Université de Jijel
Département de science de la terre
(Géologie)

Support de cours du module Informatique
1ere année Géologie

I. Notions de base sur l'informatique et structure d'un ordinateur

1. L'informatique

1.1 Notion de traitement automatique

a. **Un traitement :** est l'ensemble des opérations effectuées sur des données afin de les rendre utilisables selon le processus suivant:
L'entrée : introduire des données dans la machine.
La transformation : appliquer des actions sur les données.
La sortie : restituer les résultats.

b. **Traitement automatique** : Un traitement est dit automatique lorsqu'il est effectué par une machine automatique, inversement au traitement manuel.
Un traitement est dit manuel lorsqu'il est effectué par l'être humain.

1.2. Notion d'information: L'information est un ensemble de renseignements compréhensibles par l'esprit humain et qui prend un sens pour lui. D'autre part : les informations ne sont intéressantes que parce qu'elles portent un sens que se transforme en connaissance dans l'esprit humain. Les informations permettent donc de construire ou d'enrichir une connaissance sur un sujet

1.3 Définition de l'informatique et les domaines d'application de l'informatique

a. **L'informatique** est le domaine d'activité scientifique, technique et industriel en rapport avec le traitement automatique de l'information par des machines telles que les ordinateurs, les consoles de jeux, les robots, etc. Le terme « informatique » désigne à l'origine l'ensemble des activités liées à la conception et à l'emploi des ordinateurs, d'un point de vue principalement technique.

Remarques
1) L'objet de l'informatique est centré essentiellement sur le « le traitement automatique de l'information ».
2) Dans un traitement automatique, l'utilisation de la machine permet de réduire considérablement l'intervention humaine.

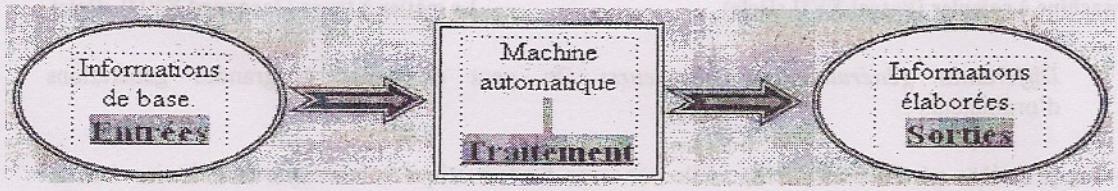


Figure 1. Schéma illustrant le traitement automatique de l'information

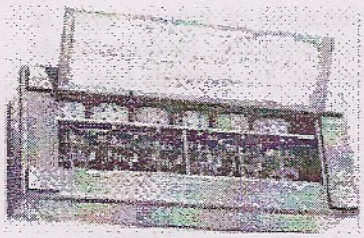
b). Les domaines d'application de l'informatique :

- Automatique : appareils de régulation tels le pilote automatique.
- Bio-informatique : outils d'aide dans la recherche en biologie.
- Bureautique : outils d'aide au travail de bureau : rédaction de documents commerciaux et correspondance.

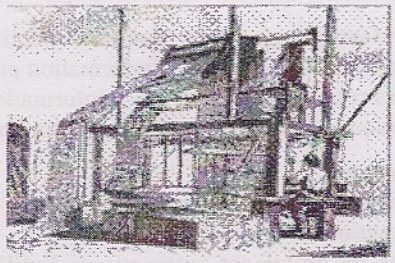
- Calcul parallèle : pour des applications qui demandent de nombreux calculs : prévisions météo ou image de synthèse.
- Cryptographie : déchiffrage d'informations chiffrées par un *code secret*.
- Domotique : commande d'appareils domestiques et systèmes d'alarme.
- Exploration de données : extraction automatique de connaissances.
- Gestion de contenu : collecte des documents électroniques d'une entreprise : mail, fax, contrats.
- Hypermédias : manipulation de documents de présentation contenant des vidéos, des images et du son.
- Imagerie informatique : création ou manipulation d'images : images de synthèse, traitement d'images, jeux vidéo, simulateurs de vol.
- Informatique décisionnelle : analyses et statistiques en vue d'aide à la décision pour les responsables d'entreprise.
- Informatique de gestion : manipulation en masse de grandes quantité d'informations : listes de clients, des fournisseurs, de produits.
- Informatique industrielle : utilisation dans des chaînes de fabrication industrielles.
- Informatique médicale : manipulations d'images médicales (scanner, échographies), dossiers médicaux.
- Informatique musicale : composition musicale.
- Instrumentation : collecte d'informations provenant de capteurs, lors d'expériences scientifiques.
- Linguistique informatique : correction d'orthographe, traduction automatique.
- *Malware* : logiciels mal intentionnés qui s'installent et agissent à l'insu de l'utilisateur : vol d'informations, falsification, usurpation d'identité.
- Nanotechnologie : aide à la recherche en nanotechnologie.
- Robotique : pilotage des machines autonomes que sont les robots.
- Télécommunications : transmission d'informations.

c. Histoire des ordinateurs

Au cours des siècles, des mathématiciens ont recherché des procédés d'automatisation des opérations et ont fabriqués «des machines à calculer». La plus célèbre est la machine de PASCAL (inventée en 1611). Parallèlement, des entrepreneurs et des inventeurs cherchent à automatiser des activités de production. C'est ainsi que Joseph-Marie JACQUARD, au début du XIX siècles, met au point la commande automatique du métier à tisser à l'aide de cartons perforés.



La machine à calculer (Pascal XVII siècle)



Le métier à tisser (Jacquard XVIII siècle)

- ***Différentes générations d'ordinateurs*:** On peut distinguer 4 grandes générations d'ordinateurs :

- La première génération (1950) : Elle concerne l'informatique scientifique et militaire dans le sens de résoudre les problèmes des calculs répétitifs. La technologie utilisée repose sur celle des tubes à vide (Tube et tore de ferrite). C'est une technologie lourde, ce qui a posé des problèmes de place et de consommation électrique. Les premiers ordinateurs commercialisés ont des tailles de celle d'un bâtiment. Seules les très grandes nations possèdent l'outil informatique.

- La deuxième génération (1960) : Cette génération d'ordinateurs concerne la naissance de l'informatique de gestion, elle coïncide avec l'apparition des transistors dans les unités centrales. L'utilisation de cette technologie améliore les vitesses de traitement et de débits des informations dans les ordinateurs. De plus, l'intégration de ces transistors permit la miniaturisation des différents composants des ordinateurs. La taille des ordinateurs est celle d'une armoire, ils consomment

2500W et traitent 200 000 instructions par seconde (IBM 704 et 1400). Les nations riches et les très grandes entreprises accèdent à l'outil informatique.

- La troisième génération (1970) : Le progrès observé dans cette nouvelle génération est le résultat de progrès spectaculaires au niveau des capacités des mémoires (stockage et vive) et au niveau de l'amélioration de la rapidité de traitement des données. Ces progrès sont eux même issus de ceux réalisés dans l'intégration des composants électroniques dans l'unité centrale de l'ordinateur, ainsi l'apparition des circuits intégrés (1964) et des microprocesseurs (1973) a donné naissance à la micro-informatique et à la personnalisation des ordinateurs. La taille des microprocesseurs est celle d'une carte, ils consomment 50 W et traitent 900 000 instructions par seconde. Les PME et PMI de tous les pays peuvent se procurer des matériels informatiques.

- La quatrième génération (milieu des années 70) : Cette génération est celle de l'apparition de la micro-informatique, (le micro-ordinateur est inventé par A.THI TRUONG en 1973) dans cette génération les ordinateurs ont bénéficié de l'intégration à grande échelle qui permettait d'intégrer des milliers de circuits sur une seule puce. On assiste depuis une dizaine d'années à une course à la puissance. En effet les constructeurs sortent tous les six mois de nouveaux composants toujours plus puissant (processeurs, carte graphiques,. ..) tout en cherchant à réduire les prix afin de se placer à la concurrence : ce qui ne déplait pas aux consommateurs, si ce n'est le fait que tous les six mois le matériel est dépassé techniquement... et qu'il faut sans cesse faire des remises à niveau (matériel et logiciel). De plus on cherche à rendre beau l'outil informatique qui prend une place de plus en plus importante dans la vie. Ceci se traduit par des innovations ergonomiques, de couleurs, et de réduction du volume des ordinateurs, et des nouveautés de toutes sortes (souris sans fil, écran plat,...). Qui pourrait dire à quoi dire à quoi ressemblerons les ordinateurs dans dix ans?!...

2. Composant d'un système informatique

Un système informatique est composé de deux parties: Composants matériels (Hardware) et Logiciel (Software).

- Matériels(Hardware) : Tout ce qui compose l'ordinateur et ses accessoires. Chaque composant possède une fonction bien particulière : calcul, stockage de des données, affichage vidéo, gestion du clavier…
- Logiciel (Software) : ensembles de programmes exécutables par l'ordinateur. On peut distinguer deux types :
  - Logiciels de base (Système d'exploitation): serrent d'interface entre le matériel et les logiciels d'application, Ex : Ms-DoS, Windows, Linux….
  - Logiciels d'application : est un ensemble de programmes informatiques qui servent à aider un utilisateur à faire un certain travail, Ex : word, excel…

3. Structure d'un ordinateur: L 'architecture de Von Neumann

L'architecture, dite architecture de Von Neumann, est le modèle utilisé pour les ordinateurs les plus courants (PC, téléphones ..) qui utilisent une structure de stockage unique pour conserver à la fois les instructions et les données requises ou générées par le calcul. De telles machines sont aussi connues sous le nom d'*ordinateurs à programme stocké en mémoire*. La séparation entre le stockage et le processeur est implicite dans ce modèle.

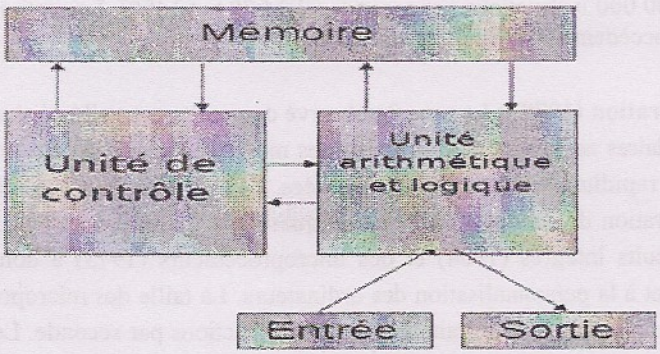


Figure2. Architecture de Von Neumann

Cette architecture est nommée d'après le mathématicien John von Neumann qui a soumis la première description d'un ordinateur dont le programme est stocké dans sa mémoire.

Elle décompose l'ordinateur en 4 parties distinctes :

1. L'unité arithmétique et logique (UAL ou ALU en anglais) ou unité de traitement : son rôle est d'effectuer les opérations de base ;
2. L'unité de contrôle (UC), chargée du séquençage des opérations ;
3. La mémoire qui contient à la fois les données et le programme qui dira à l'unité de contrôle quels calculs faire sur ces données. La mémoire est volatile (elle n'est pas persistante).
4. Les dispositifs d'entrée-sortie, qui permettent de communiquer avec le monde extérieur grâce aux périphériques.

3.1 Le processeur

Le processeur, ou CPU (de l'anglais *Central Processing Unit*, « Unité centrale de traitement »), est le composant de l'ordinateur qui exécute les programmes informatiques. Avec la mémoire notamment, c'est l'un des composants qui existent depuis les premiers ordinateurs et qui sont présents dans tous les ordinateurs. Un processeur construit en un seul circuit intégré est un microprocesseur.

L'invention du transistor en 1948 a ouvert la voie à la miniaturisation des composants électroniques. Les processeurs des débuts étaient conçus spécifiquement pour un ordinateur d'un type donné. Cette méthode coûteuse de conception des processeurs pour une application spécifique a conduit au développement de la production de masse de processeurs qui conviennent pour un ou plusieurs usages. Cette tendance à la standardisation qui débuta dans le domaine des ordinateurs centraux (*mainframes* à transistors discrets et mini-ordinateurs) a connu une accélération rapide avec l'avènement des circuits intégrés. Les circuits intégrés ont permis la miniaturisation des processeurs. La miniaturisation et la standardisation des processeurs ont conduit à leur diffusion dans la vie moderne bien au-delà des usages des machines programmables dédiées. Le processeur se divise en deux composants, l'UAL et l'UC.

i. Unité arithmétique et logique (UAL)

Les UAL peuvent être spécialisées ou pas. Les UAL élémentaires calculent sur des nombres entiers, et peuvent effectuer les opérations communes, que l'on peut séparer en quatre groupes :

1. Les opérations arithmétiques : addition, soustraction, changement de signe etc.,
2. les opérations logiques : compléments à un, à deux, ET, OU, OU-exclusif, NON, NON-ET etc.,
3. les comparaisons : test d'égalité, supérieur, inférieur, et leur équivalents « ou égal »,
4. éventuellement des décalages et rotations (mais parfois ces opérations sont externalisées).

Certaines UAL sont spécialisées dans la manipulation des nombres à virgule flottante, en simple ou double précision (on parle d'unité de calcul en virgule flottante ou *floating point unit* (FPU)) ou dans les calculs vectoriels. Typiquement, ces unités savent accomplir les opérations suivantes :

- additions, soustractions, changement de signe,
- multiplications, divisions,
- comparaisons,
- modulos

Certaines UAL, le plus souvent de la classe des FPUs, notamment celles des superordinateurs, sont susceptibles d'offrir des fonctions avancées :

- inverse (1/x),
- racine carrée,
- logarithmes,
- fonctions transcendantales (sin x, cos x, etc.),
- opération vectorielle (produit scalaire, vectoriel, etc.),
- etc.

Un processeur fait appel à plusieurs UAL, au moins deux : une située dans le chemin de contrôle pour incrémenter le registre de programme(de 1, 2, 4 ou 8 typiquement), et une autre dans le chemin de données pour traiter l'information.

ii. Unité de contrôle

Dans un système logique, en particulier dans un processeur, l'unité de contrôle (de commande) ou séquenceur commande et contrôle le fonctionnement du système, notamment du chemin de données. Une unité de contrôle est un circuit logique séquentiel qui réalise un automate fini, plus précisément une machine de Moore ou de Mealy, qui génère des signaux de contrôle pour piloter les éléments du chemin de données.

3.2 La mémoire vive

La mémoire vive, mémoire système ou mémoire volatile, aussi appelée RAM de l'anglais *Random Access Memory (que l'on traduit en français par 'mémoire à accès aléatoire')*, est la mémoire informatique dans laquelle un ordinateur place les données lors de leur traitement. Les caractéristiques de cette mémoire sont :

- sa rapidité d'accès (cette rapidité est essentielle pour fournir rapidement les données au processeur) ;
- sa volatilité (cette volatilité implique que les données sont perdues dès que l'ordinateur cesse d'être alimenté en électricité).

La mémoire vive (RAM) est généralement définie en opposition à la mémoire morte (ROM) : les données contenues dans la mémoire vive sont perdues lorsque l'alimentation électrique est coupée alors que la mémoire morte conserve ses données en absence d'alimentation électrique. La mémoire morte n'est donc pas volatile, ce qui la rend nécessaire lors du démarrage d'un ordinateur. En effet, la mémoire vive est dans un état indéterminé lors du démarrage.

Plus rarement, on utilise le sigle RWM (pour *Read Write Memory*, soit *mémoire en lecture écriture*) pour désigner la RAM en mettant l'accent sur la possibilité d'écriture plutôt que l'accès arbitraire.

Le sens littéral des termes *RAM* et *mémoire vive* peut prêter à confusion. En effet, le terme RAM implique la possibilité d'un accès arbitraire aux données, c'est-à-dire un accès à n'importe quelle donnée n'importe quand, par opposition à un accès séquentiel, comme l'accès à une bande magnétique, où les données sont nécessairement lues dans un ordre défini à l'avance. Or les ROM et les mémoires flash jouissent de la même caractéristique d'accès direct.

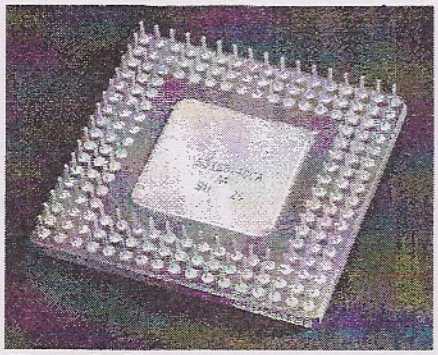


Figure 3. Un processeur (Pentium I)

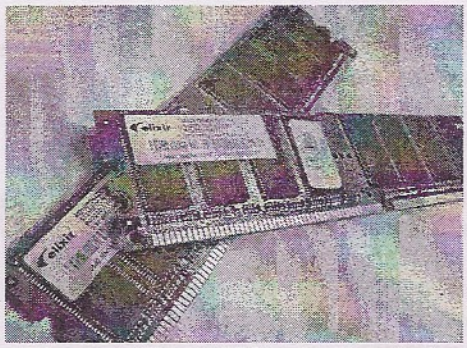


Figure 4. Une RAM (mémoire vive)

3.3 Le langage binaire

Le système binaire est un système de numération utilisant la base 2. On nomme couramment bit (de l'anglais *binary digit*, soit « chiffre binaire ») les chiffres de la numération binaire positionnelle. Ceux-ci ne peuvent prendre que deux valeurs, notées par convention 0 et 1.
C'est un concept essentiel de l'informatique. En effet, les processeurs des ordinateurs sont composés de transistors ne gérant chacun que deux états. Un calcul informatique n'est donc qu'une suite d'opérations sur des *paquets* de 0 et de 1, appelés octets lorsqu'ils sont regroupés par huit.

- **Transformation de base**

Pour pouvoir appliquer des traitements sur les données numériques de l'utilisateur (qui utilise du décimal), il est nécessaire d'effectuer une transformation de base vers le binaire (utilisé par le processeur). Une fois les traitements effectués, une transformation inverse est effectuée (vers le décimal)

- **Du binaire vers le décimal**

En considérant un nombre binaire quelconque de forme $b_n b_{n-1} b_{n-2} \ldots b_2 b_1 b_0$, il est possible grâce au polynôme des puissances successives de 2 d'obtenir l'équivalent décimale de ce nombre :

$$X_{10} = b_0 * 2^0 + b_1 * 2^1 + b_2 * 2^2 + \ldots + b_{n-1} * 2^{n-1} + b_n * 2^n$$

- **Du décimal vers le binaire**

Pour obtenir un nombre binaire à partir de son équivalent décimal, la méthode la plus simple est de recourir à des divisions successives sur 2 de ce nombre, puis de construire le nombre binaire grâce aux restes de ces divisions. Soit le nombre décimal X, il est possible de le décomposer comme suit :

$$X = 2 * X_n + r_n$$

C'est-à-dire que n'importe quel nombre décimal est divisible par 2 et le reste de la division est soit un soit zéro. Nous effectuons le même traitement sur X_n.

$$X_n = 2 * X_{n-1} + r_{n-1}$$

Nous recommençons l'opération jusqu'à ce que X_i soit nul :

$$X_{n-1} = 2 * X_{n-2} + r_{n-2}$$
$$X_{n-2} = 2 * X_{n-3} + r_{n-3}$$

$$X_1 = 2 * X_0 + r_0$$

X_0 est égale à zéro, le nombre binaire correspondant à X est :

$$X = r_0 r_1 r_2 \ldots r_{n-1} r_n$$

3.4 Le Disque Dur

Un disque dur est une mémoire de masse magnétique utilisée principalement dans les ordinateurs, mais également dans des baladeurs numériques, des caméscopes, des lecteurs/enregistreurs de DVD de salon, des consoles de jeux vidéo, des assistants numériques personnels et des téléphones mobiles.

a. Principe de fonctionnement

Dans un disque dur, on trouve des plateaux rigides en rotation. Chaque plateau est constitué d'un disque réalisé généralement en aluminium, qui a les avantages d'être léger, facilement usinable et paramagnétique. Des techniques plus récentes utilisent le verre ou la céramique, qui permettent des états de surface encore plus lisses que ceux de l'aluminium. Les faces de ces plateaux sont recouvertes d'une couche magnétique, sur laquelle sont stockées les données. Ces données sont écrites en code binaire [0,1] sur le disque grâce à une tête de lecture/écriture, petite antenne très proche du matériau magnétique. Suivant le courant électrique qui la traverse, cette tête modifie le champ magnétique local pour écrire soit un 1, soit un 0, à la surface du disque. Pour lire, le même matériel est utilisé, mais dans l'autre sens : le mouvement du champ magnétique local engendre aux bornes de la tête un potentiel électrique qui dépend de la valeur précédemment écrite, on peut ainsi lire un 1 ou un 0.

Un disque dur typique contient un axe central autour duquel les plateaux tournent à une vitesse de rotation constante. Toutes les têtes de lecture/écriture sont reliées à une armature qui se déplace à la surface des plateaux, avec une à deux têtes par plateau (une tête par face utilisée). L'armature déplace les têtes radialement à travers les plateaux pendant qu'ils tournent, permettant ainsi d'accéder à la totalité de leur surface.

b. Géométrie

Chaque plateau (possédant le plus souvent 2 surfaces utilisables) est composé de pistes concentriques séparées les unes des autres par une zone appelée « espace interpiste ». Les pistes situées à une même distance de l'axe de rotation forment un cylindre. La piste est divisée en secteurs (aussi appelés blocs) contenant les données.

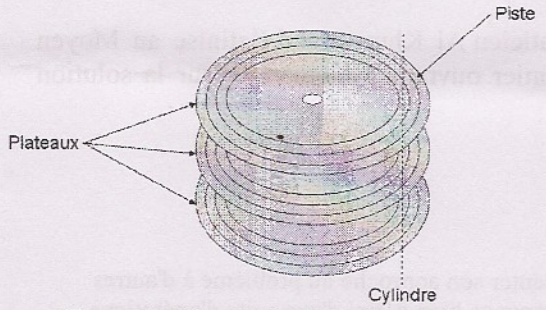


Figure 5. Géométrie d'un disque dur

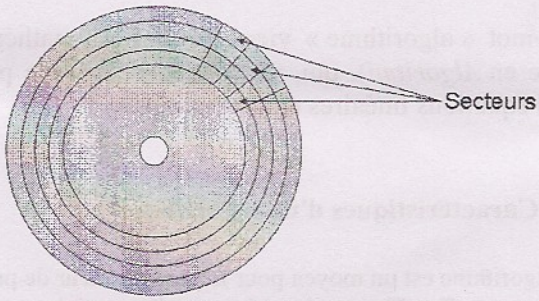


Figure 6. Géométrie d'une surface

En adressage CHS, il faut donc trois coordonnées pour accéder à un bloc (ou secteur) de disque :

1. le numéro de la tête de lecture (choix de la surface) ;
2. le numéro de la piste (détermine la position du bras portant l'ensemble des têtes) ;
3. le numéro du bloc (ou secteur) sur cette piste (détermine à partir de quel endroit il faut commencer à lire les données).

c. Capacité

La capacité d'un disque dur peut être calculée ainsi : **capacité = nombre de cylindres × nombre de têtes × nombre de secteurs par piste × nombre d'octets par secteur (généralement 512).**

d. Performances

Le temps d'accès et le débit d'un disque dur permettent d'en mesurer les performances. Les facteurs principaux à prendre en compte sont :

1. **le temps d'accès**, désigne la durée entre le moment ou le processeur demande un fichier et le moment où l'interface du disque dur fournit ce fichier.
2. **le taux de transfert** est la quantité de données transférée entre le disque dur et l'ordinateur par le biais de son interface en une seule seconde.

e. Différences entre le Disque Dur et la RAM

Disque Dur	Mémoire Vive (RAM)
• Mémoire Secondaire	• Mémoire Primaire
• Grand temps d'accès	• Petit temps d'accès
• Grande Capacité	• Petite capacité
• Mémoire de stockage	• Mémoire de travail

II. L'algorithmique

1. Définitions et généralités

L'algorithmique est l'ensemble des règles et des techniques qui sont impliquées dans la définition et la conception d'algorithmes, c'est-à-dire de processus systématiques de résolution, par le calcul, d'un problème permettant de décrire les étapes vers le résultat. En d'autres termes, un algorithme est une suite finie et non-ambiguë d'opérations permettant de donner la réponse à un problème.

Si les opérations d'un algorithme s'exécutent les unes après les autres, c'est un algorithme séquentiel, si elles s'exécutent en parallèle, c'est un algorithme parallèle. Si l'algorithme exploite des tâches s'exécutant sur un réseau de processeurs on parle d'algorithme réparti, ou distribué.

Le mot « algorithme » vient du nom du mathématicien Al Khuwarizmi (latinisé au Moyen Âge en *Algoritmi*), qui, au IXe siècle écrivit le premier ouvrage systématique sur la solution des équations linéaires et quadratiques.

2. Caractéristiques d'un algorithme

L'algorithme est un moyen pour le programmeur de présenter son approche du problème à d'autres personnes. En effet, un algorithme est l'énoncé dans un langage bien défini d'une suite d'opérations permettant de répondre au problème. Un algorithme doit donc être :

- Lisible: l'algorithme doit être compréhensible même par un non-informaticien
- De haut niveau: l'algorithme doit pouvoir être traduit en n'importe quel langage de programmation, il ne doit donc pas faire appel à des notions techniques relatives à un programme particulier ou bien à un système d'exploitation donné
- Précis: chaque élément de l'algorithme ne doit pas porter à confusion, il est donc important de lever toute ambiguïté
- Concis: un algorithme ne doit pas dépasser une page. Si c'est le cas, il faut décomposer le problème en plusieurs sous-problèmes
- Structuré: un algorithme doit être composé de différentes parties facilement identifiables d'un

3. Formalisme d'un algorithme

Un algorithme est d défini par :
— son nom
— déclaration de variable
— des instructions
— un ordre sur les instructions

Il a la forme suivante : **Algorithme** NomDeLAlgo
Déclaration de variable

Début
Séquence d'instruction
Production d'un résultat
Fin

3.1 Les variables

Les s variables correspondent à des boites, qui nous serviront à stocker nos données.

A quoi servent les variables ?

Dans un Algo on stocke des résultats ;

- Données pour l'utilisateur (décrivant le problème)
- Calculs pendant l'exécution

Sur ces données on différencie le type ;

Une variable est une boite avec ; un nom (=label) ET une forme (un type)
Un nom = > éviter des adresses mémoires directement.
Un type = > parce que la zone réservée pour chaque type est différente.

Déclaration de variables

Première chose à faire : création de la "boite" (la variable) en lui donnant un nom et une forme.
Ceci se fait au début de l'algo avant la suite d'instruction.
Remarque : sur les noms de variables est une suite de lettre et de chiffre.
On évite d'utiliser les signes de ponctuation.
Le premier caractère d'un nom de variables doit être une lettre. Les noms doivent être explicites.

Types classiques

a) types numériques

Les types numériques classiques :
Type intervalle , 1 octet [0, 255] ,entier simple [-32 768, + 32 767], entier long (libre à vous de les rechercher mais au final, on s'en fout),réel simple ,réel long.....

Type Numérique	Plage
Byte (octet)	0 à 255
Entier simple	-32 768 à 32 767
Entier long	-2 147 483 648 à 2 147 483 647
Réel simple	$-3{,}40\times10^{38}$ à $-1{,}40\times10^{45}$ pour les valeurs négatives $1{,}40\times10^{-45}$ à $3{,}40\times10^{38}$ pour les valeurs positives
Réel double	$1{,}79\times10^{308}$ à $-4{,}94\times10^{-324}$ pour les valeurs négatives $4{,}94\times10^{-324}$ à $1{,}79\times10^{308}$ pour les valeurs positives

La déclaration se fait par 2 propriétés sur les variables : le nom et le type.

b) le type alphanumérique

Ce type va représenter des chaines de caractères ou des caractères.

c) type booléen

Un booléen code VRAI ou FAUX (TRUE ou FALSE, 0 ou 1)

3.2 Instruction d'affectation

On utilise "← " pour l'affectation affecter une valeur à une variable.
EX : Toto ← 24 : Attribue 24 à la valeur Toto

- On peut également affecter une variable à une variable ;

EX ; A← toto : la valeur de toto est affectée à A (A = 24)

- On peut affecter le résultat d'une opération :

A ← toto + 4 => A = 28, toto = 2

a. Ordres des instructions

L'ordre des instructions joue un rôle prépondérant dans la production de résultat.
Exemple : Soit la variable A en Entier

Début
A ← 34
A ← 12
fin
Que vaut A ? ……A = 12
Remarque : En Algo, une variable reçoit une est une seule valeur affectée à un moment.

3.3 Expressions et opérations

L'opération ' ← ' est un lien entre un nom de variable et un contenu
sur la gauche : le nom de la variable
sur la droite : Une expression
Définition: c'est un ensemble de valeurs reliées par des opérateurs et dont la production est un unique objet.
Remarque: le type de l'expression (le type de l'objet produit par l'expression) doit être le même que celui de variable sur laquelle on procède à l'affectation.

a) Opérateurs numériques

Les opérateurs arithmétiques classiques :
+ Addition
- Soustraction
* Multiplication
/ Division

On ajoute :
^ Puissance
( ) Parenthèses

b) Opération alphanumérique

On définit l'opérateur de concaténation sur les chaines de caractère : &
" Il fait beau " & "aujourd'hui "
=> "Il fait beau aujourd'hui"
La chaine qui ne contient rien ""
"aujourd'hui" & "" → un peu inutile toutefois...
=> "aujourd'hui"

3.4 Lecture et écriture

Les notions *input* et *output*. Input → on demande à l'utilisateur d'entrer une variable. Output → fait sortir une donnée.
En algo, on peut au choix, faire sortir et entré des variables en fin et début, ou faire sortir ou entré des infos en milieu de programme. Les effets seront différent.

Entrée d'information

2 possibilités :
- en début de programmation (arguments)
- en cours d'execution

Argument

Définition : *Un argument est une donnée dont un algorithme a besoin.* Il peut être une constante
Exemple: 1 carte dans un algo d'itinéraire ou une donnée modifié par l'algo.
Les arguments sont placés en début tel que :
Algorithme nomdelalgo
(argument_1, argument_2, …), On les introduit

Primitives d'algorithmique

Qu'est ce qu'un test ?

Lors de l'évaluation d'un Algorithme, 2 suites d'instructions peuvent entrer en concurrence. Dans ce cas, il faut un critère qui permet de choisir l'une ou l'autre de ces suites.

Notion du Si

Si à la première à gauche il y a un sens interdit
Alors on prend la deuxième à gauche
Sinon on continue tout droit