



LOGIQUE DE PROGRAMMATION

GENERALITES.-

 **Informatique – Ordinateur .-** L'informatique est le traitement de texte automatique de l'information. Ce traitement se fait à l'aide d'un ordinateur.

 **Algorithme – Programme.-** Un algorithme est une suite d'instructions permettant la résolution d'un problème. Dans ce cours, on apprendra à programmer en pseudo-codes, en des mots proches du langage usuel.

 **Ecriture d'un programme.-** Plusieurs étapes sont nécessaires à l'écriture d'un programme :

- 1) Donnée du programme. Ex: Résoudre $ax + b = 0$ dans \mathbb{R} .
- 2) Enoncé du problème: Le problème étant posé, il faut lui donner un énoncé clair. Ex: Trouver la ou les valeurs de (s) de x tel que $ax + b = 0$.
- 3) Analyse du problème.

Si $a = 0$,	si $b = 0$, il y a une infinité de solutions
	si $b \neq 0$, il n'y a pas de solution

Si $a \neq 0$,	$x = -b / a$
-----------------	--------------


- 4) Ecriture de l'algorithme:
début

```

réel a, b, x,
afficher ("Bonjour. Ce programme résout l'équation  $ax + b = 0$ ")
afficher ("Entrer les valeurs de a et b")
lire (a)
lire (b)
si (a = 0) alors
    { si (b = 0) alors
        afficher ("Il y a une infinité de solutions")
    sinon
        afficher ("Il n'y a pas de solution") }
sinon
    { x = -b/a
    afficher ("La solution de l'équation",  $ax' + b' = 0$  est', x)
    }
    afficher ("Au revoir")
fin

```

5) Ecriture au moyen d'un langage de programmation.

 **Analyse d'un algorithme.-** Généralement, un algorithme comprend les parties suivantes:

- 1) Déclaration des variables: réel a,b,x
- 2) Lecture des données: lire (a)
- 3) Traitement des données: si a = 0 alors...
- 4) Affichage des résultants: afficher ("La solution de ...")

Dans un algorithme, on retrouve:

- 1) des mots-clés: début et fin marquent le début et la fin du programme
- 2) des variables définies par le programmeur: a, x
- 3) des séparateurs: { et } marquent le début et la fin d'un bloc
- 4) des calculs mathématiques: $x = -b/a$
- 5) des structures de contrôle: si a = 0 alors ...

Il faut souligner qu'un ordinateur n'est pas intelligent et ne fait qu'exécuter des instructions qui lui sont données par le programmeur au moyen de mots-clés d'un langage de programmation. Aussi, même en pseudo-codes, on demandera à l'étudiant d'observer avec rigueur les structures d'un algorithme et l'orthographe des mots-clés. On prendra l'habitude de faire un décalage vers la droite à chaque début de structure et vers la gauche pour fermer la structure. Cette présentation permet une meilleure lecture globale d'un programme facilitant la correction d'éventuelles erreurs.

VARIABLES – LECTURE – AFFICHAGE.-


Variables.-

Une variable peut être définie comme un espace mémoire où on peut stocker des valeurs. Une constante est une entité dont la valeur ne varie pas tout au cours de l'exécution du programme. Définissons certains termes relatifs aux variables:


Affectation : donner une valeur à une variable

Initialisation : donner une première valeur à une variable

Incrémentation	: augmenter la valeur d'une variable de 1
Décrémentation	: diminuer la valeur d'une variable 1
Compteur	: Variable permettant de compter le nombre de fois qu'une opération est exécutée.
Témoin	: Variable qui change de valeur lorsqu'une condition est réalisée.

 **Nom de variables.**- Le nom de variable commence par une lettre et comporte des chiffres, des lettres et des caractères significatifs variant d'un langage de programmation à l'autre, on prendra l'habitude d'utiliser des chiffres, des lettres et le caractère `_`. Ex: noms corrects : salaire, nom1, sal-emp


noms incorrects : 1ex, a/3, sal emp

 **Déclaration de variables.**- On distinguera d'abord les variables numériques et les variables alphanumériques. Les variables numériques sont celles destinées au calcul. EX: salaire, note, compteur... Une variable alphanumérique est une variable dont la valeur est une suite de caractère (chiffres, lettres et autres). Ex: numéro de téléphone, adresse, nom,... Pour des raisons d'espaces mémoires, on divisera les variables numériques en réel et entier.

Ex: entier I, j, A

réel E, F, sal

caractère carac

 **Affectation.**- Pour donner une valeur à une variable, on utilise le symbole \leftarrow

$A \leftarrow 2$: la valeur 2 est mise dans la variable A.

☛ $A \leftarrow 2$ ne signifie pas Ajouter 2 à la valeur se trouvant en A.

$A \leftarrow B$: la valeur se trouvant dans B est mise dans la variable A.

☛ $A \leftarrow B$ Après cette instruction, B conserve sa valeur précédente.

La partie de droite peut aussi être une expression calcul: $A \leftarrow (B+2)*C$

$A \leftarrow A+1$: Donner à la variable A l'ancienne valeur de A plus 1.

$C \leftarrow 'e'$

EXERCICES.-

- 1) Ecrire un algorithme initialisant deux variables et permutant leur valeur.
- 2) Ecrire un algorithme permettant de permuter trois nombres: $A=2$, $B=4$, $C=5$.

Opérateurs mathématiques.-


$+$: addition

$-$: soustraction

$*$: multiplication

$/$: division

* et / ont priorité sur + et -. Cependant, on peut utiliser les parenthèses (et) pour changer l'ordre de priorité.

 **Exercice.-** Donner, dans un tableau, les diverses valeurs que prennent A,B,C.

$A \leftarrow 2$

$B \leftarrow 3$

$A \leftarrow B$

$C \leftarrow B+A$

$A \leftarrow C$

$B \leftarrow (A+B)*2$

✎ OPERATEURS DE COMPARAISON.-

- = : égal à
- < : inférieur à
- > : supérieur à
- < = : inférieur ou égal à
- > = : supérieur ou égal à
- < > : différent de

🔗 **Proposition logique.-** Une proposition logique est une proposition qui peut avoir deux valeurs : vrai ou faux. Ex: $A > 5$. Si A est supérieur à 5, $A > 5$ est vraie, sinon, c'est-à-dire si A est inférieur ou égal à 5, $A > 5$ est fausse.

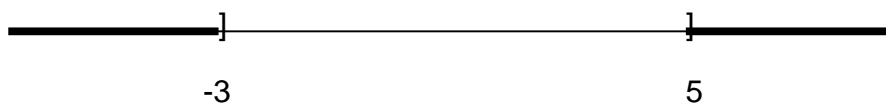
🔗 **Opérateurs logiques.-** Un opérateur logique permet de lier deux propositions logiques. Le résultat est une proposition logique. On utilisera trois opérateurs logiques ou, et, non.

ou	V	F
V	V	F
F	V	F

et	V	F
V	V	F
F	V	F

	V	F
non	F	V

$A \leq -3$ ou $A > 5$



COMMANDES D'ENTREES/SORTIES

lire (B)

Afficher ("Le salaire de "NOM" est de "A" gourds").

Ne pas confondre afficher ("A") et afficher (A). La première instruction signifie afficher le caractère tandis que la deuxième signifie afficher la valeur de la variable A.

Une commande de lecture est, en général, précédé d'une commande d'affichage:

Afficher ("Entrer la valeur de A:")

Lire (A)

La première instruction n'est qu'une commande d'affichage et ne saurait remplacer la seconde. Le libellé de la commande afficher peut être à la limite n'importe quoi et sert qu'à l'utilisateur ce qu'on attend de lui.

STRUCTURE DE CONTROLE.-

Par défaut, les instructions s'exécutent séquentiellement du haut vers le bas. Pour changer l'ordre d'exécution, on utilise les structures de contrôle. On en distingue deux types:

Les structures de condition

Les structures de répétition

Dans les lignes qui suivent:

Les mots en italiques sont des pseudo-codes

Les mots entre < et > doivent être remplacés par le programmeur

Les mots entre [et] sont optionnels.

STRUCTURE DE CONDITION SI...ALORS...SINON...

Syntaxe: Si (<condition>)alors

 {<bloc d'instruction-1>}

 [sinon

 {<bloc d'instruction-2 >}]

Si la condition est vraie, alors le bloc d'instruction-1 est exécuté. Si la condition est fausse, alors le bloc d'instruction-2 est exécuté. Dans le cas où la partie sinon est omise, si la condition est fausse, le programme continue après la structure si. Dans le cas où le bloc d'instruction ne comporte qu'une instruction, on omet {et}.

Exemple:

Début

Entier heure

Afficher ("Entrer l'heure (0-23):")

Lire (heure)

Si (heure>=6) et (heure <12) alors

 Afficher ("matin")

Sinon

 Si (heure< 6) ou (heure> = 18) alors

 {Afficher ("Nuit")

 Afficher ("aller dormir")}

 Sinon

Afficher ("Après-midi") fin

EXERCICES.-

- 1) Déterminer le plus grand de deux nombres
- 2) Classer trois nombres par ordre croissant.

STRUCTURE DE REPETITION POUR...

Cette structure est utilisée lorsqu'on connaît, avant le lancement de la structure, le nombre de fois que sera exécutées les instructions se trouvant dans la boucle

Syntaxe: Pour <compteur> allant de <début> à la <fin>
[pas<pas>]faire{<bloc d'instruction>}

Le compteur commence à la valeur <<début>>. Le bloc d'instruction est exécuté jusqu'à ce que le compteur dépasse la valeur <<fin>>. Avant le lancement d'une nouvelle série de la boucle, la nouvelle valeur du compteur est mise à ancienne valeur du compteur + pas. Le pas peut être positif ou négatif. Si le pas est omis, il est égal à 1.

En général, on ne modifie pas un compteur dans une structure pour...

Exemple:

Début

Entier I, c

Afficher (entrer une valeur 0..9")

Lire(c)

Pour I allant de 1 à 9 faire

Afficher (c," +",I,"=",c+i) fin

Exercices.-

- 1) Calculer la somme des n premiers entiers

2) Lire n nombres et calculer leur moyenne

STRUCTURE DE REPETITION REPETER...JUSQU'A

Syntaxe: répéter

{<bloc d'instruction>}

jusqu'à (<condition>)

Le bloc d'instruction est exécuté jusqu'à ce que la condition soit vraie. La condition est évaluée à la fin de boucle. La boucle est donc exécutée au moins une fois.

Exemple:

Début

Caractère: car

Répéter

Lire (car)

Jusqu'à (car = ".") fin

EXERCICE.-

Lire une phrase caractère et compter le nombre de voyelles. La phrase se termine par un point, un point-virgule ou un point d'interrogation.

STRUCTURE DE REPETITION TANT...QUE FAIRE

Syntaxe: tant que <condition> faire

{<bloc d'instruction>}

Dans cette structure, la condition est évaluée au début de la boucle.

Ne pas confondre la structure de condition <<si>> et la structure de répétition <<tant que>>.

Exemple:

Début

Caractère: car

Lire (car)

Tant que (car <>","")

Lire (car) fin

EXERCICE.-

- 1) Calculer le PPMC de deux nombres A et B. Indication: soit A le plus petit des deux nombres. Multiplier successivement A par les entiers 1..B. Le PPMC est obtenu dès que le produit obtenu est multiple de B.
- 2) Calculer le PGCD de deux nombres entiers A et B. Indication (algorithme d'Euclide) : Soit A le plus grand des deux nombres. Soit $Q=A/B$ et $R=A-B*Q$ (R:reste de la division entière – utiliser $A \bmod B$). Si R est nul, le PGCD est B sinon remplacer A par B et B par R et recommencer.

TABLEAUX.-

Les variables, que nous avons utilisées jusqu'à présent, sont des scalaires. Cependant, il est parfois nécessaire de regrouper des variables scalaires. La structure de tableau permet le regroupement d'éléments de même type, tel que les noms d'employés, les notes d'étudiants, les éléments d'une matrice dans un ensemble. Un nom unique sert de référence à l'ensemble des éléments et on accède à un élément en utilisant les indices.

Vecteur – Tableaux à une dimension.-

Dans un vecteur, on accède à chaque élément du tableau au moyen d'un seul indice. La dimension du tableau est le nombre maximal d'éléments du tableau. Les indices sont 1,2, ... dimension.

Syntaxe : déclaration <type> A (<dimension>)

Ex : entier A(20)

Caractère C (25)

Manipulation A (<indice>)

On manipule les éléments d'un tableau et non le tableau dans son ensemble.

Exemple :

Début

Entier : i,n, A(20)

Afficher ("Entrer le nombre d 'éléments : ")

Répéter

Lire (n)

Jusqu'à (n>0) et (n<20)

Pour i allant de 1 à n faire

{afficher ("A [",i,"]=")

lire (a(i))} fin

Exercices.-

- 1) Lire deux vecteurs de même dimension et calculer leur produit scalaire
- 2) Trouver le plus petit élément d'un vecteur
- 3) Modifier le vecteur 2,25,8,9,0,12 en 12,2,25,8,9,0.
- 4) Ecrire un algorithme permettant l'inversion d 'une chaîne de caractères.

Matrices – Tableaux à deux dimensions.-

Dans un tableau à deux dimensions, on accède à chaque élément au moyen de deux indices. Les éléments du tableau peuvent être considérés comme les éléments d'une matrice. Les dimensions du tableau sont le nombre de lignes et le nombre de colonnes de la matrice. Les indices sont 1,1 ;2 ;...1, nbrecols ;2,1 ;... ;nbreligs,nbrecols

Syntaxe : déclaration <type> A(<nbreligs>,<nbrecols>)

Ex : entier A(20,30)

 R  el C(25,15)

 Manipulation A(<indlig><,indcol>)

Exemple :

D  but

Entier : i,j,m,n,A(20,20)

Afficher ("entrer le nombre de lignes : ")

Lire (m)

Afficher ("Entrer le nombre de colonnes : ")

Lire (n)

Pour i allant de 1    m faire

 {pour j allant de 1    n faire

 lire (a(i,j))} fin

Exercices.-

- 1) Calculer la somme de deux matrices.
- 2) Calculer la transpos  e d'une matrice. (Utiliser une seule matrice)

TRIS.-

Un tri en informatique, est l'op  ration de mise en ordre, suivant un ou plusieurs crit  res, d'un ensemble de donn  es en vrac.

TRIS PAR RECHERCHE DU MINIMUM OU DU MAXIMUM.-

Cet algorithme recherche le minimum et le place en première position, puis recherche le minimum du reste de la liste et le place en deuxième position, et ainsi de suite.

Début

Entier $i, j, n, \text{min}, \text{posmin}, A[20]$

Lire (n)

Pour i allant de 1 à n faire

 Lire $A(i)$

 Pour i allant de 1 à $n-1$ faire

$\{\text{min} \leftarrow A(i)$

$\text{posmin} \leftarrow i$

 pour j allant de $i+1$ à n faire

$\{\text{si } \text{min} > A(j) \text{ alors}$

$\{\text{min} \leftarrow A(j)$

$\text{posmin} \leftarrow j\}$

$A(\text{posmin}) \leftarrow A(i)$

$A(i) \leftarrow \text{min}$ fin

TRI PAR INSERTION.-

Le premier élément est considéré comme une liste de triée. On prend le deuxième élément et on crée une liste triée. On prend le troisième et l'insère dans la liste triée précédente et ainsi de suite.

Début

Entier $i, j, x, A[20]$

Lire(n)

Pour i allant de 1 à n faire

Lire $A(i)$

Pour i allant de 2 à n faire

$\{x \leftarrow A(i)$

$j \leftarrow i-1$

tant que ($j > 0$) et ($x < A(j)$) faire

$\{A(j+1) \leftarrow A(j)$

$j \leftarrow j-1$

$A(j+1) \leftarrow x \}$ fin

TRI

A

BULLES.-

L'algorithme fait remonter progressivement les éléments <<légers>>.

Début

Entier $i, x, n, A[20]$

Caractère chang


```

Lire (n)
Pour i allant de 1 à n faire
    Lire A(i)
Répéter
    {chang ← "F"
    pour i allant de 1 à n-1 alors
        {si A(i) ← A(i+1) alors
            chang ← "V"}
        x ← A(i)
        A(i) ← A(i+1)
        A(i+1) ← x

Jusqu'à (chang = "F") fin

```

Autres Exercices.-

- 1) Lire une phrase caractère et déterminer le nombre de fois que la lettre <<d>> est suivi de la lettre <<e>>. La phrase se termine par un point.
- 2) Lire une phrase et déterminer le nombre de mots de la phrase. Un mot est suivi par un espace, une virgule, un point-virgule et le point.
- 3) Calculer n !
- 4) Trouver nombres d'occurrences d'un élément dans un vecteur.
- 5) Trouver la dernière position d'un élément dans un vecteur.
- 6) Trouver la ième position d'un élément dans un vecteur.
- 7) Trouver l'intersection de deux ensembles.

- 8) Trouver la réunion de deux ensembles.
- 9) Insérer un entier dans une liste triée par ordre croissant.
- 10) Calculer la trace d'une matrice carrée.
- 11) Calculer la somme des éléments de la deuxième diagonale d'une matrice carrée.
- 12) Trouver le produit de deux matrices.
- 13) Trouver toutes les positions d'un élément dans une matrice.
- 14) Trouver les plus grands éléments en ligne dans une matrice.
- 15) Trouver le vecteur somme en colonne d'une matrice.
- 16) Générer le triangle de Pascal.

Informatique, Ordinateur et Programmation

Informatique.- Science du traitement automatique de l'information par des ordinateurs. Le mot <<informatique>> est formé de la contraction de <<information>> et <<automatique>>. En informatique, on distinguera le <<hardware>> et le <<software>>.

Le <<hardware>> est constitué du matériel : l'ordinateur les périphériques, les câbles, etc... alors que le <<software>>, ce sont les logiciels ou programmes.

Architecture.- En informatique, c'est l'agencement des différents éléments d'un ordinateur. L'architecture se situe sur la carte mère de la machine et comprend un ou plusieurs microprocesseurs, des mémoires cachées, d'éventuels coprocesseurs, des circuits d'entrée-sortie et une mémoire centrale. Ces divers éléments sont reliés entre eux par des bus, sortes de voies de circulation de l'information à l'intérieur d'un ordinateur.

La structure d'un ordinateur comprend cinq éléments fondamentaux : une unité centrale de traitement (microprocesseur), des périphériques d'entrée et de sortie, des unités de stockage (mémoire externes et internes), et un bus chargé de véhiculer l'information entre les composants de la machine.

Le microprocesseur.- Au centre de l'architecture se trouve le microprocesseur, qui contient des registre (zones de stockage temporaire des données) et une unité de calcul arithmétique et logique. Il est li. Il est lié à la mémoire par deux bus : le bus d'adresses et le bus de données. Un microprocesseur constitue ainsi l'unité centrale de traitement d'un micro-ordinateur, mais est également utilisé dans de nombreux autres équipement (téléphones, cartes à puce, appareils électroménagers...).

Microprocesseur Pentium.- Le microprocesseur Pentium est fabriqué par Intel Corporation. Il contient plus de trois millions de transistors. Les matériaux des semi-conducteurs les plus communs utilisés dans la fabrication des puces informatique sont le silicium et le germanium.

Le premier microprocesseur, l'Intel 4004 de la société informatique Intel, est réalisé en 1971. Il est composé de 2300 transistors et exécute 60000 instructions par seconde. En comparaison, un microprocesseur moderne comme Intel Pentium 4 comprend plusieurs dizaines de millions de transistors et exécute plusieurs milliards d'instructions par seconde.

En fait, l'histoire du microprocesseur est intimement liée à celle du micro-ordinateur et de la micro-informatique en général. L'Intel 8080, est crée en 1974, est notamment l'un des premiers microprocesseurs (8bits) adaptés à un ordinateur individuel. Il a fortement influencé l'architecture du z80 de la société Zilog et, dans une moindre mesure, celle de la gamme des microprocesseurs 80x86 (80186, 80286, 80486) et les différents évolutions du Pentium, équipant hier ou aujourd'hui la grande majorité des PC.

Par ailleurs, les micro-ordinateurs Macintosh sont équipés de microprocesseur Motorola (famille du 68000, intégré au premier Macintosh commercialisé en 1984) et, depuis 1998, de microprocesseurs RISC (Reduced Instruction Set Computer) de la série Power PC (G3 ou G4), développés en collaboration par les firmes américaines Apple, IBM et Motorola.

Pionnier de la micro-informatique et créateur du premier microprocesseur, Intel domine le marché des microprocesseurs avec environ 80p. 100 de parts de marché. La société AMD suit de loin avec pratiquement 20p. 100 et toutes les autres sociétés représentent moins de 100 du marché. La situation est un peu différente pour toutes les autres sociétés représentent moins de 1p.100 du marché. La situation est un peu différente pour les microprocesseurs spécialisés où d'autres entreprises sont encore présentes (notamment Texas Instruments aux Etats-Unis et le SMT Electronic en Europe).

Un microprocesseur est composé d'une mémoire morte (ROM), de registres (zones de stockage temporaire), d'une unité de calcul arithmétique et logique, et d'un séquenceur. Les instructions sont traitées dans le séquenceur où elles sont décomposées en un ensemble de micro-instructions, dont la suite forme un microprogramme. Une micro-instruction est une opération de bas niveau du microprocesseur (par exemple, transfert du contenu d'un registre vers un bus, envoi d'une commande à l'unité de calcul, etc....). Ces micro-instructions sont ensuite décodées et exécutées.

On mesure la puissance d'un microprocesseur par la taille de ses registres et par sa fréquence d'horloge. La taille des registres s'exprime en nombres de bits. Elle n'a cessé de croître grâce aux progrès de la technologie des circuits intégrés : aujourd'hui, on trouve sur le marché des microprocesseurs 8 bits, 16 bits, 32 bits, 64 bits. Chaque microprocesseur dispose d'une horloge qui émet un signal à intervalle régulier. La fréquence d'horloge, exprimée en mégahertz (MHz) ou gigahertz (GHz), correspond au nombre de millions (ou de milliards) de signaux émis par l'horloge en une seconde. Le délai entre deux signaux s'appelle un cycle, et pendant un cycle d'horloge, une micro-instruction (ou plusieurs, si elles ne correspondent pas au même bus) peut-être exécutée. Par conséquent, plus la fréquence d'horloge de microprocesseur que lorsqu'ils ont une architecture identique (un PowerPC G4 de Motorola ne peut se comparer directement à un Pentium 4 d'Intel).

Par ailleurs, les fréquences d'horloge varient en fonction des types de microprocesseurs : les microprocesseurs généraux comme le Pentium 4 peuvent être cadencés à quelques gigahertz (GHz), les versions pour mobiles tournent autour de 1GHz et les microprocesseurs embarqués sur des PDA (Personal Digital Assistant, assistant numérique personnel) sont généralement cadencés à quelques centaines de mégahertz (MHz).

Les Bus.- Lorsque l'on souhaite accéder à une donnée présente en mémoire, son adresse, c'est-à-dire son emplacement dans la mémoire, transite sur le bus d'adresses. Par contre, c'est sur le bus des données que circulent les bits d'information correspondant à la valeur de cette donnée. Enfin, un bus d'entrée-sortie permet de transfert de données entre les circuits d'entrée-sortie et les ports d'ordinateur. Chaque interface de l'ordinateur (série, parallèle, SCSI, USB, etc.) est gérée par un ou plusieurs circuits électronique dédiées.

La mémoire.- La mémoire de l'ordinateur est l'organe permettant d'enregistrer, de stocker et de restituer des données. Par extension, on parle de mémoire de masse pour désigner les unités de stockage externes d'un ordinateur (disque dur, disquettes, etc.). La mémoire est généralement constituée de circuits électroniques à base de semi-conducteurs.

La mémoire, souvent appelé <<mémoire centrale>>, peut être organisée en blocs indépendants les uns des autres. Elle sert à la fois au stockage des données et des programmes en attente ou en cours d'exécution. Il est à noter qu'il n'y a pas de différence physique entre la mémoire <<programme>> et la mémoire <<donnée>>. C'est cette architecture particulière qui donne sa souplesse et sa puissance à l'ordinateur. On peut ainsi considérer une instruction de programme comme une donnée, effectuer une opération arithmétique ou logique et changer le comportement d'un programme en fonction de son déroulement.

Lorsqu'une application requiert une portion de la mémoire de l'ordinateur, elle en fait la demande au système d'exploitation, qui se charge alors d'allouer au programme un espace mémoire. On distingue deux types d'allocations : l'allocation statique, où la portion de mémoire demeure attribuée au programme jusqu'à son achèvement, et l'allocation dynamique, où la mémoire réservée est désallouée lors de l'exécution du programme.

La mémoire interne d'un ordinateur est séparée en deux sections : la mémoire morte, qui ne comporte que des données accessibles en lecture, et la mémoire vive qui comprend des données accessibles en lecture et écriture.

Mémoire Morte(Rom).- La mémoire morte, encore appelé ROM (Read Only Memory), contient des données enregistrées qui ne peuvent être modifiées par l'utilisateur. Toutefois, il est possible d'écrire sur certains types de mémoire morte, désignés par les sigles PROM (Programmable Read Only Memory) et EPROM (Erase Programmable Read Only Memory).

Une PROM autorise l'écriture de données grâce à un dispositif appelé programmeur PROM ; lorsqu'une PROM est programmée, elle est dédiée à une fonction donnée et ne peut être reprogrammée. La production des PROM n'étant rentable qu'à grande échelle, ce type de mémoire est notamment utilisé lors de la phase de conception d'un logiciel.

A la différence d'une PROM, une EPROM peut être effacée puis reprogrammée pour un autre usage. On procède généralement à cet effacement en retirant le couvercle de protection de la puce puis en soumettant le matériau semi-conducteur à rayonnement ultraviolet. Une EPROM s'avère fort utile pour charger une instruction en code machine dans un prototype de logiciel lorsque le coût de fabrication de PROM se relève trop onéreux. En effet, bien que plus coûteux qu'une PROM, la EPROM peut apparaître plus économique à l'usage, si les modifications doivent être nombreuses.

Mémoire Vive (Ram).- La mémoire vive, également nommée RAM (Random Access Memory), est une mémoire effaçable qui peut être reprogrammée par l'utilisateur. Il en existe essentiellement deux types : LA RAM statique et la RAM dynamique.

La RAM dynamique est constituée de circuits intégrés contenant des condensateurs qui se déchargent au cours du temps, si bien qu'une puce Ram doit être continuellement rafraîchi. Durant ce rafraîchissement, elle ne peut être lue, ce qui provoque un ralentissement. Malgré cet inconvénient, la RAM dynamique est beaucoup plus utilisée que la RAM statique, car plus économique en raison de sa grande simplicité interne.

La RAM statique se compose de semi-conducteurs pourvus du circuit logique appelé bascule (flip-flop), qui retient l'information stockée aussi longtemps qu'il est alimenté (voir Electronique). Une puce RAM statique n'enregistre à peu près qu'un quart des données stockées par une puce RAM dynamique de complexité équivalente, mais en revanche, elle ne nécessite pas de régénération et s'avère souvent plus rapide qu'une RAM dynamique. Son utilisation est plutôt réservée à la mémoire cache, portion de mémoire vive dans laquelle sont copiés des

données ou des éléments de programmes fréquemment utilisées. La RAM statique permet ainsi de conserver temporairement de l'information.

Les Mémoires caches.- Les mémoires caches sont des petites zones de stockage coûteuses mais beaucoup plus rapides que la mémoire centrale. Elles permettent d'augmenter sensiblement les performances de l'ordinateur en préchargeant données et instructions qui sont nécessaires. Ce mécanisme permet d'éviter que l'unité centrale ne soit inactive dans l'attente d'un transfert avec la mémoire. Toutefois, il est limité par les capacités d'anticipation du comportement du programme en cours d'exécution par le système d'exploitation.

Réseau informatique.- Ensemble de connexions entre ordinateurs et périphériques, un réseau informatique permet à ses utilisateurs d'envoyer et de recevoir des informations de leur poste de travail. Les communications entre chaque poste de travail s'effectuent soit par câbles, soit par lignes téléphoniques grâce à des modems qui convertissent les signaux analogiques en signaux numériques ou inversement. L'ensemble de ces communications est contrôlé par des serveurs, ordinateurs chargés de partager les ressources des postes de travail (par exemple, les imprimantes). La connexion principale au réseau est figurée par la ligne rouge, les connexions locales par des lignes bleues.

Elle permet, en outre, de diffuser l'information par le biais de réseaux informatiques, dont internet constitue l'exemple le plus marquant. Enfin, en s'associant aux techniques de télécommunication, elle facilite largement l'administration des bureaux en proposant ses services de télématique et de bureautique.

En contrepartie, l'informatique a engendré un nouveau type de délit (et de <<criminels>> : les hackers), le délit informatique consistant à pénétrer dans des systèmes informatiques (voir sécurité informatique).

Périphérique.- En informatique, c'est un matériel relié à l'ordinateur par divers types de connexions, internes ou externes, et contrôlé par son microprocesseur.

Une imprimante, un modem ou un joystick constituent ainsi des périphériques de l'ordinateur.

Chaque périphérique permet une utilisation supplémentaire des capacités de l'ordinateur. C'est pourquoi un lecteur de disquette, un clavier, un écran ou une souris sont souvent considérés comme des extensions du système et non comme des périphériques, car ils représentent les sources principales d'entrée et de sortie de la plupart des ordinateurs.

Programme.- C'est une suite d'instructions interprétées puis exécutées par un ordinateur. Un programme peut désigner la version source tapée au clavier ou la version exécutable en langage machine.

Lorsqu'un programmeur conçoit un programme, il le saisit sur son ordinateur, en utilisant généralement un langage évolué. Ce programme source, qui doit comprendre toutes les instructions et les fichiers nécessaires à son interprétation, est ensuite compilé en un ensemble d'instructions écrites en langage machine, suivant un format qui en autorise le changement en mémoire de l'exécution.

Cette nouvelle version du programme source, appelée programme exécutable, est alors prête à fonctionner. En interdisant l'accès aux instructions codées en langage machine, un programme exécutable conserve en son sein la confidentialité des algorithmes mis au point par le programmeur. Ainsi, une application de traitement de texte constitue un programme exécutable : l'utilisateur n'a pas besoin de modifier le programme pour l'exécuter, et quand bien même il le désirerait, il n'en aurait pas la possibilité.

Développement d'un programme.- La réalisation ou développement d'un programme passe par une phase conceptuelle indispensable durant laquelle le programmeur, appelé encore développeur, écrit ce programme au moyen d'instructions appartenant au langage de programmation choisi. Mais le seul fait de connaître un langage n'implique pas forcément la création d'un bon programme. Des connaissances supplémentaires peuvent, en effet, être nécessaires, telles que la maîtrise de la théorie des algorithmes, la conception de l'interface utilisateur, la connaissance des réseaux, des serveurs de données, etc. Le développement d'un programme ou d'un logiciel s'inscrit dans un cycle qui se décompose en une succession d'étapes bien précises : élaboration du cahier des charges, analyse du système et du problème à résoudre, codification, validation et test, installation et déploiement, et enfin maintien en condition opérationnelle (maintenance).

La programmation adoptée dépend souvent du type d'application que l'on souhaite développer, ce choix prenant en compte, suivant les cas, des critères de rapidité, de facilité de traitement du graphisme ou du calcul, etc. Il existe de nombreuses types de programmation couramment employés, parmi lesquels on peut mentionner les programmations ascendante, descendante, linéaire, logique, modulaire structurée et orientée-objet.

Programmation ascendante.- Dans ce type de programmation, les fonctions de plus bas niveau sont développées, programmées et testées en premier, puis assemblées pour former les fonctions de niveau plus élevé.

Programmation descendant.- La méthodologie de la programmation descendant définit, en premier lieu, les fonctions du programme à son niveau le plus élevé pour subdiviser ensuite chacune d'entre elles en tâches de niveau inférieur.

Programmation modulaire.- En programmation modulaire, un programme est décomposé en plusieurs modules qui sont compilés séparément. Chaque module exporte des éléments spécifiés (constantes, types de données, variables, fonctions, procédures), tous les autres demeurant internes au module. Les modules clarifient et régularisent les interfaces entre les grandes parties du programme, favorisant par conséquent la fiabilité de l'ensemble.

Programmation structurée.- Un programme écrit en programmation structurée présente un déroulement net, une conception claire et un certain degré de structure hiérarchique. Ce type de programme offre plusieurs avantages, comme celui d'être facilement compréhensible et modifiable par d'autres programmeurs.

Niveau d'un langage.- Il existe différents types de langage, allant du plus rudimentaire au complexe, que l'on classe généralement en deux familles : les langages de bas niveau et les langages évolués. On y ajoute parfois une autre catégorie, les langages de quatrième génération.

Langage de bas niveau.- Les langages de bas niveau sont des langages proches du langage machine ou des langages offrant peu d'instructions de types de données. En général, chaque instruction écrite dans un langage de bas niveau correspond à une instruction machine. Le langage machine et le langage assembleur sont considérés comme des langages de bas niveau.

Langage machine.- Le langage machine représente le langage dans lequel s'exprime le résultat final d'une compilation de langage assembleur ou d'un haut niveau quelconque. Constitué de <<0>> et <<1>>, ce langage est chargé et exécuté par le microprocesseur. Appelé également code machine, il constitue le seul langage réellement <<compris>> par l'ordinateur, tous les autres langages correspondant à des formes de structuration du langage humain.

Langage assembleur.- Le langage assembleur est un langage de programmation de bas niveau, où chaque instruction correspond à une machine unique. Le jeu d'instructions d'un tel langage est donc associée à un certain type de processeur. Ainsi, les programmes écrits en langage assembleur pour un

processeur particulier doivent être réécrits pour tourner sur un ordinateur équipé d'un processeur différent. Après écriture d'un programme en langage assembleur, le programmeur fait alors appel à l'assembleur spécifique du processeur, qui traduit ce programme en instruction machine. Le langage assembleur peut être préféré à un langage de haut niveau lorsque le programmeur recherche une vitesse d'exécution élevée ou un contrôle étroit de la machine. En effet, les programmes écrits dans ce type de langage tournent plus vite et occupent moins de place que ceux produits par un compilateur. En outre, ils donnent au programmeur la possibilité d'agir directement sur le matériel (processeur, mémoire, affichage et connexion d'entrées / sorties).

Langage évolué.- Les langage évolués, dits aussi de haut niveau ou de troisième génération, sont des langages informatiques offrant un certain niveau d'abstraction par rapport au langage machine, et manipulant des structures syntaxiques telles que les déclarations, les instructions de contrôle, etc. Usuellement, le terme <<évolué>> désigne tout langage de niveau supérieur à celui du langage assembleur.

Les langages évolués sont classés en trois grandes familles : les langages procéduraux, les langages orientés-objets et les langages orientés-listes. On retrouve ainsi dans la famille des langages procéduraux : le FORTRAN, le COBOL, le BASIC, l'ADA, le PASCAL et le C, dans la famille orientés-objets : le C++ et le java et dans la famille des langages orienté-listes : le LISP.

Langages procéduraux.- Les langages procéduraux sont des langages où la procédure (suite d'instructions) constitue l'élément de base. La plupart des langages évolués sont des langages procéduraux.

Programmation orientée-objet.- En programmation orientée-objet, un programme est considéré comme un ensemble d'objets distincts, constitués eux-mêmes d'ensembles de structures de données et de routines (sous-programmes) intégrées. Chaque objet appartient à une classe qui définit les structures de données et les routines qu'il contient. Si, par exemple, on crée une classe nommée <<cartable>>. Toute classe peut donc être utilisée comme variable dans ce type de programme, dont les objets ainsi interagissent.

Langage orientés-listes.- Les langages orientés-listes peuvent être apparentés aux langages orientés-objets, à la différence près qu'ils manipulent non pas des objets mais des listes, c'est-à-dire des structures de données multi-éléments à organisation linéaire.

Langage de quatrième génération.- Les langages de quatrième génération (L4G en abrégé), conçus pour l'interaction avec le programmeur, qualifient souvent les langages propres aux bases de données. Se situant un cran au-dessus de langages tels que le C++ ou le COBOL, ils se composent d'un jeu d'instructions s'apparentant à des macro-instructions, séquences d'instructions prédéfinies

auxquelles on accède par une commande très simple. Toutefois, ces langages conservent un aspect hybride, dérivant le plus souvent des langages évolués.

Compilateur et Interpréteur.- Les principales différences entre un compilateur et un interpréteur sont :

Un programme ne peut être exécuté qu'une fois compilé ;

La détection de beaucoup d'erreurs se fait durant la phase de compilation ; c'est le cas des erreurs de syntaxe, mais, suivant les langages, d'autres vérifications peuvent aussi avoir lieu ;

Les programmes compilés s'exécutent souvent plus rapidement que leurs équivalents en langage interprété, du fait que le langage machine est plus rapidement exécuté par le microprocesseur qu'un langage dont les instructions doivent être décodées et interprétées une par une.

Bogue.- En général, un bogue est dû à une erreur de programmation ou de logique qui amène à un fonctionnement défectueux ou des résultats incorrects. On distinguera :

Des erreurs de syntaxe.

Des erreurs en cours d'exécution (runtime error) comme lors d'une division d'un nombre par zéro.

Des erreurs logiques : le programme ne signale pas d'erreur, mais les résultats sont incorrects.

Internet.- C'est le réseau télématique international d'origine américaine. Constituant à ce jour le plus grand réseau du monde, Internet est accessible aux professionnels comme aux particuliers.

Internet est issu du réseau Arpanet, qui a été conçu dans les années 1960 par l'ARPA(Advanced Research Project Agency) pour le département américain de la Défense. A l'origine, il s'agit d'un réseau coopératif d'ordinateurs permettant le partage de données stockées sur des serveurs distants, ainsi que l'échange de messages électroniques (e-mails). Réseau à usage militaire, Arpanet s'étend alors progressivement aux universités amér américaines dans les années 1970, notamment l'Université de Californie (UCLA) et l'université de Stanford, avant d'être remplacé en 1990 par le réseau Internet, destiné dans un premier temps à la recherche civile. En 1991, Tim Berners-Lee du CERN à Genève met au point d'interface d'Internet appelée World Wide Web, qui permet d'ouvrir le réseau au grand public en simplifiant les procédures de consultation des sites. En janvier 1992, l'Internet Society (ISOC) voit le jour avec pour objectif de promouvoir et de coordonner les développements sur Internet. L'année 1993 voit l'apparition du premier navigateur ou butineur (browser), supportant le texte et les images.

Cette même année, la NFS (National Science Foundation) mandate une compagnie pour enregistrer les noms de domaine.

D'un point de vue technologique, Tim Berners-Lee, l'inventeur du Web, crée en 1994 le consortium W3C (World Web Consortium), qui a pour objectif de favoriser l'interopérabilité sur le Web, c'est-à-dire le développement de normes. Par exemple, le W3C proposera en 1998 la norme XML. (extensible Markup Language), qui définit un langage de balisage étendu pour le Web (voir HTML, langage).

Aujourd'hui, Internet permet à tout individu d'accéder à une multitude de données se présentant sous divers formats : textes, bases de données, images, sons (MP3), vidéos, etc. Il offre de nombreux services aux professionnels, comme aux particuliers : réserver un hôtel en Asie, installer un encart publicitaire aux Etats-Unis, consulter les catalogues de constructeurs automobiles, passer des commandes, acheter ou vendre aux enchères un bien, visiter des musées obtenir des informations officielles, etc.

La topologie du réseau télématique Internet, basée sur l'interconnexion d'un ensemble de réseaux distincts, permet d'interconnecter à travers le monde un nombre impressionnant de machines informatiques, qui peuvent s'échanger toutes sortes d'informations. Les ordinateurs personnels et les stations de travail sont connectés à un réseau local (Local Area Network, LAN), soit par le biais d'une connexion directe par câble au réseau local. Il existe d'autres modes de transmission de données qui permettent de se connecter à un réseau, comme les lignes T1 ou les lignes dédiées. Les ponts et les concentrateurs relient divers réseaux les uns aux autres. Les routeurs transmettent les données via les réseaux et déterminent le meilleur trajet pour les acheminer.

Pour accéder au réseau Internet, l'utilisateur doit posséder un ordinateur, une ligne téléphonique commutée (ou dans le cas des grandes entreprises, des universités et autres, un modem ou une carte réseau, un kit de connexion à internet consistant en un accès à un réseau local (le provider) relié à internet ou abonnement auprès d'un fournisseur d'accès Internet (AOL, Wanadoo, Club-Internet, etc.). Par ailleurs, pour <<surfer>> sur Internet, l'utilisateur doit également disposer d'un logiciel de navigation (Netscape ou Explorer étant les plus connus) l'aidant à parcourir les sites du réseau, et éventuellement d'un logiciel de messagerie électronique lui permettant d'envoyer ou de recevoir du courrier électronique.

Les informations présentes sur le réseau Internet sont, le plus souvent, publiées dans des pages au format HTML (Hyper Text Markup Language). Ce format permet l'introduction dans le texte de liens à d'autres pages ou d'autres sites, appelés <<liens hypertextes>>. Ces liens permettent de <<naviguer>> dans le réseau Internet. Un autre format tend à se développer sur Internet : le format XML (eXtended Markup Language). Outre les possibilités offertes par le format

HTML. Le langage XML permet de mieux structurer l'information et de séparer clairement la forme et le contenu.

L'importante quantité d'informations en tout genre rend parfois difficiles la navigation et la recherche rapide. C'est pourquoi certains sites possèdent des moteurs de recherche, outils permettant de lancer une recherche par un mot ou une expression sur tous les sites Internet recensés par ce système, et de trouver ainsi les documents susceptibles de renfermer l'information désirée par l'utilisateur.

Internet sert de support à plusieurs protocoles et services : FTP (File Transfer Protocol) pour le transfert de fichiers (téléchargement), SMTP (Simple Mail Transfer Protocol) pour l'envoi et la réception de courriers publics, HTTP (Hyper Text Transfer Protocol) pour le protocole de transmission hypertexte, Web pour la publication d'informations, etc.

Pour faire fonctionner ce vaste réseau, Internet met en œuvre des protocoles de communication. Les deux protocoles de bases sont TCP (Transmission Control Protocol) et IP (Internet Protocol). Ils sont référencés sous le vocable TCP/IP. TCP récupère les informations à transmettre, IP les transforme en paquets et délivre à la couche chargée du transport (composant électronique).

Toutefois, le rapide développement d'Internet à l'échelle mondiale entraîne actuellement une évolution de ces protocoles, qui frôlent la situation.

Aujourd'hui, parmi tous les réseaux, Internet peut être considéré comme le réseau spécialisé dans l'information : son but n'est plus, comme à l'origine, de transmettre quelques lignes à partir de messageries, mais d'échanger des documents électroniques, des données informatisées, des informations économiques, des schémas, des sons, etc. En outre, grâce aux récents progrès réalisés dans la transmission et la compression des données, Internet donne maintenant accès à une information de plus en plus immédiate. C'est pourquoi on a assisté ces derniers temps à un développement exponentiel de ce réseau, les sociétés de services et les producteurs d'informations coopérant pour trouver de nouveaux marchés par le biais d'Internet.

Internet s'est rapidement révélé être un remarquable outil de marketing, contribuant largement au développement du commerce électronique (e-business). De nombreux sites Web à pour but commercial se sont développés, notamment les sites des agences de voyage et des chaînes d'hôtellerie-restauration qui présentent leurs services en ligne, offrant ainsi aux utilisateurs la possibilité de réserver et de payer leur billet de transport et leur chambre d'hôtel sans se déplacer. Liaison Agency Tim Crosby.

En quelques années, la notoriété d'Internet est ainsi passée de la simple découverte à une explosion de services intéressant les professionnels comme les particuliers. Dans le domaine économique, Internet se présente comme un

outil de tout premier plan, offrant aux entreprises de nombreux services interactifs : marketing en direct, publicités, affiches commerciales, tarifs, documentations techniques, bases de données à forte valeur ajoutée, etc. Le commerce électronique est ainsi devenu un véritable secteur d'activité où le client a la possibilité, tout en restant chez lui, de faire ses courses, réserver un billet d'avion ou participer à une discussion sur un thème particulier avec des intervenants de différents pays.

Système binaire.- Le système binaire joue un rôle important en informatique. Le Bit est une abréviation de Binary digit, 0 ou 1 dans le système de numération binaire. En traitement ou en stockage de l'information, le bit est la plus petite unité d'information manipulable par un ordinateur, et peut être physiquement représenté par une impulsion unique sur un circuit, ou par une petite zone d'une surface de disque, capable de stocker un 0 ou un 1. Considéré isolément, un bit a peu de signification : groupés par huit, les bits forment des octets ou bytes qui peuvent représenter différentes informations, en particulier les lettres de l'alphabet et les chiffres 0 à 9.

Comme deux chiffres (appelés bits en informatique) suffisent dans le système binaire, cette numération est utilisée dans les ordinateurs, où tout nombre binaire peut, par exemple, correspondre aux positions d'une série d'interrupteurs marche-arrêt (on-off). La position marche (on) correspond à 1 et la position (off) à 0. On peut également utiliser des zones magnétisées sur une bande ou sur un disque magnétique pour représenter des nombres binaires : une zone magnétisée correspond au chiffre 1 et son absence indique le chiffre 0. Dans les ordinateurs, des circuits logiques effectuent les différentes opérations arithmétiques sur les nombres écrits dans le système binaire.

Un nombre binaire est représenté sous la forme d'une chaîne binaire, qui est une suite de symboles 0 et 1 : un exemple de chaîne binaire est 11001101. Chaque symbole 0 et 1 est appelé un bit. Chaque bit correspond à une valeur, qui est plus ou moins importante selon sa place dans la chaîne : plus le bit est placé à gauche, plus il correspond à une grande valeur. Toutes ces valeurs sont des puissances de 2 : 1, 2, 4, 8, etc. une chaîne binaire composée de 8 bits est appelée un octet.

La conversion du binaire dans la base de calcul 10 (nombres classiques utilisant les chiffres de 0 à 9) se fait de la manière suivante : on assigne d'abord à chaque position une valeur : $2^0 = 1$ pour le bit le plus à droite, $2^1 = 2$ pour celui qui est situé à sa gauche, $2^2 = 4$ pour le suivant, et ainsi de suite jusqu'à la fin de la chaîne. On multiplie ensuite toutes ces valeurs par les bits correspondants pour trouver la valeur de chaîne binaire.

La conversion de la chaîne binaire 11001101 dans la base de calcul 10 est décrite par la relation suivante :

$$11001101 = 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 128 + 64 + 8 + 4 + 1 = 205$$

Inversement, pour convertir un nombre décimal en nombre binaire, on divise le nombre et les restes successifs par 2. Ainsi, 71 correspond à 1000111.

Compter en binaire

Système hexadécimal.- Le système hexadécimal est lui aussi utilisé en informatique. Il utilise 16 symboles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (dix), B (onze), C (douze), D (treize), E (quatorze) et F (quinze). Par exemple, le nombre 2EF du système hexadécimal correspond au nombre $(2 \times 16^2) + (14 \times 16^1) + (15 \times 16^0) = 751$ dans le système décimal.