"A" est FAUX puisque 97>65
- "?"<"B" est VRAI puisque 63<66

d) La déclaration d'une variable de type caractère

Tableau de déclaration des objets

| Objet        | Type / Nature | Rôle |
|--------------|---------------|------|
| Nom-variable | CARACTERE     | Rôle |

Déclaration en Turbo Pascal

VAR

Nom-variable : CHAR;

Activité 7

Présentez le tableau de déclaration des objets pour y déclarer des variables de type caractère en spécifiant leurs rôles. Traduisez ces déclarations en Turbo Pascal.

e) Les fonctions prédéfinies relatives au type caractère

On vous présente le tableau suivant récapitulant les fonctions standard relatives au type caractère.

**N.B.** La variable c désigne un caractère et la variable n désigne un entier compris entre 0 et 255.

| Fonction         | Rôle                                                     | Exemples                                                                                     |
|------------------|----------------------------------------------------------|----------------------------------------------------------------------------------------------|
| <b>ORD(c)</b>    | Renvoie le code ASCII du caractère c.                    | ORD("A") vaut 65<br>ORD("a") vaut 97                                                         |
| <b>CHR(n)</b>    | Renvoie le caractère dont le code ASCII est n.           | CHR(66) vaut "B"<br>CHR(98) vaut "b"                                                         |
| <b>SUCC(c)</b>   | Renvoie le caractère successeur de c.                    | SUCC("F") vaut "G"<br>SUCC("0") vaut "1"                                                     |
| <b>PRED(c)</b>   | Renvoie le caractère prédécesseur de c.                  | PRED("b") vaut "a"<br>PRED("0") vaut "/"                                                     |
| <b>UPCASE(c)</b> | Convertit le caractère c en majuscule si c'est possible. | UPCASE("a") vaut "A"<br>UPCASE("A") vaut "A"<br>UPCASE("5") vaut "5"<br>UPCASE("!") vaut "!" |

Activité 8

Cherchez les résultats des expressions suivantes :

| Expressions         |
|---------------------|
| CHR(ORD("A")+1)     |
| ORD(SUCC(CHR(255))) |
| ORD(PRED(CHR(0)))   |

### III.5. Le type chaîne de caractères

#### a) Définition

Une chaîne de caractères est une entité composée d'une suite de  $n$  caractères.  $n$  étant compris entre 0 et 255. Si  $n$  est nulle, on dit que la chaîne est vide.

**N.B.** Les valeurs chaîne de caractères sont définies entre guillemets dans la spécification et l'algorithme. Ces guillemets sont remplacés par des apostrophes.

#### Exemples :

"La programmation est passionnante!"  
 "20 mars 1956"  
 " " : c'est la chaîne vide

**Attention :** ""(chaîne vide)  $\neq$  " " (l'espace ou caractère blanc)

#### b) Les opérateurs relationnels sur les chaînes de caractères

La comparaison de deux ou de plusieurs chaînes de caractères est basée sur les codes ASCII. En effet, la comparaison se fait caractère par caractère en partant des premiers.

#### Exemple :

Attribuons à CH1 la valeur "FAMILLE" et à CH2 la valeur "FAMILY" et évaluons l'expression CH1>CH2.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| F | A | M | I | L | L | E |
| F | A | M | I | L | Y |   |

La machine comparera caractère par caractère. Jusqu'au 5ème caractère, il n'y a aucune différence. La 6ème comparaison sera concluante car savoir si CH1>CH2 revient à déterminer si "L" est supérieur à "Y" et puisque ce n'est pas le cas alors CH1>CH2 est FAUSSE.

#### c) L'accès à un caractère d'une chaîne

On peut accéder au  $i^{\text{ème}}$  caractère d'une chaîne CH en utilisant la notation CH[i] avec  $1 \leq i \leq \text{Long}(\text{ch})$ .

#### Exemples

On donne à CH la valeur "TABLE", on peut dire que :

- CH[2] vaut "A"
- CH[5] vaut "E"

**d) La déclaration d'une variable de type chaîne de caractères**

**Première formulation**

Tableau de déclaration des objets

| Objet        | Type / Nature | Rôle |
|--------------|---------------|------|
| Nom-variable | CHAINE        | Rôle |

Déclaration en Turbo Pascal

VAR

Nom-variable : STRING;

Dans ce cas, la chaîne peut atteindre 256 caractères.

**Deuxième formulation**

Tableau de déclaration des objets

| Objet        | Type / Nature      | Rôle |
|--------------|--------------------|------|
| Nom-variable | CHAINE[taille_max] | Rôle |

Déclaration en Turbo Pascal

VAR

Nom-variable : STRING[taille\_max];

Dans ce cas, la chaîne a une taille maximale égale à celle spécifiée dans la déclaration.

**Exemple**

Tableau de déclaration des objets

| Objet   | Type / Nature | Rôle               |
|---------|---------------|--------------------|
| Nom     | Chaîne        | Le nom d'un élève  |
| Adresse | Chaîne [48]   | Adresse de l'élève |

Déclaration en Turbo Pascal

VAR

Nom : STRING;

Adresse: STRING[48];

**e) Les fonctions et les procédures standard sur les chaînes**

Le type chaîne est doté d'un nombre important de fonctions et de procédures standard. Nous citons les plus utilisées.

• **Les fonctions sur les chaînes de caractères**

| Fonction chaîne                | Rôle                                                                                                   | Exemples                                                                               |
|--------------------------------|--------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| <b>LENGTH(CH)</b>              | Renvoie le nombre de caractères dans CH.                                                               | LENGTH('Bravo') vaut 5<br>LENGTH(' ') vaut 0                                           |
| <b>POS(CH1,CH2)</b>            | Renvoie la position de la 1ère occurrence de CH1 dans CH2. Si CH1 n'est pas dans CH2, elle retourne 0. | POS('t','tatouer') vaut 1<br>POS('T','tatouer') vaut 0<br>POS('cas','cartable') vaut 0 |
| <b>COPY(CH,p,n)</b>            | Renvoie une sous-chaîne de n caractères à partir de la position p de CH.                               | COPY('cartable',4,5) vaut 'table'                                                      |
| <b>CONCAT(CH1,CH2,...,CHn)</b> | Renvoie la concaténation de CH1, CH2, ..., CHn. C'est l'équivalent de l'opérateur "+".                 | CONCAT('20','/', 'mars','/', '1956') vaut '20/mars/1956'                               |

• Les procédures sur les chaînes de caractères

| Procédure chaîne         | Rôle                                                                                                                                                                                                                                 | Exemples                                                                                                                                                                                                                                      |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DELETE(CH,p,n)</b>    | Enlève n caractères de CH à partir de la position P.                                                                                                                                                                                 | On attribue à CH la valeur 'ESCLAVE'<br>Après DELETE(CH,1,3), CH devient 'LAVE'                                                                                                                                                               |
| <b>INSERT(CH1,CH2,p)</b> | Insère la chaîne CH1 dans la chaîne CH2 à partir de la position P.                                                                                                                                                                   | On attribue à CH2 la valeur 'CABLE'<br>Après INSERT('RTA',CH2,3), CH2 devient 'CARTABLE'                                                                                                                                                      |
| <b>STR(n,CH)</b>         | Génère à partir de n une chaîne CH.                                                                                                                                                                                                  | On attribue à n la valeur 15<br>Après STR(n,CH) à part le n =15, on obtient CH='15'                                                                                                                                                           |
| <b>VAL(CH,n,e)</b>       | Génère à partir d'une chaîne CH une entité numérique n (Entier ou réel). e est une variable de type entier, elle contiendra 0 si la conversion est possible sinon, elle contiendra la position du premier caractère causant l'échec. | On attribue à CH la valeur '152.6':<br><b>Cas 1</b> n est du type entier<br>Après VAL(CH,n,e), on obtient n qui vaut 0 et e qui vaut 4.<br><b>Cas 2</b> n est du type réel<br>Après VAL(CH,n,e), on obtient n qui vaut 152.6 et e qui vaut 0. |

### III.6. Le type tableau

a) Définition

Le tableau unidimensionnel ou vecteur est une structure de données permettant de ranger un nombre fini d'éléments de même type.

Un vecteur est caractérisé par :

- Un nom servant d'identificateur.
- Une taille.
- Le type des éléments qu'il contient.

Activité 9

Donnez un exemple pour chaque déclaration donnée.

| Déclaration algorithmique |                                        | Déclaration Turbo Pascal        |
|---------------------------|----------------------------------------|---------------------------------|
| Type/Nature               | Rôle                                   | VAR                             |
| Tableau de 4 Entiers      | Tableau contenant des rangs de mois.   | T1:Array [1..4] of integer ;    |
| Tableau de 3 réels        | Tableau de moyennes.                   | M: Array [0..2] of real;        |
| Tableau de 4 booléens     | Tableau de résultats d'une expérience. | T3 :Array [-2..1] of Boolean ;  |
| Tableau de 6 caractères   | Tableau de voyelles.                   | A :Array [1..6] of char ;       |
| Tableau de 3 chaînes      | Tableau de noms.                       | Name: Array[1..3] of String;    |
| Tableau de 4 entiers      | Tableau d'entiers pairs.               | B :Array['a'..'d'] of integer ; |

**Remarques**

- Les cases d'un vecteur sont indicées.
- Les indices doivent être de type scalaire.

**b) L'accès à un élément d'un vecteur**

L'accès à un élément du tableau est direct. Pour accéder au i<sup>ème</sup> élément d'un tableau, il suffit de donner l'identificateur du tableau et l'indice i qui doit varier entre la borne inférieure et la borne supérieure définies au niveau de la déclaration.

**Exemple**

Soit la déclaration suivante en Pascal :

```
VAR
 V : ARRAY[1..6] OF CHAR ;
```

Pour une telle déclaration, on peut prévoir le tableau suivant :

|   |     |    |     |     |     |     |
|---|-----|----|-----|-----|-----|-----|
| V | 'y' | '! | '5' | 'A' | 'l' | 'j' |
|   | 1   | 2  | 3   | 4   | 5   | 6   |

Ainsi, on aura les résultats suivants :

```
V[2] = '!'
V[6] = 'j'
```

**c) La déclaration d'un vecteur**

**Première formulation**

**Tableau de déclaration des objets**

| Objet        | Type / Nature                          | Rôle |
|--------------|----------------------------------------|------|
| Nom-variable | Tableau de taille_max de type_éléments | Rôle |

**Déclaration en Turbo Pascal**

```
VAR
 Nom-variable : ARRAY[Binf..Bsup]
 OF type_éléments;
```

**Deuxième formulation**

**Tableau de déclaration nouveaux types**

| Type                                              |
|---------------------------------------------------|
| Nom_type = Tableau de taille_max de type_éléments |

**Déclaration en Turbo Pascal**

```
TYPE
 Nom_type = ARRAY[Binf..Bsup] OF
 type_éléments;
VAR
 Nom-variable : Nom_type ;
```

**Tableau de déclaration des objets**

| Objet        | Rôle     |
|--------------|----------|
| Nom-variable | Nom_type |

#### IV - Retenons

- Une constante est un objet qui a une seule valeur tout le long d'un programme.
- Une variable est un objet qui peut prendre différentes valeurs tout le long d'un programme.
- Les objets utilisés dans un programme sont répertoriés en types qui définissent l'ensemble de valeurs qu'ils peuvent prendre.
- Chaque type admet un domaine de valeurs au delà desquelles les calculs deviennent erronés.
- Il existe des modules de calcul standards ou prédéfinis appelés procédures ou fonctions propres à chaque type.
- Une chaîne de caractères est une entité composée d'une suite de n caractères, n étant compris entre 0 et 255.
- Un tableau est une structure de données permettant de ranger un nombre fini d'éléments de même type et selon une disposition donnée.

## EXERCICES

### Exercice n° 1

Copiez le tableau suivant sur votre cahier puis remplissez les colonnes 2 et 3 par les déclarations en algorithmique et en Pascal.

|                                                                                                                           | Déclaration en algorithmique | Déclaration en Pascal |
|---------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------|
| Un entier <b>n</b>                                                                                                        |                              |                       |
| Deux entiers <b>m</b> et <b>n</b>                                                                                         |                              |                       |
| Un réel <b>r</b>                                                                                                          |                              |                       |
| Trois réels <b>a</b> , <b>b</b> , <b>c</b>                                                                                |                              |                       |
| Un caractère <b>c</b>                                                                                                     |                              |                       |
| Une chaîne de caractères <b>CH1</b>                                                                                       |                              |                       |
| Deux chaînes de caractères <b>CH1</b> et <b>CH2</b>                                                                       |                              |                       |
| Un vecteur <b>V</b> de <b>25</b> entiers                                                                                  |                              |                       |
| Un vecteur <b>R</b> de <b>15</b> réels                                                                                    |                              |                       |
| Un vecteur <b>C</b> de <b>100</b> chaînes de caractères                                                                   |                              |                       |
| Un entier <b>n</b> , un réel <b>x</b> et un caractère <b>c</b>                                                            |                              |                       |
| Deux entiers <b>m</b> et <b>n</b> , une chaîne de caractères <b>MOT</b> et un vecteur <b>VECT</b> de <b>10</b> caractères |                              |                       |

### Exercice n° 2

Copiez le tableau suivant sur votre cahier puis remplissez la deuxième colonne par le résultat de l'évaluation de l'expression donnée en Pascal.

| Expression                     | Evaluation |
|--------------------------------|------------|
| 15 MOD 2                       |            |
| 15 DIV 2                       |            |
| 15 / 2                         |            |
| Succ ("A")                     |            |
| Pred ("z")                     |            |
| Succ("a")                      |            |
| Pred ("y")                     |            |
| "A" > "b"                      |            |
| 150 DIV 5 MOD 5                |            |
| 150 MOD 5 DIV 5                |            |
| "Bac"+ "2008"                  |            |
| CONCAT("10","/05","/", "2006") |            |
| Long ("Lycée")                 |            |
| POS("Baccalauréat", "lauréat") |            |
| POS("Baccalauréat", "Lauréat") |            |
| Valeur ("10.5",d,e)            | d =<br>e = |
| "Ali" < "Alia"                 |            |

### Exercice n° 3

Evaluez les expressions suivantes :

a) Pour A=10 , B=15 et C=3

- 1/ A \* (B + C)
- 2/ A \* (B - C)
- 3/ A + (B \* C)
- 4/ A - (B \* C)
- 5/ A + (B - C)

b) Pour A=100 , B=5 et C=8

- 1/ A DIV (B MOD C)
- 2/ A MOD (C DIV B)
- 3/ A MOD (B MOD C)
- 4/ A DIV (C DIV B)

**Exercice n° 4**

Copiez le tableau suivant sur votre cahier puis remplissez le en mettant une croix (x) dans la ou les cases du type qu'on peut choisir pour chaque résultat.

| Expression          | Type   |      |           |        |
|---------------------|--------|------|-----------|--------|
|                     | Entier | Réel | Caractère | Chaîne |
| 10 MOD 5            |        |      |           |        |
| 10 MOD 5 / 3        |        |      |           |        |
| 6 + 2*5 DIV 3       |        |      |           |        |
| 1.2+1.3*1.4         |        |      |           |        |
| CARRE(ABS(-5))      |        |      |           |        |
| 6+3*5 DIV 2         |        |      |           |        |
| 4+RACINECARREE(16)  |        |      |           |        |
| "A"+"ALI"           |        |      |           |        |
| "A" + SUCC ("a")    |        |      |           |        |
| PRED ("b")          |        |      |           |        |
| SUCC("l")+PRED("l") |        |      |           |        |

## CHAPITRE 9

# Les structures simples

### Objectifs

- Résoudre des problèmes faisant appel aux structures simples
- Ecrire des programmes en Pascal utilisant les structures simples

### Plan du chapitre

- I. Introduction
- II. Les opérations d'entrée et de sortie
- III. L'affectation
- IV. Retenons

### Exercices

## CHAPITRE 9

# Les structures simples

" Dans ce chapitre, vous allez commencer à écrire vos premiers programmes en Pascal. Les problèmes à résoudre traiteront les structures simples à savoir les opérations d'entrées/sorties et l'affectation. "

" Rien ne sert de penser, il faut réfléchir avant "  
**Pierre Dac**

## I. Introduction

Dans le chapitre 7 et pour traduire la saisie d'un objet  $n$ , nous avons utilisé la notation : **Lire (n)**.

Cette notation algorithmique signifie que le programme, lors de son exécution, demande à l'utilisateur de saisir la valeur de l'objet qu'il mettra dans le contenant  $n$ . Retenez pour l'instant que  $n$  est une case mémoire ou un endroit permettant de garder la valeur de cet objet.

Au cours de l'analyse, on pourra se trouver contraint de définir une inconnue  $V$  qui a figuré dans une expression précédente. Quand on sait que  $V$  est une donnée que l'utilisateur devra fournir au programme, on traduira ceci par : **V = Donnée**.

Cette écriture a la même signification que : **Lire (V)**. C'est uniquement le contexte au cours de l'analyse qui permettra le choix entre la notation **Lire** et la notation **Donnée**.

Voyons maintenant la notation servant à afficher quelques choses à l'écran. Le verbe couramment utilisé est : **Ecrire**. Donc, pour afficher le contenu d'un objet  $m$ , il suffit d'utiliser la notation: **Ecrire(m)**.

## II. Les opérations d'entrée et de sortie

Pour assurer la communication entre l'utilisateur et la machine, il existe des instructions appelées instructions d'entrées et de sorties ou tout simplement opérations d'entrées/sorties.

### II.1. L'opération d'entrée

Dans un premier sens, on distingue l'instruction qui permet à l'utilisateur de rentrer ou de saisir des valeurs au clavier pour qu'elles soient utilisées par le programme. Cette opération s'appelle la lecture.

Tout simplement, pour que l'utilisateur entre (à l'aide du clavier) la nouvelle valeur d'un objet  $n$ , on appliquera la syntaxe suivante :

| Analyse             | Algorithmique   | Pascal              |
|---------------------|-----------------|---------------------|
| $n = \text{Donnée}$ | <b>Lire (n)</b> | <b>Readln (n) ;</b> |

**Remarques**

- 1 Dès que le programme rencontre une instruction de lecture, l'exécution s'arrête en attendant la saisie d'une valeur au clavier. Aussitôt que la touche Entrée a été frappée, l'exécution reprend après validation de la saisie.
- 2 Quand on demande à la machine de lire une valeur, cela implique que l'utilisateur va devoir écrire cette valeur.
- 3 Avant de lire le contenu d'un objet, il est très fortement conseillé d'écrire des **libellés** ou des messages à l'écran, afin de prévenir l'utilisateur de ce qu'il doit saisir.

**Exemple :**

N= Donnée ("Donner la valeur de")

**II.2. L'opération de sortie**

Dans un deuxième sens, on distingue l'instruction qui permet au programme de communiquer des valeurs à l'utilisateur en les affichant à l'écran. Cette opération s'appelle **écriture**.

Tout simplement, pour que la machine affiche la valeur d'un objet n, on appliquera la syntaxe suivante :

| Analyse                              | Algorithmique | Pascal                               |
|--------------------------------------|---------------|--------------------------------------|
| Ecrire ("La valeur de n est de ", n) |               | Write ('La valeur de n est de ',n) ; |

**Remarques**

- 1 Quand on demande à la machine d'écrire une valeur, c'est pour que l'utilisateur puisse la lire.
- 2 En Pascal, il est possible d'utiliser l'instruction WriteLn(n) qui permet d'afficher la valeur de l'objet n puis de revenir à la ligne.

- NB.**
- 1 - La lecture et l'écriture sont des termes qui doivent être compris du côté de la machine qui sera chargée de les exécuter.
  - 2 - La lecture et l'écriture sont des instructions algorithmiques qui ne présentent pas de difficultés particulières, une fois qu'on a bien assimilé ce problème du sens du dialogue (homme à machine, ou machine à homme).

**Activité 1**

Dans le chapitre 7 vous avez résolu des problèmes faisant appel aux opérations d'entrée et de sorties. L'activité 3 en est un exemple puisqu'elle permet d'afficher la moyenne trimestrielle d'un élève dans une matière donnée. Cette moyenne est calculée en appliquant la formule suivante **Moyenne = (NC +2\*NS)/3** sachant que NC est la note du devoir de contrôle et NS est celle du devoir de synthèse.

Analysez puis déduisez un algorithme solution à cette activité.

### Spécification du problème

**Résultat :** Ecrire ("La moyenne est de",  $(NC+2*NS)/3$  )

**Traitement :** Pour afficher la moyenne d'un élève, il faut avoir ses deux notes NC et NS qui sont deux données puis d'appliquer la formule de calcul.

*NC = Donnée ("Donner la note du devoir de contrôle ")*

*NS = Donnée ("Donner la note du devoir de synthèse ")*

**Algorithme :**

0) Début Algorithme Moyenne

*// cet algorithme permet d'afficher la moyenne trimestrielle d'un élève dans une matière donnée//*

1) Ecrire ("Donner la note du devoir de contrôle") ; Lire (NC)

2) Ecrire ("Donner la note du devoir de synthèse") ; Lire (NS)

3) Ecrire ("La moyenne est de",  $(NC+2*NS)/3$ )

4) FinMoyenne

**Codification des objets :**

| Nom | Type | Rôle                          |
|-----|------|-------------------------------|
| NC  | Réel | La note du devoir de contrôle |
| NS  | Réel | La note du devoir de synthèse |

### Activité 2

On se propose d'écrire un programme en Pascal qui permet d'afficher le périmètre d'un rectangle connaissant sa longueur et sa largeur.

### Spécification du problème

**Résultat :** Ecrire ("Le périmètre est de",  $(Lo+La)*2$ )

**Traitement :** L'affichage du résultat de l'expression  $(Lo+La)*2$  aura besoin de deux objets Lo et La qui sont deux données.

*Lo = Donnée ("Donner la longueur ")*

*La = Donnée ("Donner la largeur")*

**Algorithme :**

0) Début Algorithme Périmètre

*// cet algorithme permet d'afficher le périmètre d'un rectangle donné//*

1) Ecrire ("Donner la longueur") ; Lire(Lo)

2) Ecrire ("Donner la largeur") ; Lire(La)

3) Ecrire ("Le périmètre est de",  $(Lo + La)*2$ )

4) FinPérimètre

**Codification des objets :**

| Nom | Type | Rôle                        |
|-----|------|-----------------------------|
| Lo  | Réel | La dimension de la longueur |
| La  | Réel | La dimension de la largeur  |

### Activité 3

- 1/ Utilisez le programme de l'activité précédente pour afficher le périmètre d'un carré de côté c.
- 2/ Modifier quelques instructions de ce programme pour introduire une seule fois le côté d'un carré et afficher son périmètre.

### Activité 4

On se propose d'écrire un programme en Pascal qui permet d'afficher le carré et le cube d'un entier donné.

### Spécification du problème

**Résultat :** Ecrire ("n^2 = ", carré(n), "n^3 = ", n\*carré(n) )

**Traitement :** L'affichage de  $n^2$  et de  $n^3$  nécessite la valeur de l'objet n qui est une donnée.

*n = Donnée ("Donner la valeur de n")*

**Algorithme :**

0) Début Algorithme carré\_cube

*// cet algorithme permet d'afficher le carré et le cube d'un entier n donné//*

- 1) Ecrire ("Donner la valeur de n") ; Lire(n)
- 2) Ecrire ("n^2 =", carré(n), " n ^3 =", n\*carré(n))
- 3) FinCarré\_cube

**Codification des objets :**

| Nom | Type   | Rôle            |
|-----|--------|-----------------|
| n   | entier | L'objet utilisé |

### Application

Quels sont les résultats produits par l'algorithme " pgm " suivant ?

0) Début Algorithme carré\_cube

*// cet algorithme permet d'afficher le carré et le cube d'un entier n donné//*

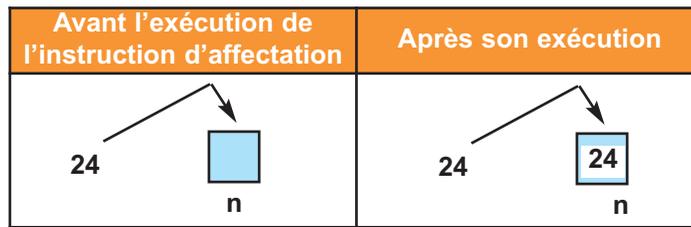
- 1) Ecrire ("Donner la valeur de n") ; Lire(n)
- 2) Ecrire ("n^2 =", carré(n), "n ^3 =", n\*carré(n))
- 3) FinCarré\_cube

## III. L'affectation

### III.1. Introduction

L'action d'affectation est le fait d'attribuer une valeur à un objet donné. L'instruction d'affectation se note avec le signe  $\leftarrow$  .

Ainsi, l'instruction  $n \leftarrow 24$  permet d'attribuer la valeur 24 à l'objet  $n$ .



### III.2. Syntaxe

La syntaxe de l'instruction d'affectation se présente comme suit :

| Analyse                | Algorithmique | Pascal           |
|------------------------|---------------|------------------|
| $n \leftarrow 4*m + 5$ |               | $n := 4*m + 5 ;$ |

#### Remarques

- Il est évident que l'objet  $n$  soit de type numérique. Si  $n$  a été défini dans un autre type, l'exécution de cette action d'affectation provoquera une erreur.
- Il est possible d'attribuer à un objet la valeur d'un autre objet telle quelle ou modifiée mais de type compatibles.

| Exemple                          | Commentaire                                                                                    |
|----------------------------------|------------------------------------------------------------------------------------------------|
| $m \leftarrow n$                 | La valeur de l'objet $m$ devient égale à la valeur de l'objet $n$ .                            |
| $x \leftarrow 2*y + 5$           | La valeur de l'objet $x$ devient égale au double de la valeur de l'objet $y$ incrémentée de 5. |
| $a \leftarrow \text{carré}(a)-3$ | La valeur de l'objet $a$ devient égale au carré de sa valeur actuelle décrétementée de 3.      |

- L'instruction d'affectation n'a modifié en rien les valeurs des objets  $n$ ,  $y$  et la variable  $a$  (finale) est modifiée de la remarque 2. En effet, une instruction d'affectation ne modifie que ce qui est situé à gauche de la flèche.

#### Activité 5

1/ Quelles seront les valeurs des variables  $A$  et  $B$  après exécution des affectations suivantes ?

$A \leftarrow 2$   
 $B \leftarrow A - 3$   
 $A \leftarrow 3$

2/ Quelles seront les valeurs des variables  $A$ ,  $B$  et  $C$  après exécution des affectations suivantes ?

$A \leftarrow 15$   
 $B \leftarrow -8$   
 $C \leftarrow A + B$   
 $A \leftarrow 6$   
 $C \leftarrow B - A$

3/ a) Quelles seront les valeurs des variables A et B après exécution des affectations suivantes ?

```
A ← 5
B ← 3
A ← B
B ← A
```

- b) Cette suite d'affectations permet-elle de permuter les contenus des objets A et B ?
- c) Si l'on inverse les deux dernières instructions, cela change-t-il quelque chose au résultat final ?
- d) Proposez une suite d'affectations qui permet de permuter les valeurs de deux objets quelconques A et B.

### Activité 6

Dans le chapitre 7 vous avez résolu des problèmes faisant appel aux opérations d'entrée et de sorties et à l'affectation. L'activité 2 en est un exemple puisqu'elle permet d'afficher la surface d'un disque de rayon donné. Cette surface est calculée en appliquant la formule suivante

**Surface** =  $\pi * r^2$  sachant que r est le rayon du disque.

Analysez puis déduisez un algorithme solution à cette activité.

### Spécification du problème

**Résultat :** Ecrire ("La surface est de", Surface )

**Traitement :** Pour afficher le contenu de l'objet **Surface**, on applique la formule donnée ci-dessus à savoir **Surface** =  $\pi * r^2$  qui est une affectation (**Surface** ←  $\pi * r^2$  ). Sachant que  $\pi$  est une constante, il suffit alors d'avoir la valeur de l'objet **r** qui est une donnée.

*r = Donnée ("Donner le rayon du disque")*

*On nommera pi une valeur approchée de  $\pi$  à  $10^{-2}$  près, soit 3.14*

**Algorithme :**

0) Début Algorithme Surface\_disque

*// cet algorithme permet de calculer et d'afficher la surface d'un disque connaissant son rayon//*

- 1) Ecrire ("Donner la valeur du rayon") ; Lire(r)
- 2) Surface ← pi\*carré(r)
- 3) Ecrire ("La surface du disque est de", surface)
- 4) FinSurface\_disque

**Codification des objets :**

| Nom     | Type      | Rôle                 |
|---------|-----------|----------------------|
| pi      | Constante | Pi=3.14              |
| r       | Réel      | Le rayon du disque   |
| Surface | Réel      | La surface du disque |

**Activité 7**

On dispose de deux chaînes de caractères CH1 et CH2. On se propose de réaliser la concaténation de CH1 et de CH2 dans une chaîne RESULTAT.  
Analysez puis déduisez un algorithme solution à cette activité.

**Spécification du problème**

**Résultat :** Ecrire ("La chaîne résultat est", RESULTAT )

**Traitement :** Pour former la chaîne RESULTAT, on a besoin d'additionner les deux chaînes CH1 et CH2 et de ranger le résultat dans la chaîne RESULTAT. Donc,  
RESULTAT ← CH1 + CH2  
Pour pouvoir calculer RESULTAT, il faut avoir les valeurs des chaînes CH1 et CH2 qui sont deux données.

CH1 = Donnée ("Donner la chaîne 1")

CH2 = Donnée ("Donner la chaîne 2")

**Algorithme :**

- 0) Début Algorithme somme\_chaines  
*// cet algorithme permet de faire la somme de deux chaînes puis de l'afficher //*
- 1) Ecrire ("Donner la chaîne 1") ; Lire(CH1)
- 2) Ecrire ("Donner la chaîne 2") ; Lire(CH2)
- 3) Résultat ← CH1 + CH2
- 4) Ecrire ("La chaîne résultat est", RESULTAT)
- 5) FinSomme\_chaines

**Codification des objets :**

| Nom      | Type                 | Rôle            |
|----------|----------------------|-----------------|
| CH1      | Chaîne de caractères | Chaîne n°1      |
| CH2      | Chaîne de caractères | Chaîne n°2      |
| RESULTAT | Chaîne de caractères | Chaîne résultat |

**Activité 7**

Résoudre le même problème en utilisant la fonction prédéfinie CONCAT.

#### IV - Retenons

- 1 - Les opérations d'entrée/sortie facilitent le dialogue entre l'utilisateur et la machine.
- 2 - L'opération d'entrée permet à l'utilisateur de saisir des valeurs au clavier pour qu'elles soient utilisées par le programme. Cette opération est **la lecture**.
- 3 - L'opération de sortie permet au programme de communiquer des valeurs à l'utilisateur en les affichant à l'écran. Cette opération est **l'écriture**.
- 4 - L'affectation est le fait d'attribuer une valeur à un objet donné.

## EXERCICES

### Exercice n° 1

Donnez la syntaxe de l'opération d'entrée (au niveau de l'analyse et de l'algorithme).

### Exercice n° 2

Donnez la syntaxe de l'opération de sortie (au niveau de l'analyse et de l'algorithme).

### Exercice n° 3

Donnez la syntaxe de l'action d'affectation (au niveau de l'analyse et de l'algorithme).

### Exercice n° 4

Pour chacune des propositions suivantes, mettez dans la case qui suit, la lettre V si elle est juste et la lettre F sinon.

a) Une opération d'entrée se traduit en Pascal par :

Write                   Readln                   Read

b) Une opération de sortie se traduit en Pascal par :

Readln                   :=                   Write

c) Une opération d'affectation se traduit en Pascal par :

Readln                   Writeln                   :=

d) Une opération d'entrée sert à :

Afficher des résultats  Introduire des données  Initialiser des objets

e) Une opération de sortie se traduit en Pascal par :

Afficher des résultats  Introduire des données  Initialiser des objets

### Exercice n° 5

Ecrivez une analyse, un algorithme et un programme en Pascal qui permet de calculer puis d'afficher la somme de deux réels.

### Exercice n° 6

Ecrivez une analyse, un algorithme et un programme en Pascal qui permet de convertir en heures, minutes et secondes un temps donné en secondes.

### Exercice n° 7

Ecrivez une analyse, un algorithme et un programme en Pascal qui permet de chercher puis d'afficher le montant du TVA d'un produit connaissant son prix hors taxes et le taux de la TVA.

### Exercice n° 8

Ecrivez une analyse, un algorithme et un programme en Pascal qui permet de déterminer puis d'afficher le quotient et le reste de la division euclidienne de deux entiers  $m$  et  $n$ .

### Exercice n° 9

Ecrivez une analyse, un algorithme et un programme en Pascal qui permet de convertir et d'afficher en octets, kilo octets, méga octets et giga octets un nombre donné en bits.

### Exercice n° 10

Ecrivez une analyse, un algorithme et un programme en Pascal qui permet de permuter les contenus de deux objets.

## CHAPITRE 10

# Les structures de contrôle conditionnelles

### Objectifs

- Résoudre des problèmes faisant appel aux structures de contrôle conditionnelles
- Ecrire des programmes en Pascal utilisant les structures de contrôle conditionnelles

### Plan du chapitre

- I. Introduction
- II. La structure conditionnelle simple
- III. La structure conditionnelle généralisée
- IV. La structure conditionnelle à choix
- V. Retenons

### Exercices

## CHAPITRE 10

# Les structures de contrôle conditionnelles

*Jusqu'à maintenant les instructions des programmes que vous avez écrit sont exécutés l'une après l'autre par le micro-ordinateur. Nous allons voir dans ce chapitre que l'ordre de l'exécution peut être contrôlé par le biais d'instructions appelées conditions.*

## I. Introduction

### I.1. Activité 1

- 1/ Dans chacune des phrases suivantes, soulignez la condition en rouge et l'action à faire en bleu.
  - Si je fais mes leçons, j'aurai de bonnes notes.
  - Si je fais beaucoup de TP, je réussirai l'examen pratique.
  - Si je fais cet exercice, j'aurai de fortes chances de comprendre les structures conditionnelles.
- 2/ Pouvez-vous donner la forme générale des conditions utilisées dans ces phrases ?
- 3/ Dans chacune des phrases suivantes, soulignez
  - la condition en rouge,
  - l'action à faire si la condition est satisfaite en vert
  - et l'action à faire si la condition n'est pas satisfaite en bleu.
  - Si je fais mes leçons, j'aurai de bonnes notes sinon j'aurai de mauvaises notes.
  - S'il fait beau, je vais me promener sinon je resterai chez moi.
- 4/ Pouvez-vous donner la forme générale des conditions utilisées dans ces phrases ?

### I.2. Présentation

Vous avez certainement remarqué que nous sommes face à des situations que nous pouvons représenter sous la forme générale suivante :

Si expression **alors** traitement1 **sinon** traitement2

avec :

- expression est une condition ou une variable de type booléen
- traitement1 et traitement2 sont deux séquences d'instructions.

Si l'expression est vraie, le traitement1 sera exécuté et le second traitement sera ignoré ; si l'expression est fausse, seule le second traitement sera effectué.

Le mot **sinon** indique où se termine la première séquence d'instructions et où commence la seconde.

## Qu'est ce qu'une **condition** ?

Une condition est une expression booléenne. Elle est soit vraie soit fausse :

Fréquemment, cette condition se présente sous forme de comparaison.

A priori, les valeurs peuvent être de n'importe quel type (numériques, caractères...). Mais si l'on veut que la comparaison ait un sens, il faut que les deux valeurs de la comparaison soient du même type !

Les opérateurs de comparaison sont :

- = égal à ...
- <> différent de ...
- < strictement inférieur à ...
- > strictement supérieur à ...
- =< inférieur ou égal à ...
- >= supérieur ou égal à ...

L'ensemble des trois éléments constituant la condition forme donc, si l'on veut, une affirmation ou un prédicat, qui est VRAI ou FAUX.

### Remarques

- 1 Dans certains cas, lorsque l'expression est fausse, aucune instruction ne doit être exécutée. La condition s'exprime alors plus simplement sous la forme :

**si** expression **alors** séquence d'instructions

- 2 Quelle que soit la séquence choisie et exécutée, les instructions qui suivent la structure conditionnelle seront exécutées.
- 3 Certains problèmes exigent parfois de formuler des conditions qui ne peuvent pas être exprimées sous la forme simple exposée ci-dessus. Dans ce cas, vous utiliserez des opérateurs logiques ; les plus utilisés sont : **ET OU** et **NON**. Les deux premiers portent sur deux opérandes et le dernier sur un seul. Il est évident que ces opérandes sont des propositions logiques et ne peuvent prendre que deux valeurs : **vrai** ou **faux**.
  - a) **op1 ET op2** n'a la valeur **vrai** que si les deux opérandes ont la valeur **vrai**, sinon l'expression a la valeur **faux**.
  - b) **op1 OU op2** n'a la valeur **faux** que si les deux opérandes ont la valeur **faux**, sinon l'expression a la valeur **vrai**.
  - c) **NON opérande** a la valeur **vrai** si l'opérande a la valeur **faux** et inversement.

- 4 Pour faire apparaître plus clairement la structure, on écrit les séquences d'instructions légèrement en retrait des mots-clefs (**si**, **alors**, **sinon**). On dit qu'on **indente** le texte de l'algorithme.

D'une façon générale, une structure conditionnelle peut être présentée sous trois formes :

- 1) La structure conditionnelle simple.
- 2) La structure conditionnelle généralisée.
- 3) La structure conditionnelle à choix.

Dans la suite de ce chapitre, vous allez découvrir le mode d'utilisation de chacune de ces structures.

## II. La structure conditionnelle simple

### II.1. Activité 2

Reprenez l'activité 1 du chapitre précédent.

1/ Que fait le programme de cette activité ?

2/ On se propose maintenant d'utiliser cette activité pour afficher, en plus de la moyenne annuelle de l'élève, la décision du conseil des classes. La décision "Admis" ou "Redouble" est prise en fonction de la moyenne.

#### Spécification du problème

**Résultat :** Ecrire (moyenne, décision)

**Traitement :** Le calcul et l'affichage de la moyenne annuelle se font de la même façon que celle de la moyenne trimestrielle. Nous allons nous intéresser à l'affichage de la décision. Pour réaliser cette tâche, nous allons utiliser une condition sur la valeur de la moyenne. En effet, si elle est supérieure ou égale à 10, la décision est "Admis", sinon la décision est "Redouble".

En notant MA, la moyenne annuelle, on obtient

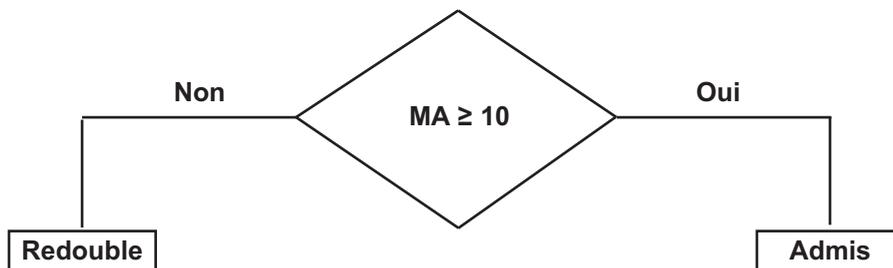
Si  $(MA \geq 10)$  alors  
 décision  $\leftarrow$  "Admis"

Sinon  
 décision  $\leftarrow$  "Redouble"

$MA \leftarrow (MT1 + 2*MT2 + 2*MT3) / 5$

MT1, MT2, MT3 = Données

Nous pouvons schématiser cette action comme suit :





### Activité 4

Allumez votre ordinateur et, avec l'aide de l'enseignant, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom "**DECISION**".

## II.4. Applications

### a) Application 1

Ecrivez un algorithme qui permet de vérifier la parité d'un entier donné.

### Spécification du problème

**Résultat :** Ecrire ( message )

**Traitement :** Pour afficher la parité d'un entier donné, on teste sa valeur.  
Si n est pair alors le message "pair" est affiché sinon le message " impair " est affiché.

Si  $n \bmod 2 = 0$       alors message  $\leftarrow$  "pair"  
                                 sinon mes  $\leftarrow$  "impair"

$N = \text{Donnée}$

### Algorithme :

0) Début Algorithme Parité

*// cet algorithme permet d'afficher la parité d'un entier n donné//*

1) Ecrire ("Donner un entier n") ; Lire(n)

2) Si  $(n2) = 0$  alors

    message  $\leftarrow$  "Pair"

    Sinon

        message  $\leftarrow$  "Impair"

    FinSi

3) Ecrire (n,"est", message)

4) FinParité

### Codification des objets :

| Nom     | Type                 | Rôle                                    |
|---------|----------------------|-----------------------------------------|
| n       | entier               | Une donnée                              |
| Message | Chaîne de caractères | Une chaîne contenant 'Pair' ou 'Impair' |

### Activité 5

Allumez votre ordinateur et, avec l'aide de l'enseignant, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom "**PARITE**".

### b) Application 2

Ecrivez un algorithme qui calcule et affiche la valeur absolue d'un réel donné.

### Spécification du problème

**Résultat :** Ecrire ("La valeur absolue de",x,"est",v<sub>a</sub>)

**Traitement :** Rappelons tout d'abord que la valeur absolue d'un nombre est définie comme suit :

$$|x| = x \text{ si } x \geq 0 \text{ et } |x| = -x \text{ si } x < 0$$

L'affichage de la valeur absolue de n se fait suivant le résultat d'une condition sur le signe de x.

Si  $x \geq 0$       alors     $v_a \leftarrow x$   
                       sinon     $v_a \leftarrow -x$

$x = \text{Donnée}$

### Algorithme :

- 0) Début Algorithme valeur\_absolue  
                                   // cet algorithme permet d'afficher la valeur absolue d'un réel x donné//
- 1) Ecrire ("Donner un réel x") ; Lire (x)
- 2) SI (x >= 0) alors v<sub>a</sub> ← x  
       SINON  
           v<sub>a</sub> ← -x  
       FinSi
- 3) Ecrire ("La valeur absolue de", x,"est",v<sub>a</sub>)
- 4) FinValeur\_absolue

### Codification des objets :

| Nom            | Type | Rôle        |
|----------------|------|-------------|
| x              | réel | Une donnée  |
| v <sub>a</sub> | réel | Le résultat |

### Activité 6

1/ Allumez votre ordinateur et, avec l'aide de l'enseignant, traduisez cet algorithme en Pascal testez-le et enregistrez le programme source sous le nom " **VALEUR\_ABSOLUE\_1** ".

2/ On rappelle qu'il existe une fonction prédéfinie appelée ABS(a) qui retourne la valeur absolue du réel a. Modifiez votre programme en utilisant cette fonction et enregistrez le programme source sous le nom " **VALEUR\_ABSOLUE\_2** ".

### c) Application 3

Ecrivez un algorithme qui permet de chercher puis d'afficher le plus grand de deux entiers donnés.

### Spécification du problème

**Résultat :** Ecrire ("Le plus grand de",a,"et",b,"est",Max)

**Traitement :** L'affichage du plus grand entier se fait suivant le résultat d'une comparaison des deux entiers.

```

Si a ≥ b alors
 max ← a
Sinon
 max ← b
FinSi
a, b = Données

```

**Algorithme :**

0) **DEBUT** Algorithme Max\_de\_2

*// cet algorithme permet d'afficher le plus grand de deux entiers a et b donnés//*

- 1) Ecrire ("Donner le premier entier") ; Lire(a)
- 2) Ecrire ("Donner le second entier") ; Lire(b)
- 3) Si (a >=b) alors
  - max ← a
  - Sinon
  - max ← b
  - FinSi
- 4) Ecrire ("Le plus grand de",a," et",b,"est", Max)
- 5) **FinMax\_de\_2**

**Codification des objets :**

| Nom | Type   | Rôle                                                      |
|-----|--------|-----------------------------------------------------------|
| a   | entier | Le premier entier                                         |
| b   | entier | Le second entier                                          |
| max | entier | La variable qui contient le plus grand des entiers a et b |

### Activité 7

Mettez votre micro-ordinateur sous tension et, avec l'aide de l'enseignant, traduisez cet algorithme en un programme Pascal, testez-le et enregistrez le programme source sous le nom "**Max\_2\_entiers**".

## III. La structure conditionnelle généralisée

### III.1. Activité 8

Reprenez l'activité 2 de ce chapitre.

1/ Que fait cette activité ?

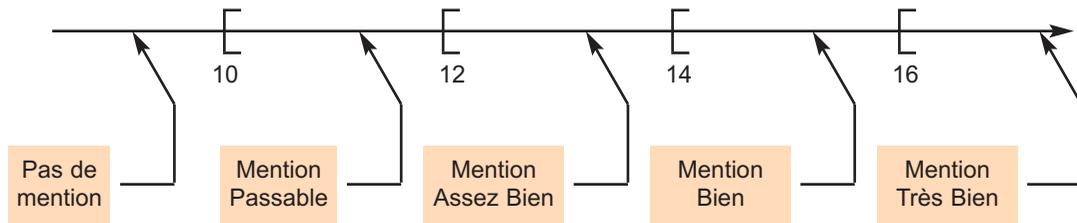
2/ On se propose maintenant d'afficher, en plus de la moyenne annuelle de l'élève et de la décision du conseil des classes, la mention dans le cas où l'élève est admis. Rappelons que pour déterminer la mention, on suit la démarche suivante :

*si (MA <12) alors la mention est " **Passable** " sinon si (MA <14) alors la mention est " **Assez bien** " sinon si (MA <16) alors la mention est " **Bien** " sinon la mention est " **Très bien** ".*

### Spécification du problème

**Résultat :** Ecrire ("Moyenne annuelle=",MA,"decision=",DEC,"Mention=",ME)

**Traitement :** L'affichage de la moyenne annuelle et de la décision étant déjà faits, nous allons nous intéresser à l'affichage de la **mention**. Pour réaliser cette tâche, nous allons utiliser une condition sur la valeur de la moyenne annuelle MA. En effet la représentation suivante va vous aider à déduire les intervalles de calcul de la mention.



### III.2. Qu'est ce qu'une condition généralisée ?

Nous avons vu dans la première partie de ce chapitre qu'une structure conditionnelle simple permet d'ouvrir deux situations ou deux états correspondant à deux traitements différents. Mais il y a des situations où deux états ne suffisent pas. Par exemple, la solution de l'activité 8 devant donner la mention d'un élève selon sa moyenne fait le choix entre quatre cas possibles : Passable, Assez bien, Bien ou Très bien.

Une première solution serait la suivante :

*// Cette partie de l'algorithme permet d'afficher la mention //*

- 1) Si  $(MA \geq 10)$  et  $(MA < 12)$  alors Mention  $\leftarrow$  "Passable" FinSi
- 2) Si  $(MA \geq 12)$  et  $(MA < 14)$  alors Mention  $\leftarrow$  "Assez bien" FinSi
- 3) Si  $(MA \geq 14)$  et  $(MA < 16)$  alors Mention  $\leftarrow$  "Bien" FinSi
- 4) Si  $(MA \geq 16)$  alors Mention  $\leftarrow$  "Très bien" FinSi

Vous constaterez que cette solution est un peu laborieuse. Les conditions se ressemblent plus ou moins, et surtout on oblige la machine à examiner 4 tests successifs alors qu'ils portent tous sur un même objet, la moyenne annuelle MA. Il serait intéressant de disposer les tests de la manière suivante:

*// Cette partie de l'algorithme permet d'afficher la mention en utilisant la structure conditionnelle généralisée//*

```

Si $(MA < 12)$ alors,
 Mention \leftarrow "Passable"
Sinon Si $MA < 14$ alors,
 Mention \leftarrow "Assez bien"
Sinon Si $MA < 16$ alors,
 Mention \leftarrow "Bien"
Sinon Mention \leftarrow "Très bien"
FinSi

```

Nous avons fait des économies de tests : au lieu de devoir taper quatre conditions, dont trois sont composées, nous n'avons plus que deux conditions simples. Cette économie sur les tests se répercute sur le temps d'exécution. Si la moyenne est inférieure à 12, la variable Mention reçoit la valeur "Passable" et l'exécution du programme passe directement à la fin, sans être ralentie par l'évaluation des autres cas qui suivent et qui sont forcément fausses.

Cette deuxième version est donc non seulement plus simple à écrire et à lire mais est également plus performante lors de l'exécution.

Voyons maintenant comment représenter une telle forme conditionnelle au niveau des structures algorithmiques puis dans le langage de programmation Pascal.

| Syntaxe en algorithmique                                              | Syntaxe en Pascal                                                                  |
|-----------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <b>Si</b> <i>condition1</i> <b>alors</b><br><i>Traitement1</i>        | <b>if</b> <i>condition1</i> <b>then</b><br><b>BEGIN</b><br><i>Traitement1</i>      |
| <b>Sinon Si</b> <i>condition2</i> <b>alors</b><br><i>Traitement2</i>  | <b>END</b>                                                                         |
| <b>Sinon Si</b> <i>condition3</i> <b>alors</b><br><i>Traitement3</i>  | <b>else if</b> <i>condition2</i> <b>then</b><br><b>BEGIN</b><br><i>Traitement2</i> |
| <b>Sinon Si</b> .....                                                 | <b>END</b>                                                                         |
| <b>Sinon Si</b> <i>condition n</i> <b>alors</b><br><i>Traitementn</i> | <b>ELSE IF</b> <i>condition</i> <b>THEN</b><br><b>BEGIN</b><br><i>Traitement n</i> |
| <b>Sinon</b> <i>Traitement (n + 1)</i>                                | <b>ELSE BEGIN</b> <i>Traitement (n + 1)</i>                                        |
| <b>Fin si</b>                                                         | <b>END ;</b>                                                                       |

### Activité 9

Utiliser la syntaxe de la structure conditionnelle généralisée pour rédiger un algorithme solution de l'activité 8.

#### Algorithme :

0) Début Algorithme Mention

*// Cet algorithme permet d'afficher la moyenne annuelle, la décision du conseil des classes et la mention attribuée à un élève//*

1) Ecrire ("Donner la moyenne annuelle") Lire (MA)

2) Si (MA < 10) alors DEC ← "Redouble"  
    ME ← " "

    Sinon

        DEC ← "Admis"

        Si (MA < 12) alors

            ME ← "Passable"

        Sinon Si (MA < 14) alors

            ME ← "Assez Bien"

        Sinon Si (MA < 16) alors

            ME ← "Bien"

    Sinon

        ME ← "Très Bien"

    FinSi

FinSi

3) Ecrire ("Moyenne annuelle = ", MA, " Décision =", DEC, "Mention =", ME )

4) FinMention

#### Codification des objets :

| Nom | Type                 | Rôle                                                  |
|-----|----------------------|-------------------------------------------------------|
| MA  | Réel                 | Une variable contenant la moyenne annuelle de l'élève |
| DEC | Chaîne de caractères | Une variable contenant la valeur de la décision       |
| ME  | Chaîne de caractères | Une variable contenant la valeur de la mention        |

### III.3. Applications

#### a) Application 1

Reprenez l'application 3 de la partie II de ce chapitre et cherchez puis affichez le plus grand (ou le maximum) de trois réels donnés.

#### Spécification du problème

**Résultat :** Ecrire (max)

**Traitement :** L'affichage du plus grand entier se fait suivant le résultat d'une comparaison des trois nombres.

- Commencez par comparer les deux premiers nombres
- A chaque fois, comparez le résultat avec le troisième réel.

```

Si (a > b) alors Max ← a
 Si c > Max alors Max ← c
 sinon Max ← b
 Si c > Max alors Max ← c
a, b = données

```

**Algorithme :**

- 0) Début Algorithme Maxde3  
*// cet algorithme permet d'afficher le plus grand de deux entiers a et b donnés//*
- 1) Ecrire ("Donner le premier nombre") Lire(a)
- 2) Ecrire ("Donner le deuxième nombre") Lire(b)
- 3) Ecrire ("Donner le troisième nombre") Lire(c)
- 4) Si a > b alors Max ← a
  - Si c > Max alors Max ← c
  - FinSi
  - Sinon Max ← b
  - Si c > Max alors Max ← c
  - FinSi
- FinSi
- 5) Ecrire ("Le plus grand de ",a, b, "et",c, "est", Max)
- 6) FinMaxde3

**Codification des objets :**

| Nom | Type | Rôle                                                       |
|-----|------|------------------------------------------------------------|
| a   | Réel | Le premier réel                                            |
| b   | Réel | Le deuxième réel                                           |
| c   | Réel | Le troisième réel                                          |
| Max | Réel | La variable qui contient le plus grand des réels a, b et c |

**Activité 10**

Cherchez une autre solution à ce problème et traduisez l'algorithme obtenu au Pascal et enregistrez le programme source sous le nom " *Max\_3\_entiers-S2* ".

**a) Application 1**

- 1/ On se propose de résoudre une équation de premier degré de la forme  $ax + b = 0$ .  $a$ ,  $b$  et  $x$  réels
  - a) Quelles sont les solutions possibles ?
  - b) Pour chaque condition, définir la (ou les) condition(s) de réalisation.
- 2/ Ecrire un algorithme puis un programme Pascal permettant de résoudre une équation au premier degré.

**Spécification du problème**

**Résultat :** Solution

**Traitement :** Trois solutions possibles existent dans la résolution d'une équation au premier degré de la forme  $ax + b = 0$  :

- 1- si  $a \neq 0$  alors la solution est  $-b/a$ .
- 2- si  $a=0$  et  $b=0$  alors la solution est  $\mathbb{R}$  (l'ensemble des réels).
- 3- si  $a=0$  et  $b \neq 0$  alors la solution est  $\{\}$  (l'ensemble vide).

Ce qui nous amène à la résolution suivante :

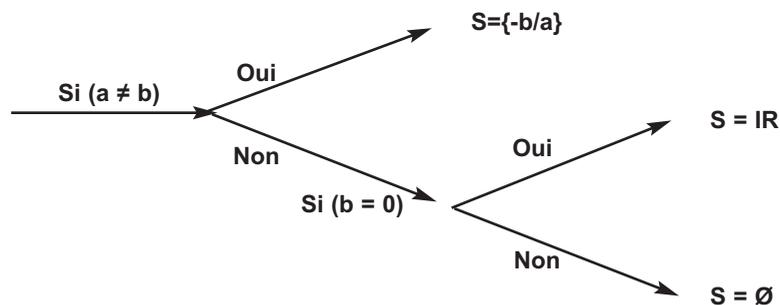
```

Si a ≠ 0 alors
 Ecrire (-b/a)
Sinon si b = 0 alors
 Ecrire ("IR")
Sinon
 Ecrire ("{}")
FinSi

```

$a, b = \text{données}$

Le traitement peut être schématisé comme suit :



**Algorithme :**

- 0) Début Algorithme Equation1  
*// Cet algorithme permet d'afficher la solution d'une équation de premier degré de la forme  $ax + b = 0$*
- 1) Ecrire ("a = ") ; Lire(a)
- 2) Ecrire ("b = ") ; Lire(b)
- 3) Si (a ≠ 0) alors  
 Ecrire (-b/a)  
 Sinon si b = 0 alors  
 Ecrire (" IR")  
 Sinon  
 Ecrire ("{}")  
 FinSi
- 4) FinEquation1

**Codification des objets :**

| Nom | Type | Rôle                    |
|-----|------|-------------------------|
| a   | réel | Le coefficient de x     |
| b   | réel | Le deuxième coefficient |

**Activité 12**

Mettez en marche votre ordinateur et traduisez cet algorithme en un programme Pascal, testez-le et enregistrez le programme source sous le nom "**Equation\_1**".

**b) Application 2**

Ecrire un algorithme puis un programme Pascal qui, à partir du numéro d'un mois, permet d'afficher la saison.

Les mois sont numérotés de 1 à 12 en commençant par janvier.

**Spécification du problème**

**Résultat :** Ecrire (Saison)

**Traitement :** Sachant qu'il existe quatre saisons qui sont le printemps, l'été, l'automne et l'hiver, on doit tester sur la valeur du numéro du mois pour afficher le nom de la saison correspondante.

Si (mois ∈ [3,5]) alors  
 Saison ← "Printemps"  
 Sinon si mois ∈ [6,8] alors  
 Saison ← "Eté"  
 Sinon si mois ∈ [9,11] alors  
 Saison ← "Automne"  
 Sinon Saison ← "Hiver"  
 mois = Donnée

**NB :** Nous remarquons que la condition de cette structure repose sur la relation d'appartenance.

**Algorithme :**

- 0) Début Algorithme Saison  
*// Cet algorithme permet d'afficher le nom de la saison qui correspond au numéro d'un mois donné //*
- 1) Ecrire ("Donner le numéro du mois") ; Lire (m)

```

2) Si (mois ∈ [3,5]) alors
 Saison ← "Printemps"
Sinon (si mois ∈ [6,8]) alors
 Saison ← "Eté"
Sinon (si mois ∈ [9,11]) alors
 Saison ← "Automne"
Sinon (Saison ← "Hiver")
FinSi
3) FinSaison

```

**Codification des objets :**

| Nom    | Type                 | Rôle                |
|--------|----------------------|---------------------|
| mois   | entier               | Le numéro du mois   |
| Saison | chaîne de caractères | Le nom de la saison |

### Activité 13

Allumez votre ordinateur et traduisez cet algorithme en programme Pascal, testez-le et enregistrez le programme source sous le nom "**Saison\_1**".

## IV. La structure conditionnelle à choix

### V.1. Activité 14

Reprenez l'application 3 de la partie II de ce chapitre.

1/ Que fait cette application ?

2/ Quel est le critère des conditions utilisées dans ce traitement ? Quel est son type ?

### Spécification du problème

**Résultat :** Ecrire (Saison)

**Traitement :** L'affichage du nom de la saison dépend de la valeur du numéro du mois qui est une variable de type entier. Au lieu d'utiliser la structure conditionnelle généralisée, il nous est possible de représenter notre solution en utilisant une nouvelle structure algorithmique : la structure conditionnelle à choix. En effet, pour choisir le nom de la saison, nous allons tester l'appartenance du numéro du mois à l'un des intervalles suivants :

|               |         |           |          |           |
|---------------|---------|-----------|----------|-----------|
| <b>Mois</b>   | 6, 7, 8 | 9, 10, 11 | 12, 1, 2 | 3, 4, 5   |
| <b>Saison</b> | Eté     | Automne   | Hiver    | Printemps |

## VI. Qu'est ce qu'une condition à choix ?

Supposons qu'on nous demande de choisir dans un menu, une des trois possibilités proposées. Le choix présenté n'est plus une alternative vu que le nombre de choix est supérieur à 2.

Nous nous trouvons en présence d'un choix multiple qui dépend d'un sélecteur (le choix) et qui peut s'écrire :

```

Selon choix faire
 1 : bloc1
 2 : bloc2
 3 : bloc3
Sinon écrire ("Mauvais choix")
FinSelon

```

Si le choix est 1, le premier bloc d'instructions (**bloc1**) est exécuté puis il y a saut aux instructions qui suivent le mot **fin selon**. Si le choix est différent de 1, 2 et 3, le message 'Mauvais choix' sera affiché et on passe directement à ce qui suit **fin selon**.

**Remarques**

- 1 Afin d'éviter toute ambiguïté, on exige que les différentes expressions logiques soient incompatibles deux à deux.
- 2 La traduction de cette structure en Pascal est restrictive : le sélecteur doit être de type scalaire (il doit avoir un successeur et un prédécesseur).

Voyons maintenant comment représenter une telle forme conditionnelle dans l'algorithme puis dans le langage de programmation Pascal.

| Syntaxe en algorithmique                                                                                                               | Syntaxe en Pascal                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Selon</b> <i>sélecteur</i> <b>Faire</b><br>Valeur1 : Bloc1<br>Valeur2 : Bloc2<br>.<br>.<br>.<br>sinon<br>Bloc n<br><b>Fin selon</b> | <b>Case</b> <i>sélecteur</i> <b>of</b><br>Valeur1 : Bloc d'instructions1 ;<br>Valeur2 : Bloc d'instructions2 ;<br>.<br>.<br>.<br>else<br>Bloc d'instructions n ;<br><b>end;</b> |

**Remarque**

En Pascal, dans le cas où un bloc comporte plusieurs instructions, il doit être délimité par **BEGIN** et **END**.

**Activité 15**

Utiliser la syntaxe de la structure conditionnelle à choix pour rédiger un algorithme solution de l'activité 14.

**Algorithme :**

- 0) Début Algorithme Saisons  
       // Cet algorithme permet d'afficher le nom de la saison qui correspond  
       au numéro d'un mois donné //
- 1) Ecrire("Donner le numéro du mois");Lire (mois)
- 2) Selon mois Faire
  - 3..5 : Saison ← "Printemps"
  - 6..8 : Saison ← "Eté"
  - 9..11 : Saison ← "Automne"
  - 12,1..2 : Saison ← "Hiver"
 Sinon

Saison ← "Mauvais choix"

Fin selon

3) Ecrire (Saison)

4) FinSaisons

**Codification des objets :**

| Nom    | Type   | Rôle                                       |
|--------|--------|--------------------------------------------|
| mois   | entier | Le numéro du mois                          |
| Saison | chaîne | Le nom de la saison ou le message d'erreur |

### Remarques

- 1 L'expression de la condition utilisée pour la saison de l'hiver aurait pu être exprimée comme suit :  $12,1,2 : \text{Saison} \leftarrow \text{'Hiver'}$
- 2 Si le numéro du mois saisi n'est pas valide ( $m \notin [1,12]$ ) alors la variable Saison contiendra un message d'erreur.

### Activité 16

Allumez votre ordinateur et traduisez cet algorithme en programme Pascal, testez-le et enregistrez le programme source sous le nom "**Saison\_2**".

## IV.3. Applications

### a) Application 1

Ecrire un algorithme puis un programme Pascal qui, à partir du numéro d'un mois, permet d'afficher le nombre de jours qui lui correspond.

### Spécification du problème

**Résultat :** Ecrire ("Le nombre de jours du mois", "est", nbj, "jours")

**Traitement :** Vous savez que le nombre de jours d'un mois donné autre que février est soit 31, soit 30. Pour le mois de février, le nombre de jours est 29 pour les années bissextiles sinon il est de 28.

Le choix de l'utilisateur de votre programme porte sur le numéro du mois qui est un entier donc de type scalaire. Par conséquent, il est possible d'utiliser la structure conditionnelle à choix.

On suppose que l'année est comprise entre 1950 et 2050

Selon mois faire

1,3,5,7,8,10,12 : nbj ← 31

4,6,9,11 : nbj ← 30

2 : Ecrire('Quelle année ?')

$a = \text{donnée}$

Si  $(a \text{ MOD } 4 = 0)$  alors

nbj ← 29

Sinon

nbj ← 28

FinSi

mois = Donnée

**Algorithme :**

- 0) Début Algorithme Nbre\_jours  
*// Cet algorithme permet d'afficher le nombre de jours d'un mois donné //*
- 1) Ecrire ("Donner le numéro du mois") ; Lire (mois)
- 2) Selon mois Faire
  - 1,3,5,7,8,10,12 : nbj ← 31
  - 4,6,9,11 : nbj ← 30
  - 2 : Ecrire ("Quelle année ?")  
 Lire(a)  
 Si (a MOD 4 = 0) alors  
     nbj ← 29  
 Sinon  
     nbj ← 28
- FinSelon
- 3) Ecrire ("Le nombre de jours du mois",mois," est de",nbj," jours")
- 4) FinNbre\_jours

**Codification des objets :**

| Nom  | Type   | Rôle                       |
|------|--------|----------------------------|
| mois | entier | Le numéro du mois          |
| a    | entier | L'année                    |
| nbj  | entier | Le nombre de jours du mois |

**Activité 17**

- 1/ Allumez votre ordinateur et traduisez cet algorithme en programme Pascal, testez-le et enregistrez le programme source sous le nom "**Nbj\_mois\_1**".
- 2/ Reprenez votre solution et traitez le cas où le numéro du mois est invalide ( $m \notin [1,12]$ ). Traduisez et enregistrez votre solution sous le nom "**Nbj\_mois\_2**".
- 3/ La condition utilisée pour déterminer si une année est bissextile est incomplète si on traite une année quelconque, avec l'aide de votre enseignant reprenez votre solution et traitez ce cas puis traduisez l'algorithme en Pascal et enregistrez le programme source sous le nom "**Nbj\_mois\_3**".

**b) Application 2**

Ecrire un algorithme puis un programme Pascal qui, à partir de la saisie de deux entiers et un opérateur (une opération) affiche le résultat après exécution de l'opération choisie.

**Exemple :**

Si les entiers sont 14 et 2 et l'opérateur est '+' alors le résultat affiché est 14+2=16.  
 Pour les mêmes entiers et si l'opérateur est '/' alors le résultat affiché est 14/2=7

**Spécification du problème**

**Résultat :** Ecrire (R)

**Traitement :** L'affichage du résultat dépend du choix de l'opérateur qui est un caractère ne pouvant prendre que l'une des quatre valeurs suivantes :

- “+” pour l'addition
- “-” pour la soustraction
- “\*” pour la multiplication
- “/” pour la division.

Le choix de l'utilisateur dans votre programme porte sur la valeur de l'opération qui est un caractère donc de type scalaire. Par conséquent, il est possible d'utiliser la structure conditionnelle à choix.

Selon op Faire

```

 "+" : R ← m + n
 "-" : R ← m - n
 "*" : R ← m * n
 "/" : R ← m / n

```

FinSelon

m, n, op = données

### Algorithme :

- 0) Début Algorithme calcul  
     *// Cet algorithme permet d'afficher le résultat d'une opération //*
- 1) Ecrire("Donner les deux entiers ");Lire(m,n)
- 2) Ecrire("Saisir l'opération +, -, \* ou /");Lire(op)
- 3) Selon op Faire
 

```

 "+" : R ← m + n
 "-" : R ← m - n
 "*" : R ← m * n
 "/" : R ← m / n

```
- FinSelon
- 4) Ecrire (m,op ,n, " = ",R)
- 5) FinCalcul

### Codification des objets :

| Nom | Type      | Rôle                       |
|-----|-----------|----------------------------|
| m   | entier    | Le premier entier          |
| n   | entier    | Le second entier           |
| op  | Caractère | Le code de l'opération     |
| R   | Réel      | Le résultat de l'opération |

### Activité 18

- 1/ Allumez votre ordinateur et, traduisez cet algorithme en programme Pascal, testez-le et enregistrez le programme source sous le nom "**Calcul\_1**".
- 2/ Reprenez votre solution et traitez le cas où le code de l'opération est invalide ( $op \notin \{ '+', '-', '*', '/' \}$ ). Traduisez et enregistrez votre solution sous le nom "**Calcul\_2**".
- 3/ Reprenez votre solution et traitez le cas où l'opération est '/' et le diviseur est nul. Traduisez et enregistrez votre solution sous le nom "**Calcul\_3**".

## V - Retenons

Les structures conditionnelles offrent la possibilité de résoudre des problèmes comportant divers choix. Il existe trois structures conditionnelles qui sont :

1 - la structure conditionnelle simple :

```

Si condition alors
 Traitement1
Sinon
 Traitement2
Fin si

```

2 - la structure conditionnelle généralisée :

```

Si condition 1 alors
 Traitement1
Sinon si condition 2 alors
 Traitement2
 :
 :
Sinon si condition n alors
 Traitement n
Sinon Traitement n+1
Fin si

```

3 - la structure conditionnelle à choix :

```

Selon sélecteur Faire
 Valeur 1 : Bloc1
 Valeur2 : Bloc2
 :
 :
Sinon
 Bloc n
Fin selon

```

## EXERCICES

Analysez puis écrivez un programme Pascal solution à chacun des exercices suivants

**Exercice n° 1**

Saisir un entier  $m$  puis afficher sa moitié s'il est pair et son double s'il est impair.

**Exemple :**

Si  $m = 5 \rightarrow 10$  l'affichage sera "le double de 5 est 10"

Si  $m = 10 \rightarrow 5$  l'affichage sera "la moitié de 10 est 5".

**Exercice n° 2**

Saisir deux entiers  $a$  et  $b$ , puis afficher si  $a$  est un diviseur de  $b$ .

**Exemple :**

Si  $a = 5$  et  $b = 30 \rightarrow 5$  est un diviseur de 30

Si  $a = 5$  et  $b = 27 \rightarrow 5$  n'est pas un diviseur de 27

**Exercice n° 3**

Saisir un entier  $a$  puis afficher s'il est pair ou impair.

**Exercice n° 4**

Saisir un entier entre 1 et 7 puis afficher en toutes lettres le nom du jour de la semaine qui lui correspond.

**Exercice n° 5**

Saisir un entier  $n$  et afficher sa racine carrée s'il est positif et son carré s'il est négatif.

**Exemple :**

Si  $n = -5 \rightarrow$  le carré de  $(-5)$  est 25

Si  $n = 16 \rightarrow$  la racine carrée de 16 est 4

**Exercice n° 6**

Résoudre dans  $\mathbb{R}$  une équation de second degré de la forme :  $ax^2 + bx + c = 0$  où  $a$ ,  $b$  et  $c$  sont aussi réels.

**Exercice n° 7**

Afficher dans l'ordre croissant quatre entiers saisis au clavier.

**Exercice n° 8**

Saisir une date sous la forme JJ MM AA puis chercher et afficher le nombre de jours du mois MM ainsi que le nombre de jours qui restent pour la fin de ce mois.

**Exemple :**

Si  $jj = 20,$   
 $MM = 3$  et  
 $AA = 2005$

$\rightarrow$  MM comporte 31 jours et il reste 11 jours pour la fin du mois.

### Exercice n° 9

Saisir un entier a compris entre 0 et 9 puis l'afficher en toutes lettres.

**Exemple :**

Si  $a = 5 \Rightarrow$  l'affichage sera "cinq"

Si  $a = 9 \Rightarrow$  l'affichage sera "neuf"

### Exercice n° 10

Pour un caractère alphabétique saisi au clavier, afficher un message indiquant si ce caractère est majuscule ou minuscule.

### Exercice n° 11

Traduire en toutes lettres un entier naturel n composé d'au maximum deux chiffres.

**Exemple :**

Si  $n = 45 \Rightarrow$  Quarante cinq

Si  $n = 61 \Rightarrow$  Soixante et un

## CHAPITRE 11

# Les structures de contrôle itératives

### Objectifs

- Résoudre des problèmes faisant appel aux structures de contrôle itératives
- Choisir la structure itérative adéquate pour résoudre un problème donné.

### Plan du chapitre

#### I. Introduction

#### II. La structure itérative complète : la boucle Pour ... Faire ...

#### III. La structure itérative à condition d'arrêt

1. La boucle **Répéter ... Jusqu'à ...**
2. La boucle **Tant Que ... Faire ...**

#### IV. Retenons

### Exercices

## CHAPITRE 11

# Les structures de contrôle itératives

*Jusqu'à maintenant les instructions des programmes que vous avez écrit sont exécutées au maximum une fois par le micro-ordinateur. L'exécution peut être répétée plusieurs fois et de différentes manières. Dans ce chapitre, vous allez apprendre comment exécuter plusieurs fois des instructions d'un programme.*

### I. Introduction

#### Activité 1

Le tableau ci-dessous présente des exemples de la vie courante, étudiez bien son contenu et proposez d'autres exemples.

| Exemples de la vie courante                                                                              | Comment traduire cet exemple ?                                                     | Remarques                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Un élève à une série de 6 exercices à faire.                                                             | POUR Exercice DE 1 A 6<br>FAIRE<br>Lire (énoncé)<br>Résoudre(problème)<br>FIN POUR | Le nombre des exercices est <b>connu à l'avance</b> .                                                                                                                                                                                             |
| Avant de faire cette série, l'élève doit lire et relire son cours jusqu'à ce qu'il assimile son contenu. | REPETER<br>Lire(cours)<br>JUSQU'A<br>Comprendre(cours)                             | <ul style="list-style-type: none"> <li>L'élève doit lire <b>au moins une fois</b> son cours.</li> <li>Le nombre de lectures est <b>inconnu à l'avance</b>.</li> <li>L'élève arrête de lire son cours lorsqu'il a assimilé son contenu.</li> </ul> |
| En classe, l'élève peut poser des questions à son professeur en cas de notions non claires.              | TANT QUE Notions non claires<br>FAIRE<br>Poser(question)<br>FIN TANT QUE           | <ul style="list-style-type: none"> <li>L'élève peut ne pas poser de question.</li> <li>Le nombre de questions est <b>inconnu à l'avance</b>.</li> </ul>                                                                                           |

D'après ces exemples, vous pouvez remarquer l'existence de deux types de structures itératives :

- 1) Les structures itératives complètes où le nombre d'itérations est connu à l'avance.
- 2) Les structures itératives à condition d'arrêt où le nombre d'itérations est inconnu à l'avance.

## II. La structure itérative complète

### Activité 2

Reprenez l'activité 9 du chapitre précédent et supposons qu'on se propose de saisir les moyennes annuelles des  $n$  élèves d'une classe et de les ranger dans un tableau  $M$ .

En utilisant les structures algorithmiques que nous avons vues jusqu'à maintenant, on peut écrire la séquence d'actions suivante :

```

Ecrire("Introduire la moyenne de l'élève 1")
Lire(M[1])
Ecrire("Introduire la moyenne de l'élève 2")
Lire(M[2])
Ecrire("Introduire la moyenne de l'élève 3")
Lire(M[3])
...
Ecrire("Introduire la moyenne de l'élève n")
Lire(M[n])

```

D'une façon générale si  $i$  est le numéro d'un élève donné, les deux lignes utilisées pour saisir sa moyenne et qui sont reprises pour chaque élève sont :

```

Ecrire("Introduire le contenu de la case", i)
Lire(M[i])

```

Ce qui nous amène à proposer le bloc d'instructions suivant :

```

Pour i de 1 à n Faire
 Ecrire ("Introduire le contenu de la case",i)
 Lire (M[i])
FinPour

```

Ainsi, en faisant varier le compteur  $i$  de 1 à  $n$ , toutes les cases seront remplies une à une.

### II.1. Syntaxe d'une structure itérative complète

| Au niveau de l'algorithme                                                                                                                                        | Au niveau du Pascal                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ----- } ----- }  Initialisation(s)                 </pre>                                                                                                  | <pre> ----- ; ----- ;                 </pre>                                                                                                                                 |
| <pre> <b>Pour</b> compteur de Vi à Vf <b>Faire</b>     Instruction 1     Instruction 2     ....     ....     Instruction n <b>FinPour</b>                 </pre> | <pre> FOR compteur := Vi TO Vf DO     Begin         Instruction 1;         Instruction 2;         ....;         ....;         Instruction n;     End;                 </pre> |

## II.2. Caractéristiques de la définition itérative complète

Un résultat a une définition itérative complète s'il est la répétition d'une suite d'instructions, un nombre fini de fois connu à l'avance.

### Activité 3

Soit à afficher le mot "Hello" 5 fois ; comment réaliser cette tâche ?

Pour i de 1 à 5 Faire  
Ecrire ("Hello")  
FinPour

**OU**

Pour i de 0 à 4 Faire  
Ecrire("Hello")  
FinPour

**OU**

Pour j de 'B' à 'F' Faire  
Ecrire("Hello")  
FinPour

i est un compteur de type entier  
Le nombre d'itérations est  $V_f - V_i + 1$

j est un compteur de type caractère  
Le nombre des itérations est  $ORD(V_f) - ORD(V_i) + 1$

### Activité 4

Proposez d'autres définitions répétitives afin de réaliser la même tâche.

### Remarques

- Le compteur doit être du type scalaire.
- La partie "Initialisations" concerne :
  - les éventuelles *-initialisations-* des variables qui seront mises à jour au niveau du traitement répétitif.
  - la définition des objets constants ne dépend pas du ième traitement.
- L'incréméntation du compteur est automatique avec un pas de 1 ou un pas de - 1.
- Le traitement répétitif de la boucle Pour peut s'exécuter 0 ou n fois ( $n \geq 0$ ). Il peut ne pas s'exécuter dans le cas où  $V_f < V_i$ .
- En Pascal, le module à répéter est délimité par **Begin** et **End**.

1) Ecrire un bloc d'instructions permettant de trouver le nombre de voyelles dans une chaîne de caractères ch.

```

nv ← 0
Pour i de 1 à n Faire
 Si (MAJUS(ch[i]) dans ["A","E","I","O","U","Y"]) Alors
 nv←nv+1
 FinSi
FinPour

```

2) Ecrire un bloc d'instructions permettant de trouver et d'afficher les nombres cubiques de 3 chiffres. Un nombre est dit cubique s'il est égal à la somme des cubes de ses chiffres (153 est cubique car  $153 = 1^3 + 5^3 + 3^3$ ).

```

Ecrire ("Les nombres cubiques de trois chiffres sont")
Pour i de 100 à 999 Faire
 c← i DIV 100
 d← i MOD 100 DIV 10
 u← i MOD 10
 Si i = c3+d3+u3 ALORS
 Ecrire (i)
 FinSi
FinPour

```

3) Ecrire un bloc d'instructions permettant d'afficher une chaîne de caractères ch à l'envers.

```

Pour i de LONG(ch) à 1 (pas=-1) Faire
 Ecrire(ch[i])
FinPour

```

4) Ecrire un bloc d'instructions permettant de calculer la factorielle d'un entier naturel n. On rappelle que  $n! = 1 * 2 * 3 * \dots * n-1 * n$  avec  $0! = 1$

```

f ← 1
Pour i de 2 à n Faire
 f ← f*i
FinPour

```

Dans le cas où  $n=0$ , on aura  $V_f < V_i$  et le traitement itératif ne sera pas exécuté. L'initialisation reste valable ce qui donne  $0! = 1$

## II.3. Applications

### 1) Application 1

Ecrire un algorithme puis un programme Pascal permettant de trouver et d'afficher le plus grand élément d'un vecteur T de n entiers (n est au maximum égal à 100).

**Spécification du problème**

**Résultat :** Ecrire (“La valeur la plus élevée est”,Max)

**Traitement :** La recherche de Max nécessite un parcours total du tableau, donc il est évident que sa définition est itérative complète. En voici les étapes à répéter.

1. On initialise l’objet Max avec la valeur du premier élément T[1] du tableau T.

2. Puis, on parcourt le tableau T de 2 à n.

– on compare chaque valeur du tableau avec la valeur de l’objet Max.

Si on trouve un élément qui lui est supérieur on le prend comme Max ; c-à-d on affecte Max de T [i]. On obtient la séquence suivante :

```

Max ← T[1]
Pour i de 2 à n Faire
 Si (T[i] > Max) alors Max ← T[i]
FinSi
FinPour

```

– Saisir la valeur de l’objet n et les valeurs des éléments du tableau T.

Le remplissage du tableau T se fait par des saisies successives de n éléments.

```

n = Donnée
T = Pour i de 1 à n Faire
 T[i] = Donnée
FinPour

```

**Algorithme :**

- 0) **DEBUT** Maximum
- 1) Ecrire(“n = ”); Lire(n)
- 2) Pour i de 1 à n faire
  - Ecrire(“Introduire l’entier”,i); Lire(T[i])
  - FinPour
- 3) Max ← T[1]
  - Pour i de 2 à n faire
  - Si (T[i] > max) Alors
  - Max ← T[i]
  - FinSi
  - FinPour
- 4) Ecrire(“La valeur la plus élevée est”,Max)
- 5) FinMaximum

**Codification des objets :**

| Nom | Type/Nature            | Rôle                                                    |
|-----|------------------------|---------------------------------------------------------|
| n   | Entier                 | Nombre d’entiers                                        |
| i   | Entier                 | Compteur                                                |
| Max | Entier                 | L’objet qui va contenir la valeur maximale du tableau T |
| T   | Tableau de 100 entiers | Le tableau des entiers                                  |

**Activité 5**

1/ Allumez votre micro-ordinateur et, traduisez cet algorithme en programme Pascal, testez-le et enregistrez le programme source sous le nom "*Maximum\_n*".

2/ Utilisez la solution précédente pour déterminer le minimum d'un tableau de n réels. Traduisez votre solution et enregistrez le programme source sous le nom "*Minimum\_n*".

**2) Application 2**

Ecrire un algorithme puis un programme Pascal permettant de trouver et d'afficher le nombre d'occurrences d'un réel R dans un vecteur T de n réels (n est au maximum égal à 50).

**Spécification du problème**

**Résultat :** Ecrire ("Le nombre d'occurrences de",R,"est",NOC)

**Traitement :** La recherche de NOC nécessite aussi un parcours total du tableau. Donc on doit utiliser une itération complète.

1. On initialise le nombre d'occurrences (NOC) à zéro
2. On parcourt le tableau T de 1 à n.
  - On teste pour chaque valeur si elle est égale à R.
  - Si c'est le cas, on incrémente NOC de 1.

On obtient la séquence suivante :

```

NOC ← 0
Pour c de 1 à n Faire
 Si (T[c] = R) Alors
 NOC ← NOC + 1
 FinSi
FinPour

```

– Saisir les valeurs des objets n et R puis les valeurs des éléments du tableau T. Le remplissage du tableau T se fait par les saisies successives des n éléments.

```

n, R = Donnée
T = Pour i de 1 à n Faire
 T[i] = Donnée
FinPour

```

**Algorithme :**

- 0) Début Occurrences
- 1) Ecrire ("R =");Lire(R)
- 2) Ecrire ("n = ");Lire(n)
- 3) Pour i de 1 à n Faire
  - Ecrire ("Introduire le réel ",i); Lire(T[i])
  - FinPour
- 4) NOC ← 0
  - Pour i de 1 à n Faire
    - Si (T[i]= R) Alors
    - NOC ← NOC + 1
    - FinSi
  - FinPour
- 5) Ecrire ("Le nombre d'occurrences de",R," est", NOC)
- 6) FinOccurrences

**Codification des objets :**

| Nom | Type/Nature          | Rôle                             |
|-----|----------------------|----------------------------------|
| R   | Réel                 | Réel objet de la recherche       |
| n   | Entier               | Nombre des entiers du tableau    |
| i   | Entier               | Compteur                         |
| NOC | Entier               | Nombre d'occurrences de R dans T |
| T   | Tableau de 100 réels | Le tableau des réels             |

**Activité 6**

Allumez votre micro-ordinateur et, avec l'aide de l'enseignant, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom "**Occurrences\_n**".

**3) Application 3**

On se propose d'écrire un algorithme puis un programme Pascal permettant de trouver la moyenne la plus élevée et le taux de réussite d'une classe de n élèves avec  $5 < n < 31$ .

**Exemple**

Soit n = 10

|   |       |      |       |       |       |       |      |       |       |      |
|---|-------|------|-------|-------|-------|-------|------|-------|-------|------|
| T | 10.23 | 9.57 | 17.56 | 12.40 | 13.45 | 11.76 | 8.10 | 15.80 | 14.34 | 9.98 |
|   | 1     | 2    | 3     | 4     | 5     | 6     | 7    | 8     | 9     | 10   |

La moyenne la plus élevée est 17.56

Le taux de réussite est de 70.00%

**Spécification du problème**

**Résultat :** Ecrire ("la moyenne la plus élevée est", M\_max)  
Ecrire ("Le taux de réussite est de", Taux, "%")

**Traitement :** – L'objet M\_max est déterminé par le biais d'un traitement équivalent à celui qui permet de chercher le maximum d'un tableau donné. Cette recherche nécessite une structure itérative complète :

- On initialise M\_max à la 1ère moyenne du tableau.
- On parcourt le tableau des moyennes de 2 à n.

Pour chaque moyenne, on teste si elle est supérieure à M\_max, si c'est le cas on lui affecte cette moyenne.

```

M_max ← M [1]
Pour i de 2 à n Faire
 Si (M[i] > M_max) Alors M_max ← M[i]
FinSi

```

– L'objet Taux est égal à  $(\text{Nbr\_admis} / n) * 100$  (Nbr\_admis étant l'objet utilisé pour avoir le nombre des élèves admis).

```

(Taux ← Nbr_admis / n) * 100

```

Pour avoir la valeur de l'objet Taux, on doit déterminer la valeur de l'objet Nbr\_admis.

- le nombre des élèves admis (Nbr\_admis) nécessite une boucle POUR :

- On initialise l'objet Nbr\_admis à 0.
- On parcourt le tableau des moyennes de 1 à n.  
 Si la moyenne est supérieure ou égale à 10 on incrémente Nbr\_admis de 1.

```

Nbr_admis ← 0
Pour i de 1 à n Faire
 Si (M[i] ≥ 10) alors Nbr_admis ← Nbr_admis + 1
FinSi

```

– Saisir la valeur de l'objet n et les valeurs des éléments du tableau M. Le remplissage du tableau M se fait par la saisie successive des n moyennes.

```

n = Donnée
M = Pour i de 1 à n Faire
 M[i] = Donnée
FinPour

```

**Algorithme :**

- 0) Début Classe
- 1) Ecrire ("Introduire le nombre des élèves"); Lire(n)
- 2) Pour i de 1 à n Faire
  - Ecrire ("Introduire la moyenne de l'élève",i); Lire(T[i])
 FinPour
- 3) M\_max\_OM [1]
  - Pour i de 2 à n Faire
    - Si (M[i]>M\_max) Alors
      - M\_max = M[i]
 FinSi
 FinPour
- 4) Nbr\_admis ← 0
  - Pour i de 1 à n Faire
    - Si (M[i] ≥ 10) Alors
      - Nbr\_admis ← Nbr\_admis + 1
 FinSi
 FinPour
- 5) Taux ← (Nbr\_admis/n)\*100
- 6) Ecrire("La moyenne la plus élevée est",M\_max)
  - Ecrire("Le taux de réussite est de",Taux,"%",)
- 7) FinClasse

| Nom       | Type/Nature         | Rôle                             |
|-----------|---------------------|----------------------------------|
| n         | Entier              | Nombre d'élèves                  |
| i         | Entier              | Compteur                         |
| Nbr_admis | Entier              | Nombre des élèves admis          |
| M_max     | Réel                | La valeur de la moyenne maximale |
| Taux      | Réel                | Le taux de réussite              |
| M         | Tableau de 30 réels | Le tableau des moyennes          |

**Activité 7**

- 1/ Allumez votre micro-ordinateur et, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom "**Moyenne\_classe\_1**".
- 2/ Pouvez vous déterminer les valeurs des deux objets M\_max et Taux dans une seule structure itérative ? Autrement dit, est-il possible de déterminer les valeurs de M\_max et de Taux en un seul parcours du tableau **M** ? Allumez votre micro-ordinateur et, avec l'aide de l'enseignant, traduisez l'algorithme solution en un programme Pascal et enregistrez le programme source sous le nom "**Moyenne\_classe\_2**".

**III. La structure itérative à condition d'arrêt**

Comme on l'a déjà signalé dans l'introduction, certains traitements nécessitent une définition itérative à condition d'arrêt car le nombre d'itérations est imprévisible. Bien que les deux derniers exemples de l'activité 1 nécessitent une définition itérative à condition d'arrêt, il existe une légère nuance entre eux :

- Le traitement itératif relatif à l'exemple 2 (lecture de la leçon) s'exécute au moins une fois.
- Le traitement itératif relatif à l'exemple 3 (poser une question au professeur) risque de ne pas avoir lieu (dans le cas où l'élève a bien compris la leçon).

C'est la principale différence entre la première formulation Répéter... Jusqu'à et la deuxième formulation TantQue ... Faire de la définition itérative.

**III.1. La structure Répéter ... Jusqu'à**

**1) Syntaxe**

| Au niveau de l'algorithme             | Au niveau du Pascal           |
|---------------------------------------|-------------------------------|
| Initialisation(s)                     | ___ ;<br>___ ;                |
| <b>Répéter</b>                        | Repeat                        |
| Instruction 1                         | Instruction 1;                |
| Instruction 2                         | Instruction 2;                |
| ....                                  | ....;                         |
| ....                                  | ....;                         |
| Instruction n                         | Instruction n;                |
| <b>Jusqu'à</b> (condition(s) d'arrêt) | Until (condition(s) d'arrêt); |

**2) Caractéristiques**

La structure Répéter...Jusqu'à est utilisée dans le cas où :

- on a une répétition d'une suite d'instructions, **un nombre de fois inconnu à l'avance** dont l'arrêt est géré par une condition.
- et
- on est sûr que le traitement itératif s'exécutera **au moins une fois**.

**Remarques**

- 1 La condition d'arrêt est considérée comme une condition de sortie car, une fois elle est vérifiée, on quitte la boucle sinon on itère encore jusqu'à ce qu'elle passe à l'état vrai.
- 2 La condition d'arrêt peut être simple ou composée. Elle est dite composée si plusieurs conditions sont à vérifier simultanément.
- 3 Les caractéristiques de la boucle Répéter...Jusqu'à en font une structure adaptée pour le contrôle de la saisie des données. On répète la saisie jusqu'à ce que les contraintes spécifiées dans l'énoncé du problème soient respectées.

**Exemples**

1) Reprenez l'activité 2 et ajoutez un contrôle de saisie sur les moyennes de telle façon qu'elles soient dans [ 0 , 20 ].

Pour contrôler la validité des moyennes saisies, il est possible d'utiliser la structure Répéter ... Jusqu'à... qui, pour pouvoir saisir une nouvelle moyenne, vérifie que la précédente satisfait la condition.

Bien entendu, vous ne pouvez pas utiliser la structure Pour car le nombre d'itérations est inconnu (l'utilisateur peut saisir la bonne moyenne dès la première saisie, comme il peut ne pas le faire).

```
Répéter
 Ecrire ("Introduire le contenu de la case", i)
 Lire (M[i])
Jusqu'à (M[i] ≥ 0) ET (M[i] ≤ 20)
```

2) Reprenez l'activité 2 de ce chapitre et supposez que le nombre d'élèves de la classe est compris entre 20 et 30. Au niveau de la résolution, cette contrainte n'est pas prise en compte, donc l'utilisateur peut se tromper et saisir une valeur négative ou toute autre valeur n'appartenant pas à l'intervalle spécifié.

Pour s'assurer que n est correctement saisi, vous pouvez insérer la séquence d'actions suivantes :

```
Répéter
 Ecrire("Introduire le nombre des élèves"),
 Lire(n)
Jusqu'à n dans [20..30]
```

3) Ecrivez un bloc d'instructions permettant de saisir une lettre de l'alphabet.

La réponse pourrait être :

```
Répéter
 Ecrire("donner une lettre alphabétique"),
 Lire(l)
Jusqu'à Majus(l) dans ["A".."Z"]
```

**NB.** Désormais, toute saisie de données doit respecter les contraintes logiques ou celles spécifiées dans l'énoncé.

### 3) Applications

#### a) Application 1

Ecrire un algorithme puis un programme Pascal permettant de saisir  $n$  entiers pairs puis d'afficher leur moyenne.  $n$  étant un entier de l'intervalle  $[15,30]$ .

#### Spécification du problème

**Résultat :** Ecrire("La moyenne est",Moy)

**Traitement :** – Remarquez tout d'abord que pour calculer la moyenne, il suffit de calculer la somme et de diviser cette somme par le nombre d'éléments  $n$ .

– Trouvez la moyenne de  $n$  entiers est un traitement répétitif qui nécessite une structure itérative complète.

- On initialise la moyenne (Moy) à zéro

- On répète  $n$  fois les actions suivantes

- saisir un entier

- vérifier sa parité, s'il est pair on continue sinon on refait la saisie

- ajouter l'entier saisi à la moyenne

– Divisez la valeur obtenue dans Moy par  $n$

On obtient alors la séquence suivante :

Moy  $\leftarrow$  0

Pour  $i$  de 1 à  $n$  faire

Répéter

$k = \text{donnée}$

Jusqu'à ( $k$  pair)

Moy  $\leftarrow$  Moy +  $k$

FinPour

Moy  $\leftarrow$  Moy /  $n$

– Saisir la valeur de  $n$  en vérifiant que  $n$  est dans l'intervalle  $[15,30]$ .

Répéter

$n = \text{Donnée}$

Jusqu'à  $n$  dans  $[15..30]$

#### Algorithme :

0) Début Moyenne\_pairs

1) Répéter

Ecrire("donner la valeur de ( $n / n$  dans  $[15,30]$ "));Lire( $n$ )

Jusqu'à  $n$  dans  $[15..30]$

2) Moy  $\leftarrow$  0

3) Pour  $i$  de 1 à  $n$  faire

Répéter

Ecrire("Saisir un entier pair");Lire( $k$ )

Jusqu'à ( $k \bmod 2 = 0$ )

Moy  $\leftarrow$  Moy +  $k$

FinPour

4) Moy  $\leftarrow$  Moy /  $n$

5) Ecrire("La moyenne est de",Moy)

6) FinMoyenne\_pairs

Codification des objets :

| Nom | Type/Nature | Rôle                               |
|-----|-------------|------------------------------------|
| Moy | Réel        | La moyenne des n entiers pairs     |
| n   | Entier      | Le nombre d'entiers pairs à saisir |
| i   | Entier      | Compteur de la boucle POUR         |
| k   | Entier      | L'entier à saisir                  |

### Activité 8

Allumez votre micro-ordinateur et, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom "**Moyenne\_pairs**".

**b) Application 2** (Examen pratique – Bac 2004)

Ecrire un algorithme puis un programme Pascal permettant de saisir n entiers ( $10 \leq n \leq 20$ ), comportant obligatoirement des valeurs positives et négatives, dans un tableau T.

Déterminer et afficher le nombre d'éléments positifs et la somme des valeurs négatives.

### Spécification du problème

**Résultat :** Ecrire("Le nombre d'éléments positifs est",NEP)  
Ecrire("La somme des valeurs négatives est",SVN)

**Traitement :** – Trouver le nombre d'éléments positifs (NEP) et la somme des valeurs négatives (SVN) est un traitement répétitif qui nécessite une structure itérative complète.

- On commence par initialiser NEP et SVN à zéro
- On répète n fois les actions suivantes
  - Lire un élément du tableau T
  - Si cet élément est positif, on incrémente NEP de 1, sinon on l'ajoute à SVN

```

NVP ← 0, SVN ← 0
Pour i de 1 à n faire
 Lire (T[i])
 Si T[i] < 0 Alors
 SVN ← SVN + T[i]
 Sinon NVP ← NVP + 1
 FinSi

```

FinPour

– Saisir la valeur de l'objet n en vérifiant que n est dans l'intervalle [10,20] puis les n éléments du tableau T. Le remplissage du tableau T se fait par les saisies successives des n éléments.

```

Répéter
 n = Donnée
Jusqu'à n dans [10,20]
T = Pour i de 1 à n Faire
 T[i] = Donnée
FinPour

```

**Algorithme :**

- 0) Début Positifs\_négatifs
- 1) Répéter
  - Ecrire ("donner la valeur de n / n dans [10,20]"); Lire(n)
  - Jusqu'à (n dans [10..20])
- 2) Pour i de 1 à n faire
  - Ecrire("Saisir un nombre")
  - Lire (T[i])
 FinPour
- 3) NVP ← 0, SVN ← 0
- 4) Pour i de 1 à n faire
  - Lire (T[i])
  - Si T[i] < 0 Alors
    - SVN ← SVN + T[i]
  - Sinon
    - NVP ← NVP + 1
 FinSi
 FinPour
- 5) Ecrire ("La somme des valeurs négatives est de",SVN)
- 6) Ecrire ("Le nombre des valeurs positives est de",NVP)
- 7) FinPositifs\_négatifs

**Codification des objets :**

| Nom | Type/Nature | Rôle                          |
|-----|-------------|-------------------------------|
| n   | Entier      | Le nombre de valeurs à saisir |
| i   | Entier      | Compteur de la boucle POUR    |
| T   | Tableau     | Tableau de n valeurs          |
| SVN | Entier      | Somme des valeurs négatives   |
| NVP | Entier      | Nombre de valeurs positives   |

**Activité 9**

1/ Allumez votre micro-ordinateur et, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom "**Positifs\_négatifs\_1**".

2/ Utilisez la solution précédente pour déterminer la somme des valeurs positives et celle des valeurs négatives. Traduisez votre solution et enregistrez le programme source sous le nom "**Positifs\_négatifs\_2**".

**c) Application 3** (Examen pratique – Bac 2003)

Ecrire un algorithme puis un programme Pascal permettant de chercher puis d'afficher la plus grande valeur d'un tableau T contenant n entiers ( $5 \leq n \leq 20$ ) ainsi que son indice. Dans le cas d'ex æquo, on affiche l'indice de la première occurrence.

**Spécification du problème**

**Résultat :** Ecrire ("La valeur maximale est",Max,"Son indice est",Indice)

**Traitement :** – La recherche de la valeur maximale et de son indice est un traitement répétitif qui nécessite une structure itérative complète.

- On commence par initialiser Indice à 1 et Max à T[1]
- On répète n-1 fois les actions suivantes
  - Lire un élément du tableau T
  - Si cet élément est strictement supérieur à Max, on remplace Max par cette valeur et Indice par l'indice de cette valeur.

```

indice ← 1, Max ← T[1]
Pour i de 2 à n faire
 Lire(T[i])
 Si (T[i] > Max) Alors Max ← T[i]
 Indice ← i
 FinSi
FinPour

```

– Saisir la valeur de l'objet n en vérifiant que n est dans l'intervalle [5,20] puis les n éléments du tableau T. Le remplissage du tableau T se fait par la saisie successive des n entiers.

```

Répéter
 n = Donnée
Jusqu'à n dans [5,20]
T = Pour i de 1 à n Faire
 T[i] = Donnée
FinPour

```

**Algorithme :**

- 0) Début Max\_Indexe
- 1) Répéter
  - Ecrire("donner la valeur de n / n dans [5,20]"),
  - Lire(n)
  - Jusqu'à (n dans [5..20])
- 2) Pour i de 1 à n faire
  - Ecrire("Saisir un entier")
  - Lire (T[i])
 FinPour
- 3) Indice ← 1, Max ← T[1]
- 4) Pour i de 2 à n faire
  - Lire(T[i])
  - Si (T[i] > Max) Alors
    - Max ← T[i]
    - Indice ← i
 FinSi
 FinPour
- 5) Ecrire("La valeur maximale est de",Max,"son indice est de",Indice)
- 6) FinMax\_Indexe

**Codification des objets :**

| Nom    | Type/Nature | Rôle                               |
|--------|-------------|------------------------------------|
| n      | Entier      | Le nombre d'entiers à saisir       |
| T      | Tableau     | Tableau où sont rangés les entiers |
| i      | Entier      | Compteur de la boucle Pour         |
| Max    | Entier      | La valeur maximale                 |
| Indice | Entier      | L'indice de la valeur maximale     |

**Activité 10**

- 1/ Allumez votre micro-ordinateur et, avec l'aide de l'enseignant, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom "**max\_indice\_1**".
- 2/ Utilisez la solution précédente pour déterminer et afficher la valeur maximale et la valeur minimale ainsi que leur indice. Traduisez votre solution et enregistrez le programme source sous le nom "**max\_indice\_2**".

**III.2. La boucle Tant Que...Faire**

**1) Syntaxe**

| Au niveau de l'algorithme                                                                                                                        | Au niveau du Pascal                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ----- } ----- }  Initialisation(s)                     </pre>                                                                              | <pre> ----- ; ----- ;                     </pre>                                                                                                       |
| <p><b>Tant Que condition(s) Faire</b></p> <pre> Instruction 1 Instruction 2 .... .... Instruction n <b>FinTantQue</b>                     </pre> | <p><b>While condition(s) do</b></p> <pre> <b>Begin</b> Instruction 1; Instruction 2; ....; ....; Instruction n; <b>End;</b>                     </pre> |

**2) Caractéristiques**

- La structure Tant Que...Faire est utilisée si le traitement :
  - a. est la répétition d'une suite d'instructions, un nombre de fois inconnu à l'avance,
  - b. peut ne pas avoir lieu.
- La condition mentionnée dans la notation est considérée comme une condition d'entrée car, une fois elle n'est plus vérifiée, on quitte la boucle sinon on itère encore jusqu'à sa non vérification (sa valeur est égale à faux).

**Remarque**

Avant d'entrer dans la boucle Tant Que, vous devez vous assurer que la condition a été initialisée. Si l'initialisation n'a pas été faite, le traitement itératif risque de ne pas être exécuté.

**3) Applications**

**a) Application 1**

Il s'agit de programmer la commande "Rechercher et remplacer" du menu Edition d'un logiciel de traitement de texte, qui, en fournissant le *texte*, un *mot1* et un *mot2*, permet de remplacer toute occurrence de *mot1* par *mot2* dans *texte*.

### Exemple

Soit le texte suivant :

" Les structures conditionnelles sont simples à comprendre. L'emploi des structures conditionnelles rend la programmation plus intéressante. "

Soit *mot1* = "conditionnelles"

Soit *mot2* = "itératives"

Le programme doit afficher :

" Les structures itératives sont simples à comprendre. L'emploi des structures itératives rend la programmation plus intéressante. "

### Spécification du problème

**Résultat :** Ecrire (Texte)

**Traitement :** – Chercher et remplacer les occurrences de l'ancien mot (*mot1*) par le nouveau mot (*mot2*) nécessite une définition itérative à condition d'arrêt de la forme **Tant Que** car l'ancien mot risque de ne pas se trouver dans *le texte* :

- On cherche la position de *mot1* dans le texte.
- On supprime l'occurrence trouvée.
- On insère dans la même position *mot2*.

Le traitement itératif s'arrête lorsque *mot1* ne se trouve plus dans le texte.

```
Tant Que POS(old, texte) ≠ 0 Faire
 p ← Pos (old, texte)
 Efface (texte, p, Long(old))
 Insère (new, texte, p)
```

FinTantQue

– Saisir *texte*, *old* et *new*.

*Texte* = Donnée

*old*, *new* = Données

### Algorithme :

- 0) Début Cherche\_remplace
- 1) Ecrire ("Introduire le texte initial"); Lire (texte)
- 2) Ecrire ("Introduire le mot à remplacer"); Lire (old)
- 3) Ecrire ("Introduire le nouveau mot") ; Lire (new)
- 4) Tant Que POS(old, texte) ≠ 0 Faire
  - p ← POS (old, texte)
  - Efface(texte,p, LONG(old))
  - Insère (new, texte, p)
- FinTantQue
- 5) Ecrire ('Le tesxte devient ', texte)
- 6) FinCherche\_remplace

**Codification des objets :**

| Nom   | Type/Nature | Rôle                                   |
|-------|-------------|----------------------------------------|
| Texte | Chaîne      | Le texte à saisir et le texte résultat |
| old   | Chaîne      | Le mot à remplacer                     |
| new   | chaîne      | Le mot de remplacement                 |
| p     | Entier      | La position dans Texte                 |

**Activité 11**

Allumez votre micro-ordinateur et, traduisez cet algorithme en un programme Pascal et enregistrez le programme source sous le nom "**Rech\_Remplace**".

**b) Application 2**

On se propose d'écrire un algorithme et un programme Pascal permettant d'afficher la somme d'une liste de nombres saisis. La liste se termine par -1 qui ne fera pas partie de la somme.

**Exemples**

- 1) Si la liste est formée par les entiers 4, 5, 8, -1  
Le résultat affiché est 17
- 2) Si la liste est formée uniquement par -1  
Le résultat affiché est 0

**Spécification du problème**

**Résultat :** Ecrire (S)

**Traitement :** – Le calcul de la valeur de l'objet S fait appel à une définition itérative à condition d'arrêt de la forme TantQue car ce calcul peut ne pas se faire dans le cas où la liste est réduite à un seul élément (-1) :

- On ajoute x à S.
- On saisit un nouvel élément x.

Le traitement itératif s'arrête lorsque la valeur de l'objet x est de -1.

Tant que  $x \neq -1$  Faire

$S \leftarrow S + x$

$x = \text{Donnée}$

FinTantQue

– Initialiser l'objet S à 0 et saisir la première valeur de l'objet x.

$S \leftarrow 0$

$x = \text{Donnée}$

**Algorithme :**

- 0) Début Somme
- 1)  $S \leftarrow 0$
- 2) Ecrire ("Introduire la valeur de x"), Lire(x)
- 3) TantQue  $x \neq -1$  Faire
  - $S \leftarrow S + x$
  - Ecrire ("Introduire la valeur de x"); Lire(x)
- FinTantQue

- 4) Ecrire("La somme est de", S)
- 6) FinSomme

**Codification des objets :**

| Objets | Type/Nature | Rôle               |
|--------|-------------|--------------------|
| S      | Réel        | La somme           |
| x      | Réel        | Le nombre à saisir |

**Activité 12**

- 1/ Allumez votre micro-ordinateur et, avec l'aide de l'enseignant, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom "**Somme\_1**".
- 2/ a) Quelle est la structure itérative à utiliser si l'initialisation de la variable S devient  $S \leftarrow 1$  ?  
 b) Traduisez votre solution et enregistrez le programme source sous le nom "**Somme\_2**".

**c) Application 3**

Le PGCD veut dire le plus grand commun diviseur. Le calcul du PGCD de deux entiers peut être effectué de plusieurs manières. La méthode de la différence permet de faire ce calcul en appliquant la démarche suivante :

$$\text{PGCD}(m, n) = \text{PGCD}(m-n, n) \text{ si } m > n \text{ et on pose } m = m - n$$

ou

$$\text{PGCD}(m, n) = \text{PGCD}(m, n-m) \text{ si } n > m \text{ et on pose } n = n - m$$

Le calcul s'arrête quand m devient égal à n

**Exemples**

- 1)  $\text{PGCD}(24, 10) = \text{PGCD}(14, 10)$   
 $= \text{PGCD}(4, 10)$   
 $= \text{PGCD}(4, 6)$   
 $= \text{PGCD}(4, 2)$   
 $= \text{PGCD}(2, 2)$   
 $= 2$
- 2)  $\text{PGCD}(184, 184) = 184 \rightarrow$  aucun calcul ne sera fait.

**Spécification du problème**

**Résultat :** Ecrire ("le PGCD est de",m)

**Traitement :** – Chercher le PGCD de deux entiers en utilisant la méthode proposée est un traitement itératif qui peut ne pas être exécuté dans le cas où les deux entiers saisis sont égaux. Par conséquent la structure itérative à utiliser est celle à condition d'arrêt **TantQue** :

Si  $(m > n)$  alors remplacez  $m$  par  $m-n$  sinon remplacer  $n$  par  $n-m$ .  
Le traitement itératif s'arrête lorsque  $m$  devient égal à  $n$ .

```
TantQue $m \neq n$ Faire
 Si $(m > n)$ alors
 $m \leftarrow m - n$
 Sinon
 $n \leftarrow n - m$
 FinSi
FinTantQue
```

– Saisir les valeurs des objets  $m$  et  $n$  en contrôlant leurs valeurs.

```
Répéter
 $m = \text{donnée}$
 $n = \text{donnée}$
Jusqu'à $(m > 0)$ et $(n > 0)$
```

**Algorithme :**

- 0) Début PGCD
- 1) Répéter
  - Ecrire ("Introduire un entier strictement positif "); Lire( $m$ )
  - Ecrire ("Introduire un entier strictement positif"); Lire( $n$ )
  - Jusqu'à  $(m > 0)$  et  $(n > 0)$
- 2) TantQue  $m \neq n$  Faire
  - Si  $(m > n)$  Alors
    - $m \leftarrow m-n$
  - Sinon
    - $n \leftarrow n-m$
  - FinSi
  - FinTantQue
- 3) Ecrire("Le PGCD est de",  $m$ )
- 4) FinPGCD

**Codification des objets :**

| Objets | Type/Nature | Rôle              |
|--------|-------------|-------------------|
| $m$    | Entier      | Le premier entier |
| $n$    | Entier      | Le second entier  |

**Activité 13**

Allumez votre micro-ordinateur et, avec l'aide de l'enseignant, traduisez cet algorithme en Pascal et enregistrez le programme source sous le nom **"PGCD"**.

#### IV - Retenons

Les structures répétitives permettent la résolution de problèmes comportant des traitements itératifs. Il existe trois structures répétitives qui sont :

1 - la structure itérative complète **Pour ... Faire ...** :

**Pour** compteur de  $V_i$  à  $V_f$  **Faire**

Traitement

**FinPour**

2 - la structure itérative à condition d'arrêt **Répéter ... jusqu'à ...** :

**Répéter**

Traitement

**Jusqu'à** *condition*

3 - la structure itérative à condition d'arrêt **Tant que ... Faire ...** :

**Tant Que** *condition* **Faire**

Traitement

**FinTant Que**

## EXERCICES

**Analysez puis écrivez un algorithme et le traduire en un programme Pascal solution à chacun des exercices suivants. Testez chaque programme obtenu.**

### Exercice n° 1

Saisir une chaîne de caractères puis afficher le nombre de ses lettres, le nombre de ses chiffres et le nombre de ses symboles.

### Exercice n° 2

Saisir une chaîne de caractères puis en extraire 3 chaînes :

- La chaîne de lettres L
- la chaîne de chiffres C
- la chaîne de symboles S

Afficher les chaînes obtenues.

### Exercice n° 3

Saisir une chaîne de caractères puis trouver et afficher la première voyelle rencontrée.

1. Version 1 : La chaîne contient sûrement une voyelle
2. Version 2 : La chaîne peut ne pas contenir de voyelle

### Exercice n° 4

Saisir un réel a et un entier b puis calculer et afficher le résultat de l'opération "a à la puissance b".  
Discuter tous les cas.

### Exercice n° 5

Saisir un entier naturel différent de 1 puis afficher s'il est premier ou non.

### Exercice n° 6

Chercher tous les nombres premiers se trouvant dans un intervalle [a,b].

### Exercice n° 7

Soit l'algorithme suivant :

- 0- Début premier
- 1- Ecrire ("Introduire un entier naturel strictement supérieur à 1")  
Lire (n)
- 2-  $d \leftarrow 0$   
Pour i de 1 à n Faire  
    Si  $n \text{ MOD } i = 0$  Alors  
         $d \leftarrow d + 1$   
    FinSi  
FinPour
- 3- Si  $d = 2$   
Alors  
    Ecrire ("Le nombre",n,"est premier")  
Sinon  
    Ecrire ("Le nombre",n,"n'est pas premier")  
FinSi
- 4- FinPremier

- 1/ Traduire cet algorithme en Pascal puis l'enregistrer sous le nom **VERSION1**.
- 2/ Ajouter le contrôle de saisie au niveau de la séquence 1.
- 3/ Ecrire, à la fin du programme et sous forme de commentaire { }, le rôle de la variable d
- 4/ La boucle POUR itère n fois. Transformer la séquence 2 afin que la boucle POUR itère le moins possible tout en gardant son rôle initial.
- 5/ Enregistrer le fichier sous le nom **VERSION2**. Afin d'optimiser la solution, remplacer la boucle POUR par la boucle REPETER . Que devient la séquence 3 ?

### Exercice n° 8

Le nombre 1089 passe à l'un de ses multiples lorsqu'il est écrit à l'envers ( 9801 = 9 \* 1089 ) ; un autre nombre de 4 chiffres possède la même propriété.

Chercher le nombre de 4 chiffres non palindrome autre que 1089 et qui passe à l'un de ses multiples lorsqu'il est écrit à l'envers.

### Exercice n° 9

Le jeu " 1 2 3 BOUM ! " consiste à demander à plusieurs joueurs de se succéder en comptant jusqu'à un nombre maximum (Max) décidé à l'avance. Tous les nombres divisibles par un nombre n ou qui contiennent la séquence de chiffres qui forment le nombre n sont à écarter. A leur place les joueurs doivent dire " BOUM ".

Lire un entier naturel non nul n et un nombre Max (Max est le nombre maximal jusqu'au quel on comptera).

**Exemple :** Soit n = 4 et Max = 20 ; les joueurs doivent citer ceci :

1 2 3 BOUM 5 6 7 BOUM 9 10 11 BOUM 13 BOUM 15 BOUM 17 18 19 BOUM

### Exercice n° 10

Il vous est demandé de saisir une chaîne de chiffres CH, de chercher la combinaison maximale CMAX et la combinaison minimale CMIN qu'on peut obtenir à partir des chiffres de CH, et enfin de les afficher.

Pour déduire CMAX à partir de CH, on vous propose les étapes suivantes :

- 1/ Chercher le plus grand chiffre dans CH.
- 2/ Le placer dans la chaîne CMAX.
- 3/ Remplacer le chiffre qui était le plus grand par le caractère '\*'.

**Exemple :** C1='51687412'

La combinaison maximale est : '87654211'

La combinaison minimale est : '11245678'

### Exercice n° 11

Chaque caractère possède un code ASCII ; pour que ce code soit " compris " par le micro-ordinateur, il est converti en code binaire (base 2 ) sous forme de 0 et de 1.

Ecrire un programme Pascal permettant de remplir un tableau T de 8 entiers égaux à 0 ou à 1 (ce qui va représenter le code binaire) puis retrouver l'équivalent de ce code en base 10 (c'est à dire trouver le code ASCII) et enfin afficher le caractère correspondant.

Code binaire (en 0 et 1) → Code ASCII (base 10) → Caractère correspondant

$$\text{Code ASCII} = \sum_{i=0} T[i] * 2^{7-i}$$

**Exemple :**

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| T | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|

$$\text{Code ASCII} = 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 69$$

Le caractère ayant comme code ASCII 69 est E

### Exercice n° 12

Tout nombre positif de deux chiffres  $ab$ , tel que  $a \neq b$ , possède une liste appelée " liste vers 9 ". Le principe est le suivant : on calcule la différence entre  $ab$  et son symétrique  $ba$ ; le résultat trouvé subit le même traitement ; on répète ce processus jusqu'à obtenir une différence =9. L'ensemble constitué par le nombre initial et les résultats des différences est appelé " liste vers 9 ".

**Exemple :**

Soit  $x = 18$  ;  $81 - 18 = 63$  ;  $63 - 36 = 27$  ;  $72 - 27 = 45$  ;  $54 - 45 = 9 \rightarrow$  Fin du traitement

La liste vers 9 est la suivante : **18 63 27 45 9**

Il vous est demandé d'introduire un nombre positif composé de deux chiffres obligatoirement différents, de générer sa " **liste vers 9** " et enfin de l'afficher.

## CHAPITRE 12

# Les sous programmes

### Objectifs

- Décomposer un problème en modules
- Présenter les solutions sous forme de procédures et de fonctions
- Ecrire des algorithmes et des programmes solutions

### Plan du chapitre

- I. Introduction
- II. Les fonctions
- III. Les procédures
- IV. Retenons

### Exercices

## CHAPITRE 12

## Les sous programmes

" Diviser les difficultés en autant de parcelles qu'il se peut afin de mieux les résoudre "  
DECARTES

*Dans ce chapitre, vous allez apprendre comment décomposer un problème en sous problèmes de taille réduite.*

## I. Introduction

Pour résoudre des problèmes complexes, il est souvent préférable de les décomposer en sous problèmes. Ces derniers sont à leur tour décomposés selon le besoin. La décomposition s'arrête aux sous problèmes relativement simples à résoudre. Par la suite, on associe à chaque sous problème un module assurant sa résolution. Un module peut être une fonction ou une procédure. L'algorithme résolvant le problème initial présentera un ensemble d'appels aux modules envisagés.

Une telle approche s'appelle analyse modulaire. Elle repose sur le vieil adage "Diviser pour régner".

Le recours à cette méthode de résolution présente des avantages indéniables tels que:

- Présenter des solutions claires en évitant la redondance des codes dans un programme.
- Simplifier la résolution du problème initial en supposant que les différents modules prévus sont déjà résolus.
- Se concentrer sur la résolution d'un sous problème à la fois.
- Détecter facilement les parties à consulter ou à modifier.
- Réutiliser des modules en les définissant dans une bibliothèque.

## II. Les fonctions

## Activité 1

Citez des **fonctions standard** que vous avez déjà utilisées dans vos programmes. N'oubliez pas de citer aussi **les paramètres** nécessaires au fonctionnement de chacune d'elles.

Par exemple, on peut citer la fonction **racine carrée** ( $x$ ) ou **SQRT**( $x$ ) :  $x$  est un paramètre de la fonction SQR, il peut être du type Entier, comme il peut être du type Réel.

## Activité 2

Soit à afficher la racine carrée d'un réel positif  $x$  donné. Ceci peut être réalisé de deux manières :

- |                                                                                                                                                                                                |                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• <math>y \leftarrow</math> Racine Carrée(<math>x</math>)</li> <li>• ECRIRE(<math>y</math>)</li> <li>• ECRIRE(Racine Carrée(<math>x</math>))</li> </ul> | <ul style="list-style-type: none"> <li>• <math>Y :=</math>SQRT(<math>x</math>) ;</li> <li>• Write(<math>y</math>) ;</li> <li>• Write(SQRT(<math>x</math>)) ;</li> </ul> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**N. B. :** L'appel d'une fonction renvoie toujours une valeur.

### Activité 3

Soit à saisir un entier naturel  $n$  (avec  $0 \leq n \leq 50$ ). On se propose de calculer et d'afficher la somme des factorielles des chiffres de  $n$ .

#### Exemples

Soit  $n = 30$   
 $\rightarrow 3! + 0! = 7$

Soit  $n = 5$   
 $\rightarrow 5! = 120$

Respectez le format d'affichage.

### Spécification du problème

**Résultat :** Afficher la somme des factorielles des chiffres du nombre saisi.

**Traitement :** Il s'agit de distinguer entre deux cas :

- Le cas où  $n$  est composé d'un seul chiffre ( $n$  dans  $[0 .. 9]$ ) : il s'agira alors de calculer puis d'afficher la factorielle de  $n$ .
- Le cas où  $n$  est composé de deux chiffres : il s'agira alors de calculer la factorielle du chiffre des dizaines, faire de même pour le chiffre des unités, calculer leur somme et l'afficher.
- Le calcul de la factorielle d'un nombre est un traitement répétitif qui nécessite une structure itérative complète :
  - On initialise la factorielle à 1
  - La nouvelle valeur de la factorielle est le produit de l'ancienne valeur de la factorielle par un compteur variant de 2 au nombre concerné par le calcul.

**Algorithme :**

0) Début Algorithme Som\_fact

1) Répéter

Ecrire("Introduire un entier compris entre 0 et 50"); Lire(n)

Jusqu'à n Dans [0..50]

2) Si (n Dans [0..9]) Alors

$fn \leftarrow 1$

Pour  $i$  de 1 à n Faire

$fn \leftarrow fn * i$

FinPour

Ecrire(n, ' ! = ', fn)

Sinon

$d \leftarrow n \text{ DIV } 10$

$fd \leftarrow 1$

Pour  $i$  de 1 à d Faire

$fd \leftarrow fd * i$

FinPour

```

u ← n MOD 10
fu ← 1
Pour i de 1 à u Faire
 fu ← fu*i
FinPour
Ecrire(d, " !+ ",u," != ",fd+fu)
FinSi

```

3) FinSom\_fact

Au niveau de cette solution algorithmique, on remarque la redondance de la partie calcul de la factorielle : celle où on calcule la factorielle de n, puis celle du chiffre des dizaines d et enfin celle du chiffre des unités u.

On peut éviter une telle redondance en définissant une fonction permettant de calculer la factorielle de n'importe quel entier naturel. Cette fonction sera **appelée** en cas de besoin.

En supposant que cette fonction est nommée **Factorielle**, l'algorithme **Som\_fact** devient :

0) Début Algorithme Som\_fact

1) Répéter

```

Ecrire("Introduire un entier compris entre 0 et 50") ; Lire(n)
Jusqu'à n Dans [0..50]

```

2) Si (n Dans [0..9]) Alors

```

Ecrire(n, " != ", Factorielle(n))

```

Sinon

```

d ← n DIV 10

```

```

u ← n MOD 10

```

```

Ecrire(d, " !+ ",u," != ", FactorielleE(d)+ Factorielle(u))

```

FinSi

3) FinSom\_fact

En utilisant la fonction Factorielle, l'algorithme Som\_fact devient plus concis et plus simple.

Il nous reste maintenant à définir la fonction factorielle.

### Définition de la fonction Factorielle

0) Fonction Factorielle(e :Entier) :Entier

1) f ← 1

```

Pour i de 2 à e Faire

```

```

 f ← f*i

```

```

FinPour

```

2) Factorielle ← f

3) FinFactorielle

La fonction factorielle est définie pour n'importe quel entier naturel e. C'est un **paramètre formel** de la fonction.

La fonction factorielle est appelée trois fois, au niveau du **programme principal** Som\_fact :

une première fois, en **substituant** e par n,

une deuxième fois, en **substituant** e par d, et

une troisième fois, en **substituant** e par u.

n, d et u sont appelés paramètres effectifs car ils constituent les paramètres manipulés réellement et pour lesquels nous voulons calculer les factorielles.

**NB.** Les paramètres effectifs et les paramètres formels doivent correspondre en nombre, ordre et type.

L'appel d'une fonction se fait via son nom suivi des paramètres effectifs. Ces derniers vont se substituer aux paramètres formels pour donner le résultat de la fonction.

## II.1. Syntaxes

### Syntaxe en algorithmique

```

0- Fonction nom_fonction(liste des paramètres formels) : Type du résultat
de la fonction
1-
2-
.
.
n-1- nom_fonction←Résultat
n- FIN nom_fonction

```

### Syntaxe en Pascal

```

Function nom_fonction(liste des paramètres formels) : Type du résultat de
la fonction ;
 {Partie déclarative des éventuels objets locaux }
Begin
 ;
 ;
 .
 .
 .
 nom_fonction:=Résultat;
End;

```

## II.2. Caractéristiques

- Une fonction est un sous-programme qui retourne une valeur d'un type identique à celui de la fonction. Une fonction a donc un type : c'est un type simple.

**Fonction nom\_fonction (liste des paramètres formels) : type du résultat**

- Puisque le type du résultat est celui de la fonction, on peut comprendre l'importance de la dernière action :

**Nom\_fonction ← Résultat trouvé**

En oubliant cette action, la fonction n'assurera pas sa tâche lors de l'appel.

- Tout nouvel objet utilisé dans une fonction est appelé **objet local**. Dans l'activité 3, i et f sont des objets locaux. Les objets locaux ne sont pas vus en dehors de la fonction où ils sont définis.

- Tout objet déclaré dans le programme principal est appelé **objet global**. C'est le cas des variables n, d et u au niveau de l'activité 3. De même, la fonction factorielle est considérée comme un objet global.
- La définition d'une fonction est constituée de trois parties:
  - La partie en-tête comportant le nom de la fonction suivi, entre parenthèses, de la liste des paramètres formels, puis du type du résultat qu'elle renvoie.
  - La partie déclarative des éventuels objets locaux.
  - La partie formant le corps de la fonction.
- L'appel d'une fonction provoque l'activation de celle-ci; il peut être effectué au niveau du programme principal ou au niveau d'un module appelant. L'appel se fait en mentionnant le nom de la fonction suivi de la liste des paramètres effectifs. Par exemple, en se référant toujours à l'activité 3, les paramètres n, d et u sont dits paramètres effectifs et ils se substitueront, à chaque appel, au paramètre formel e lors de l'exécution de la fonction.
- Pour que l'appel d'une fonction soit correct, il faut respecter les contraintes suivantes :
  - Une fonction retourne un résultat de type simple contenu dans son identificateur ; de ce fait, une fonction appelée retourne une valeur qu'on peut affecter à une variable, faire figurer dans une expression arithmétique ou logique ou afficher directement.
  - Le nom de la fonction appelée doit correspondre exactement au nom utilisé lors de sa définition.
  - Les paramètres effectifs et les paramètres formels doivent se correspondre en nombre, type et ordre.

### Remarque

Retenons qu'une fonction est essentiellement un module de calcul. Eviter d'y insérer les actions d'entrée et de sortie.

### Activité 4

Allumez votre micro-ordinateur et traduisez les algorithmes précédents en Pascal et enregistrez le programme source sous le nom "**Som\_fact**".

Faites attention à l'emplacement des différentes parties de votre programme.

## II.3. Applications

### 1) Application 1

Il s'agit de remplir le tableau suivant, où pour chaque description d'une tâche, on note si elle est réalisable ou non par une fonction. Dans le cas affirmatif, on présente l'entête de la fonction et dans le cas contraire, on explique pourquoi une fonction ne pourrait pas être choisie.

| Description d'une tâche à résoudre                                                 | Réalisable/Non réalisable par une fonction | Entête/Explication                                                                                                                                                                                             |
|------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Soit à calculer la moyenne arithmétique de n moyennes contenues dans un tableau T. | Réalisable                                 | Function<br>Moyenne(n:Integer;T:TAB):Real;                                                                                                                                                                     |
| Soit à calculer le produit de deux entiers.                                        | Réalisable                                 | Function Produit(a,b:Integer):Integer;                                                                                                                                                                         |
| Soit à convertir une durée donnée en secondes en heures, minutes et secondes.      | Non réalisable                             | Le module doit retourner trois résultats ce qui est contradictoire avec la définition d'une fonction. Elle retourne un seul résultat.                                                                          |
| Soit à calculer l'occurrence d'un caractère dans une chaîne.                       | Réalisable                                 | Function<br>Occurrence(c:Char;ch:String):Integer;                                                                                                                                                              |
| Soit à supprimer un entier occupant la position p d'un tableau T de n entiers.     | Non réalisable                             | Le tableau a subi une modification. Si on va le considérer comme résultat, la fonction ne pourra pas assurer cette tâche puisque son résultat doit être du type simple et ce n'est pas le cas du type Tableau. |
| Soit à calculer un réel à la puissance d'un entier.                                | Réalisable                                 | Function<br>Puissance(x:Real;b:Integer):Real;                                                                                                                                                                  |
| Soit à vérifier l'existence d'une chaîne dans un tableau de n chaînes.             | Réalisable                                 | Function<br>Existe(ch:String;T:TAB):Boolean;                                                                                                                                                                   |

## 2) Application 2

Soit à saisir les moyennes de n élèves ( $5 < n < 33$ ) dans un tableau. Il s'agit de trouver puis d'afficher:

- Le nombre d'élèves qui ont réussi.
- La plus basse moyenne.

### Analyse du programme principal

**Résultat :** Afficher le nombre d'élèves qui ont réussi.  
Afficher la plus basse moyenne de la classe.

**Traitement :** Chacun des traitements demandés peut être réalisé par une fonction. Ces fonctions seront appelées au niveau du programme principal dans un contexte d'affichage.  
Les données sont :  
La taille n qui doit être comprise entre 6 et 32  
Un tableau T de n moyennes où chaque moyenne doit être comprise entre 0 et 20.

**Algorithme :**

- 0) Début Algorithme Classe1
- 1) Répéter
  - Ecrire ("Donner le nombre d'élèves")
  - Lire(n)
  - Jusqu'à n Dans [6..32]
- 2) Pour i de 1 à n Faire
  - Répéter
    - Ecrire("Donner la moyenne de l'élève",i);Lire(T[i] ),
    - Jusqu'à (T[i]≥0) ET (T[i]≤20)
  - FinPour
- 3) Ecrire("Le nombre d'élèves ayant la moyenne est", **Nombre(n,T)**)
- 4) Ecrire("La moyenne la plus basse est ", **Minimum(n,T)**)
- 5) FinClasse1

**Déclaration des nouveaux types**

| Type(s)                 |
|-------------------------|
| TAB=Tableau de 32 réels |

**Déclaration des objets globaux**

| Objets  | Type/Nature | Rôle                                                |
|---------|-------------|-----------------------------------------------------|
| i       | Entier      | Compteur                                            |
| n       | Entier      | Nombre d'élèves                                     |
| T       | TAB         | Tableau contenant les moyennes                      |
| Nombre  | Fonction    | Permet de trouver le nombre d'élèves qui ont réussi |
| Minimum | Fonction    | Permet de trouver la plus basse moyenne             |

**Analyse de la fonction Nombre**

**Résultat :** Trouver le nombre d'élèves qui ont réussi

**Traitement :** Trouver le nombre d'élèves qui ont réussi est un traitement répétitif qui nécessite une structure itérative complète :

- On initialise le nombre recherché à zéro
- On parcourt le tableau T de 1 à n.
- On teste pour chaque valeur si elle est supérieure ou égale à 10 ; si c'est le cas, on incrémente le nombre de 1.

La taille n et le tableau T sont des données.

**Algorithme :**

- 0) Fonction Nombre(n:Entier; T:TAB):Entier
- 1)  $ner \leftarrow 0$   
     Pour j de 1 à n Faire  
         Si  $T[j] \geq 10$  Alors  
              $ner \leftarrow ner + 1$   
         FinSi  
     FinPour
- 2) Nombre  $\leftarrow ner$
- 3) FinNombre

**Déclaration des objets globaux**

| Objets | Type/Nature | Rôle                           |
|--------|-------------|--------------------------------|
| j      | Entier      | Compteur                       |
| ner    | Entier      | Nombre d'élèves qui ont réussi |

**Analyse de la fonction Minimum**

**Résultat :** Trouver la moyenne la plus basse

**Traitement :** Trouver la moyenne la plus basse est un traitement répétitif qui nécessite une structure itérative complète :

- On initialise le minimum à la 1ère moyenne du tableau
- On parcourt le tableau T de 2 à n.
- On teste pour chaque valeur si elle est inférieure au minimum prévu ; si c'est le cas, on affecte au minimum la valeur qui lui est inférieure.

*La taille n et le tableau T sont des données.*

**Algorithme :**

- 0) Fonction Minimum(n:Entier; T:TAB):**Réel**
- 1)  $m \leftarrow T[1]$   
     Pour j de 2 à n Faire  
         Si  $(T[j] < m)$  Alors  
              $m \leftarrow T[j]$   
         FinSi  
     FinPour
- 2) **Minimum**  $\leftarrow m$
- 3) FinMinimum

**Déclaration des objets locaux**

| Objets | Type/Nature | Rôle                         |
|--------|-------------|------------------------------|
| j      | Entier      | Compteur                     |
| m      | Réel        | Contient la moyenne minimale |

**Activité 5**

Allumez votre micro-ordinateur et, avec l'aide de l'enseignant, traduisez ces algorithmes en Pascal et enregistrez le programme source sous le nom "**Classe1**".

**3) Application 3**

Soit à saisir les moyennes de n élèves ( $n < 35$ ) dans un tableau T et leurs noms dans un tableau Nom. Il s'agit d'afficher le nom de chaque élève accompagné de son rang en classe.

**Exemple :**

Soit  $n = 5$

|          |      |      |       |       |       |
|----------|------|------|-------|-------|-------|
| <b>T</b> | 10.5 | 9.75 | 12.25 | 14.10 | 11.95 |
|          | 1    | 2    | 3     | 4     | 5     |

|            |        |        |       |        |          |
|------------|--------|--------|-------|--------|----------|
| <b>Nom</b> | Gabtni | Sarray | Faleh | Chedly | Chmengui |
|            | 1      | 2      | 3     | 4      | 5        |

Le programme doit afficher :

Gabtni a le rang 4  
 Sarray a le rang 5  
 Faleh a le rang 2  
 Chedly a le rang 1  
 Chmengui a le rang 3

**Analyse du programme principal**

**Résultat :** Afficher le nom de chaque élève accompagné de son rang en classe

**Traitement :** L'affichage des noms des élèves accompagnés de leurs rangs est un traitement répétitif qui nécessite une structure itérative complète:

- On parcourt les tableaux T et Nom
- On affiche à chaque fois le nom Nom[i] accompagné du rang qui est le résultat de l'appel d'une fonction Rang.

*Les données sont :*

*La taille n qui doit être comprise entre 1 et 34*

*Un tableau T de n moyennes où chaque moyenne doit être comprise entre 0 et 20*

*Un tableau Nom de n noms*

**Algorithme :**

- 0) Début Algorithme Classe2
- 1) Répéter
  - Ecrire ("Donner le nombre d'élèves :")
  - Lire(n)
  - Jusqu'à (n Dans [1..34])
- 2) Pour i de 1 à n Faire
  - Répéter
  - Ecrire("Donner la moyenne de l'élève",i)
  - Lire(T[i] )
  - Jusqu'à (T[i]≥0) ET (T[i]≤20)
- FinPour

- 3) Pour i De 1 à n Faire  
    Ecrire(Nom[i], " a le rang", Rang(T[i], T, n))  
    FinPour
- 4) FinClasse2

**Déclaration des nouveaux types**

| Type(s)                    |
|----------------------------|
| TAB1=Tableau de 34 réels   |
| TAB2=Tableau de 34 chaînes |

**Déclaration des objets globaux**

| Objets | Type/Nature | Rôle                                 |
|--------|-------------|--------------------------------------|
| i      | Entier      | Compteur                             |
| n      | Entier      | Nombre d'élèves                      |
| T      | TAB1        | Tableau contenant les moyennes       |
| Nom    | TAB2        | Tableau contenant les noms           |
| Rang   | Fonction    | Permet de trouver le rang d'un élève |

**Analyse de la fonction Rang**

**Résultat :** Trouver le rang d'un élève

**Traitement :** Trouver le rang d'un élève est un traitement répétitif qui nécessite une structure itérative complète:

- On initialise le rang à 1
- On parcourt le tableau T de 1 à n.
- On teste pour chaque valeur si elle est supérieure à la moyenne de l'élève; si c'est le cas, on incrémente le rang de 1.

*La moyenne d'un élève, la taille n et le tableau T sont des données.*

**Algorithme :**

- 0) Fonction Rang(m:Réel;n:Entier; T:TAB):**Entier**
- 1) r←1  
    Pour j de 1 à n Faire  
        Si (T[j]>m) Alors  
            r←r +1  
        FinSi  
    FinPour
- 2) Rang←r
- 3) FinRang

**Déclaration des objets locaux**

| Objets | Type/Nature | Rôle            |
|--------|-------------|-----------------|
| j      | Entier      | Compteur        |
| r      | Entier      | Rang d'un élève |

**Activité 6**

Allumez votre micro-ordinateur et, avec l'aide de l'enseignant, traduisez ces algorithmes en Pascal et enregistrez le programme source sous le nom " **Classe2**".

**III. Les procédures**

**Activité 1**

Citez les insuffisances des fonctions.

**Activité 2**

Citez des procédures standard que vous avez déjà utilisées. N'oubliez pas de citer aussi les paramètres nécessaires au fonctionnement de chacune d'elles. Par exemple, on peut citer :

- la procédure Delete (Ch,p,n) où Ch est la chaîne concernée par la suppression, p désigne l'indice à partir duquel commence la suppression et n est le nombre de caractères à supprimer.
- La procédure Read ou Readln où les paramètres sont les variables à lire.
- La procédure Clrscr (CLearSCReen) qui est une procédure non paramétrée et qui permet d'effacer l'écran.
- .....

**Activité 3**

Soit à insérer une deuxième chaîne mot2 dans une première chaîne mot1 à une position donnée, puis afficher la nouvelle chaîne obtenue.

|                                                                            |  |                                                                                |
|----------------------------------------------------------------------------|--|--------------------------------------------------------------------------------|
| <pre>mot1 ← "cable" mot2 ← "rta" Insérer (mot2, mot1,3) Ecrire(mot1)</pre> |  | <pre>mot1 := "cable"; mot2 := "rta"; Insert (mot2, mot1,3); Write(mot1);</pre> |
|----------------------------------------------------------------------------|--|--------------------------------------------------------------------------------|

Au début mot1 contenait la chaîne "cable", après l'insertion, mot1 contient désormais la chaîne "cartable".

**Activité 4**

Soit à remplir un tableau T par n caractères (avec  $6 \leq n \leq 30$ ). Il s'agit de répartir ces n caractères sur trois tableaux et les afficher :

- TL : un tableau de lettres
- TC : un tableau de chiffres
- TS : un tableau de symboles

**Exemple :**

Soit n = 10

|   |   |   |   |   |   |   |   |    |   |    |
|---|---|---|---|---|---|---|---|----|---|----|
| T | h | 4 | ! | K | } | 2 | r | \$ | 8 | d  |
|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8  | 9 | 10 |

On doit obtenir les tableaux suivants :

Soit  $n = 10$

|           |   |   |    |   |
|-----------|---|---|----|---|
| <b>TL</b> | h | K | r  | d |
|           | 1 | 2 | 3  | 4 |
| <b>TC</b> | 4 | 2 | 8  |   |
|           | 1 | 2 | 3  |   |
| <b>TS</b> | ! | } | \$ |   |
|           | 1 | 2 | 3  |   |

### Spécification du problème

**Résultat :** Afficher le contenu de trois tableaux où les caractères de T sont classés.  
L'affichage de chacun de ces trois tableaux est un traitement répétitif nécessitant une structure itérative complète où on parcourt le tableau concerné de 1 à sa taille spécifique et on affiche chacune de ses cases.

**Traitement :** Trouver le rang d'un élève est un traitement répétitif qui nécessite une structure itérative complète :

- On initialise les compteurs des trois tableaux à 0
- On parcourt le tableau T de 1 à n.
- On teste pour chaque caractère de T si c'est une lettre, un chiffre ou un symbole : pour chaque cas on crée une nouvelle case dans le tableau concerné en incrémentant son compteur, puis on affecte à cette case le caractère testé.

*Le tableau T de caractères et sa taille n sont des données.*

### Algorithme :

```

0) Début Algorithme Classement
1) Répéter
 Ecrire("Introduire le nombre de cases")
 Lire(n)
 Jusqu'à n Dans [6..30]
2) Pour i de 1 à n Faire
 Ecrire ("Donner le caractère de la case",i)
 Lire(T[i])
 FinPour
3) a←0 ; b←0 ; c← 0
 Pour i de 1 à n Faire
 Si (Majus(T[i]) Dans ["A".."Z"]) Alors
 a←a+1
 TL[a]← T[i]
 Sinon Si T[i] Dans ["0".."9"] Alors
 b←b+1
 TC[b]← T[i]
 Sinon
 c←c+1
 TS[c]← T[i]
 FinSi
 FinPour

```

- 4) Pour i de 1 à a Faire  
    Ecrire(TL[i])  
    FinPour
- 5) Pour i de 1 à b Faire  
    Ecrire(TC[i])  
    FinPour
- 6) Pour i de 1 à c Faire  
    Ecrire(TS[i])  
    FinPour
- 7) FinClassement

Au niveau de cette solution algorithmique, on distingue une partie redondante : celle où on affiche le contenu d'un tableau : le contenu de TL, le contenu de TC et le contenu de TS.

On peut éviter une telle redondance, en définissant une procédure permettant d'afficher le contenu d'un tableau. Cette procédure sera appelée en cas de besoin.

En supposant que cette procédure à définir est nommée Affiche, l'algorithme Classement devient:

- 0) Début Classement
- 1) Répéter  
    Ecrire("Introduire le nombre de cases")  
    Lire(n)  
    Jusqu'à n Dans [6..30]
- 2) Pour i de 1 à n Faire  
    Ecrire ("Donner le caractère de la case",i)  
    Lire(T[i])  
    FinPour
- 3) a←0 b←0 c← 0  
    Pour i de 1 à n Faire  
        Si Majus(T[i]) Dans ["A".."Z"] Alors  
            a←a+1  
            TL[a]←T[i]  
        Sinon Si T[i] Dans ["0".."9"] Alors  
            b←b+1  
            TC[b]← T[i]  
        Sinon  
            c←c+1  
            TS[c]← T[i]  
    FinSi  
    FinPour
- 4) Affiche(a, TL)
- 5) Affiche(b, TC)
- 6) Affiche(c, TS)
- 7) FinClassement

Il nous reste maintenant à définir la procédure Affiche.

#### Définition de la procédure Affiche

- 0) Procédure Affiche(m:Entier;F :TAB)
- 1) Pour i de 1 à m Faire  
    Ecrire(F[i])  
    FinPour
- 2) FinAffiche

La procédure Affiche est définie pour n'importe quel tableau F de taille m. F et m sont des **paramètres formels**.

La procédure Affiche est appelée trois fois, au niveau du programme principal Classement:

- Une première fois, en **substituant** m par a et F par TL,
- une deuxième fois, en **substituant** m par b et F par TC, et
- une troisième fois, en **substituant** m par c et F par TS.

TL,TC,TD, a, b et c sont des paramètres effectifs car ils constituent les paramètres qui sont manipulés réellement.

**L'appel d'une procédure se fait par son nom suivi des paramètres effectifs qui se substitueront aux paramètres formels lors de l'exécution de la procédure.**

### III.1. Syntaxes

**Syntaxe en algorithmique**

```

0- Procédure nom_procédure(liste des éventuels paramètres formels)
1-
2-
.
.
.
n- FIN nom_procédure

```

**Syntaxe en Pascal**

```

procedure nom_procédure(liste des éventuels paramètres formels);
Var
 {Partie déclarative des objets locaux s'ils existent}
Begin
 ;
 ;
 .
 .
 .
End;

```

#### Activité 5

Allumez votre micro-ordinateur et traduisez ces algorithmes en Pascal et enregistrez le programme source sous le nom "**Eclatement**".

### III.2. Les caractéristiques d'une procédure

- La définition d'une procédure est constituée de trois parties :
  - La partie en-tête comportant le nom de la procédure suivi, entre parenthèses, de la liste des paramètres formels.
  - La partie déclarative des éventuels objets locaux.
  - La partie formant le corps de la procédure.
- Pour que l'appel d'une procédure soit correct, il faut respecter les contraintes suivantes :
  - Une procédure est une instruction à exécuter. Donc elle ne peut pas figurer directement dans une expression.
  - Le nom de la procédure appelée doit correspondre exactement au nom utilisé lors de sa définition.
  - Les paramètres effectifs et les paramètres formels doivent se correspondre en nombre, type et ordre.

#### Activité 6

Afin de profiter des avantages de l'analyse modulaire, il s'agit de tout résoudre sous forme de modules. Ainsi, nous allons transformer l'algorithme **Classement** en une solution totalement modulaire.

A part l'affichage, on a les tâches suivantes :

- Saisie de n, la taille du tableau initial et remplissage de T. On va attribuer cette tâche à une procédure appelée Saisie, par exemple.
- Répartition des n caractères de T sur les tableaux TL, TC et TS. On va prévoir une procédure appelée Traitement, par exemple.

L'algorithme Classement devient :

- 0) Début Classement
- 2) Saisie(n,T)
- 3) Traitement(n,T,a,b,c,TL,TC,TS)
- 4) Affiche(TL,a)
- 5) Affiche(TC,b)
- 6) Affiche(TS,c)
- 7) FinClassement

Complétez la résolution de cette application.

### III.3. Les paramètres

#### a- Les types de paramètres :

Il existe deux types de paramètres :

- **Les paramètres formels** qui figurent dans l'en tête de la déclaration de la procédure et correspondent à des variables locales utilisées dans son corps.
- **les paramètres effectifs** utilisés dans l'instruction d'appel de la procédure et sont substitués aux paramètres formels au moment de l'appel.

### **b- Les modes de passages des paramètres :**

La substitution des paramètres effectifs aux paramètres formels s'appelle mode de passage de paramètres qui correspond à un transfert de données entre le programme appelant et le programme appelé. Il existe deux modes de passages de paramètres :

**1- Passage par valeur :** Le programme appelant transmet une valeur au programme appelé. Toute modification du contenu du paramètre dans le programme appelé est sans conséquence sur la valeur du paramètre effectif dans le programme appelant. Les paramètres transmis par valeur font communiquer l'information dans un seul sens : du programme appelant (programme principal, par exemple) au programme appelé (une procédure, par exemple).

**2- Passage par variable :** Le programme appelant et le programme appelé échangent le contenu d'un paramètre ce qui signifie que les paramètres transmis par variable font communiquer l'information dans les deux sens. En effet, toute modification du contenu du paramètre formel entraîne automatiquement la modification de la valeur du paramètre effectif.

**Remarque :** Dans l'entête de la procédure, il faut précéder les paramètres formels transmis par variable par le mot clé VAR.

### IV - Retenons

- Afin de faciliter la résolution d'un problème complexe et/ou de grande taille, on a intérêt à le décomposer en sous-problèmes indépendants et de taille réduite.
- Une fonction est un sous programme qui permet de calculer puis de renvoyer une valeur d'un seul type.
- Une fonction se compose de trois parties :
  - L'entête de la fonction
  - La déclaration locale
  - Le corps de la fonction
- Une procédure est un sous programme qui peut avoir plusieurs résultats.

## EXERCICES

## Exercice n° 1

Ecrivez une fonction **MAX2** qui donne le maximum de **deux** réels.

## Exercice n° 2

Ecrivez une fonction **MAX3** qui donne le maximum de trois réels.

## Exercice n° 3

On dispose de  $n$  réels rangés dans un vecteur **V**.

1/ Ecrivez une fonction **MAXn** qui donne le maximum de **V**.

2/ Ecrivez une fonction qui renvoie l'indice de la valeur maximale (dans le cas d'égalité la fonction retourne l'indice de la première occurrence de cette valeur).

3/ Ecrivez une procédure qui détermine puis affiche la valeur maximale.

## Exercice n° 4

Ecrivez une procédure qui permet de remplir un vecteur **MOYENNE** par **50** réels compris entre 0 et 20.

## Exercice n° 5

Ecrivez une procédure qui permet d'afficher les **50** réels d'un vecteur **MOYENNE**.

## Exercice n° 6

Soit  $T$  un tableau de  $n$  caractères ( $2 \leq n \leq 20$ ). On se propose d'afficher le nombre d'occurrences d'un caractère  $c$  donné, dans le tableau  $T$ .

Appliquez l'analyse modulaire pour résoudre ce problème.

## Exercice n° 7

Sachant que  $6 + 6/22 + 6/32 + 6/42 + \dots + 6/n^2$  tend vers  $\pi^2$ . On se propose de calculer puis d'afficher une valeur approchée de  $\pi^2$  avec une erreur maximale  $\varepsilon$ . La valeur de  $\varepsilon$  est une donnée.

Appliquez l'analyse modulaire pour résoudre ce problème.

## Exercice n° 8

Soit  $T$  un tableau de  $n$  entiers ( $2 \leq n \leq 20$ ). On se propose d'afficher la valeur maximale  $V_{\max}$  et la valeur minimale  $V_{\min}$  d'une séquence de  $T$  allant de  $i$  à  $j$ . ( $i$  et  $j$  sont deux entiers donnés vérifiant la condition  $i \leq j \leq n$ ).

Appliquez l'analyse modulaire pour résoudre ce problème.

## Exercice n° 9

On veut écrire un programme Pascal permettant de coder un message selon le procédé suivant : *permuter chaque caractère d'indice pair avec le caractère qui le précède.*

**Exemple :**

Le codage de la chaîne de caractères : "**Baccalauréat**" donne "**aBcclauaérta**".

Appliquez l'analyse modulaire pour résoudre ce problème.

### Exercice n° 10

Soit la somme  $S_n$  suivante :  $S_n = 1 + 3/2^2 + 5/3^3 + 7/4^4 + \dots + (2n-1)/n^n$

On se propose de calculer et d'afficher la somme  $S_n$  pour un entier  $n$  positif donné en utilisant la formule ci-dessus.

Appliquez l'analyse modulaire pour résoudre ce problème.

### Exercice n° 11

On se propose de:

- saisir une chaîne de caractères **Ch**
- parcourir la chaîne **Ch** et afficher l'occurrence de chacun de ses caractères.

Appliquez l'analyse modulaire pour résoudre ce problème.

### Exercice n° 12

L'horoscope chinois est composé de 12 signes. Il associe un animal à chaque année selon un cycle de 12 ans. Voici l'ordre du cycle :

|           |           |          |          |           |              |
|-----------|-----------|----------|----------|-----------|--------------|
| 1- Rat    | 2- Bœuf   | 3- Tigre | 4- Lapin | 5- Dragon | 6- Serpent   |
| 7- Cheval | 8- Bélier | 9- Singe | 10- Coq  | 11- Chien | 12- Sanglier |

1900 est une année du rat et 1905 est une année du serpent.

Les années du tigre et du dragon sont considérées bénéfiques.

Pour une année de naissance donnée, on se propose d'afficher son signe et un message indiquant si elle est bénéfique ou non.

#### Questions :

- 1/ Trouvez l'horoscope chinois des personnes nées en 1800, 1920 et en 1960
- 2/ Analysez ce problème en le décomposant en modules.
- 3/ Analysez les modules proposés et déduire les algorithmes de résolution.

### Exercice n° 13

On se propose d'écrire un programme qui permet de simuler le jeu suivant :

– A tour de rôle 4 joueurs lancent un dé.

– Pour chaque lancer, le nombre de points est calculé comme suit :

- Si la face obtenue est égale à 6 alors le joueur gagne 6 points et lance le dé une autre fois.
- Si la face obtenue est 4 ou 5 alors le joueur gagne 3 points
- Si la face obtenue 1, 2 ou 3 alors le joueur gagne 1 point

– Le jeu se termine quand un des 4 joueurs totalise 50 points. Ce joueur sera considéré comme gagnant.

- 1/ Analysez le problème en le décomposant en modules.
- 2/ Analysez chacun des modules proposés ainsi que le module principal.
- 3/ Déduisez les algorithmes des différents modules.

### Exercice n° 14

On se propose d'écrire un programme de cryptage d'un texte en utilisant le procédé suivant :

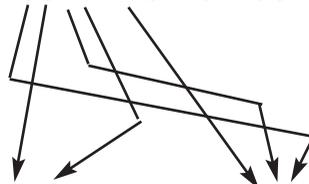
- Transformer toutes les lettres du texte en minuscule.
- Crypter le texte en suivant la méthode suivante :
  - le premier caractère du texte initial sera le dernier caractère du texte crypté
  - le deuxième caractère du texte initial sera le premier caractère du texte crypté
  - le troisième caractère du texte initial sera l'avant dernier caractère du texte crypté
  - le quatrième caractère du texte initial sera le deuxième caractère du texte crypté

et ainsi de suite.

- Les espaces seront remplacés par \*

**Exemple :**

Le texte initial : **les 7 merveilles**



Le texte crypté : **e\*\*evilselerm7sl**

**Questions**

- 1/ Analysez le problème et le décomposer en modules.
- 2/ Analysez chacun des modules de la question 1.
- 3/ Déduisez un algorithme pour chaque module ainsi qu'un algorithme pour le programme principal.

### Exercice n° 15

Un CODEC est un logiciel compresseur-décompresseur de fichiers. En effet, les suite de bits composant un fichier comportent des similitudes comme 10000111. Plutôt que de stocker la totalité de cet octet, on gagne de la place en écrivant 14031 (qui se lit **un quatre zéros 3 uns**). Cet octet (huit bits) retrouvera ensuite son format original à la décompression.

Il s'agit alors de saisir une chaîne de huit chiffres formée uniquement de 0 et de 1 pour désigner un octet puis la compresser suivant le principe de compression du CODEC et enfin l'afficher.

**Exemple :**

Si octet = "10010111"

Alors l'octet compressé est : "1201031"

**Questions**

- 1/ Analysez ce problème en le décomposant en au moins 2 modules.
- 2/ Analysez chaque module.
- 3/ Déduisez un algorithme pour le programme principal ainsi que pour chacun des modules.

## Annexe

### Table des codes ASCII.

| HEX | DEC | CAR | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0   | 0   | NUL | 20  | 32  |     | 40  | 64  | @   | 60  | 96  | `   |
| 1   | 1   | SQH | 21  | 33  | !   | 41  | 65  | A   | 61  | 97  | a   |
| 2   | 2   | STX | 22  | 34  | "   | 42  | 66  | B   | 62  | 98  | b   |
| 3   | 3   | ETX | 23  | 35  | #   | 43  | 67  | C   | 63  | 99  | c   |
| 4   | 4   | EOT | 24  | 36  | \$  | 44  | 68  | D   | 64  | 100 | d   |
| 5   | 5   | ENQ | 25  | 37  | %   | 45  | 69  | E   | 65  | 101 | e   |
| 6   | 6   | ACK | 26  | 38  | &   | 46  | 70  | F   | 66  | 102 | f   |
| 7   | 7   | BEL | 27  | 39  | '   | 47  | 71  | G   | 67  | 103 | g   |
| 8   | 8   | BS  | 28  | 40  | (   | 48  | 72  | H   | 68  | 104 | h   |
| 9   | 9   | HT  | 29  | 41  | )   | 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.