

Université A. Mira Béjaia  
Faculté de Sciences Exactes  
Département Mathématiques  
L1 Math (tronc commun)  
2024/2025



## **Module : Algorithmique et Structures de Données 1 (ASD1)**

**Semestre : 01**

**Unité d'enseignement : Fondamentale**

**Crédits : 6**

**Coefficient : 4**

**Mode d'évaluation : Examen (60%) + Contrôle Continue (40%)**

### **Objectif :**

L'objectif principal de ce cours est d'apprendre à analyser un problème, en le décrivant en termes algorithmiques, à choisir les structures de données adéquates et à maîtriser correctement les difficultés inhérentes à la programmation.

A l'issue de ce cours, les étudiants de la première année (math) doivent être capables : de modéliser et d'analyser un problème, de programmer une solution et de la valider.

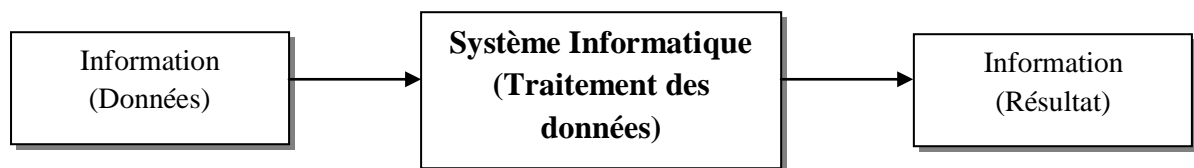
## Chapitre 1 : Introduction à l'algorithmique

### 1. Introduction

Le mot informatique a été proposé par Philippe Dreyfus en 1962 ; Le mot « informatique » est une contraction de deux mots « **information** » et « **automatique** ». L'informatique c'est donc une automatisation de l'information, plus exactement un traitement automatique de l'information. L'information désigne ici tout ce qui peut être traité par l'ordinateur (textes, images, sons, vidéos,...).

L'outil utilisé pour traiter l'information de manière automatique s'appelle un « ordinateur ». Ce nom a été proposé par Jacques Perret (professeur de Latin à La Sorbonne) en 1954. Ce mot était à l'origine un adjectif qui signifiait "qui met de l'ordre", "qui arrange".

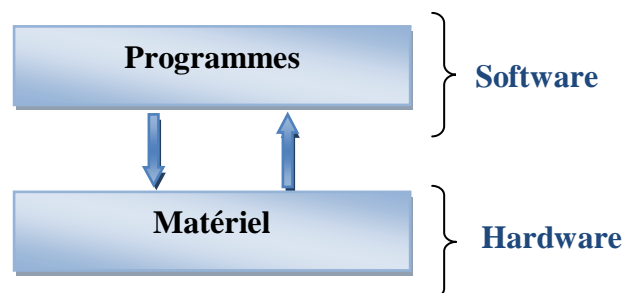
L'anglais, plus restrictif, utilise le terme de « computer » qui peut se traduire par calculateur, machine à calculer.



**Figure 1 :** Traitement automatique de l'information

### 2. Le système informatique

Un système informatique est un ensemble de composants matériel et logiciel. On parle de « **Hardware** » pour désigner l'ensemble des éléments matériels et de « **software** » pour désigner la partie logicielle.



**Figure 2 :** Système Informatique

#### 2.1. Hardware

##### 2.1.1. L'architecture de Von Neumann

*Jon Von Neumann*, un mathématicien hongrois, est le premier à mettre en œuvre la notion de « Mémoire » pour enregistrer un programme en utilisant le système binaire basé sur deux valeurs (0 et 1). La machine de Von Neumann était composée en trois parties.

Cette architecture est à la base de la conception des ordinateurs jusqu'à nos jours.

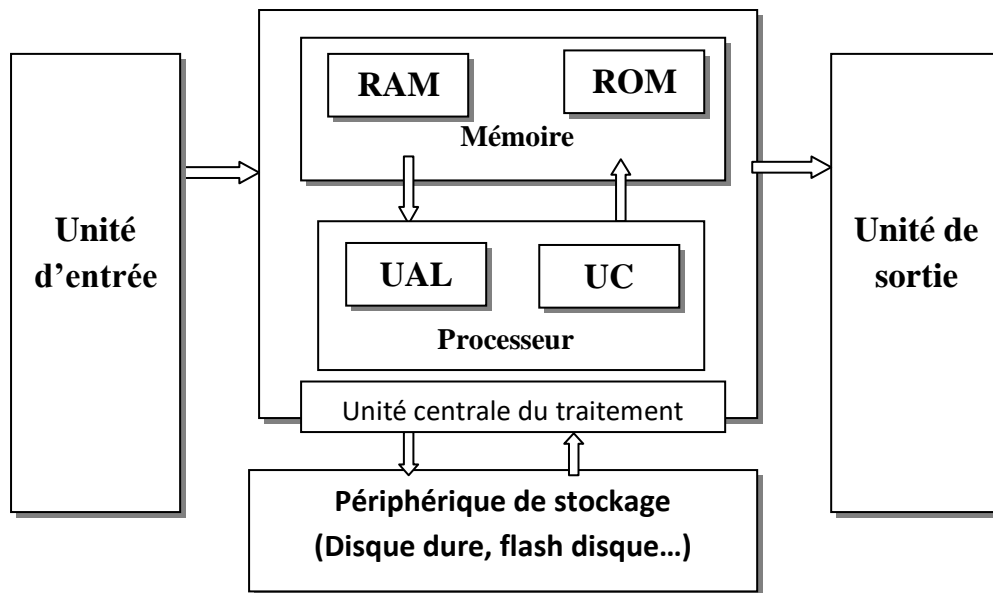


Figure 3: Architecture Von Neumann

1) **Unité centrale du traitement** est divisée en deux parties, l'unité de contrôle et de commande (UCC) et l'unité arithmétique et logique (UAL).

- **Une unité de contrôle et commande:** responsable de la lecture en mémoire principale et du décodage des instructions
- **Unité arithmétique et logique** ou **unité de traitement** : son rôle est d'effectuer les opérations de base mathématiques ou logiques.

2) **La mémoire** : qui contient à la fois les données et le programme exécuté par l'unité de commande.

Chaque case élémentaire capable de mémoriser 0 ou 1 est appelée un bit (BInary digiT). La capacité d'une mémoire peut se mesurer en nombre d'octets disponibles tels que : 1 octet (o) = 8 bits ; 1 kilo-octet (ko) = 1024 octets ; 1 Méga-octet (Mo) = 1024 ko ; 1 Géga-octet (Go) = 1024 Mo ; 1 Téra-octet (To) = 1024 Go.

Elle comporte deux types de mémoires RAM, ROM

- **RAM ou mémoire vive** est une mémoire primaire d'un ordinateur, rapidement accessible, dans laquelle les données peuvent être lues, écrites ou effacées. Elle est faite de circuit intégré où les informations peuvent être modifiées. Les informations sont perdues à la mise hors tension.
- **ROM ou la mémoire morte:** (Read Only Memory, dont la traduction est mémoire en lecture seule) est appelée aussi parfois mémoire non volatile, car elle ne s'efface pas lors de la mise hors tension du système.

3) **L'interface d'entrée-sortie** : qui permet de communiquer avec le monde extérieur. Elle comporte deux composants : L'unité d'entrée (UE) et l'unité de sortie (US)

- **Les périphériques d'entrée (Unités entrées)** : Ils permettent de véhiculer les informations du monde extérieur vers la mémoire de l'ordinateur.
  - **Exp** : le clavier, la souris, le scanner, le microphone, lecteur de codes-barres, crayon à lecture optique.

- **Les périphériques de sorties** : permettent de véhiculer les informations de la mémoire de l'ordinateur vers le monde extérieur.
  - **Exp** : l'écran, l'imprimante, les enceintes audio.
- ❖ **Les périphériques de stockage** : permettent de sauvegarder les informations.  
**Exp** : Flash disque, disque dur, SSD Disque, lecteur de CD-ROM ou DVD gravés.

## 2.2. Le Software

On appelle « Software » la partie logique de l'ordinateur, elle est constituée de tous les programmes qui permettent de faire fonctionner la machine ainsi que ceux permettant à l'utilisateur d'exploiter l'ordinateur selon ses besoins.

*Qu'est ce qu'un programme ?*

- **Un programme** est une suite d'instructions, écrit dans un langage que l'ordinateur comprend. Un programme a besoin de données pour s'exécuter ; pour réaliser une tâche, le processeur fait l'appel au programme qui se trouve dans la mémoire et exécute ces instructions les unes après les autres.

L'ensemble de ces programmes peut être reparti en deux ensembles :

- ✚ Les programmes de base
- ✚ Les programmes d'application

### 2.2.1. Les programmes de base

Les programmes de base sont des programmes conçus pour pouvoir gérer les éléments physiques de l'ordinateur. Les programmes de base visent à éloigner l'utilisateur de la machine physique. Ils constituent ce qu'on l'appelle actuellement « le Système d'exploitation ».

**Exemples :**

**MS-DOS** (MicroSoft Disc Operating System), **Windows** (successeur de MS-DOS), **Mac-OS** (Fonctionne sur les micro-ordinateurs de type macintosh), **Unix, Lunix** (nouveau système d'exploitation qui vient de remplacer Unix).

### 2.2.2. Les programmes d'application

Pour ces besoins, l'utilisateur utilise des programmes particuliers. Des programmes d'application dans différents domaines de l'informatique sont créés sans cesse pour permettre à l'utilisateur d'exploiter l'ordinateur afin de se faciliter la vie.

**Exemples :**

On ne peut pas donner ici tous les logiciels d'application qui existent, mais on peut donner une classification qui est loin d'être exhaustive :

- **Les logiciels de bureautique**: logiciels de traitement de texte, tabulation (**Exp** : la série Microsoft Office : Word, Excel...)
- **Les logiciels de programmation**: les langages de programmation permettent à l'utilisateur d'écrire ses propres programmes. (**Exp** : Pascal, Delphi, JAVA, C, C++, ...)
- **Les logiciels de gestion et de calcul (progiciel)** : ils regroupent les logiciels conçus pour gérer un domaine de gestion particulier (comptabilité, stock...)
- **Les logiciels de messagerie et de l'Internet** sont des logiciels capables de faciliter la communication via le réseau ou l'Internet, (**Exp** : les navigateurs Web, les applications des réseaux sociaux ...)

- **Les logiciels de simulation** de l'intelligence artificielle et de contrôle industriel : sont des logiciels très sophistiqués créés pour réaliser des tâches bien spéciales. (dans les laboratoires, des usines, station nucléaires, aéroports ...).

### 3. Notion d'un Algorithme / un programme

#### 3.1. Algorithme

Un *algorithme* est une suite finie et non ambiguë d'opérations ou *d'instructions* permettant de résoudre un problème. C'est une méthode, pour arriver dans un temps fini, à un *résultat* déterminé à partir d'une situation *donnée*.

#### 3.2. Programme

Un programme informatique est une séquence d'instructions écrites dans un langage compréhensible par ordinateur (langage de programmation), en d'autre terme c'est la traduction de l'algorithme dans un langage de programmation.

#### 3.3. Langage de programmation

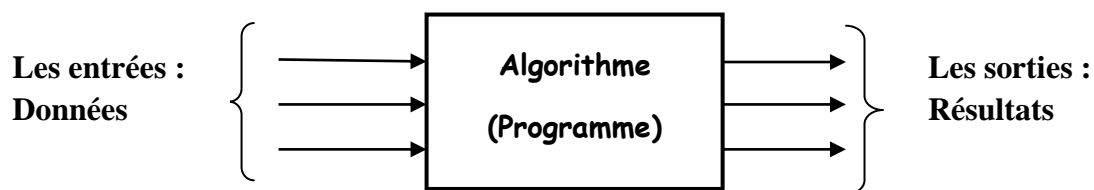
Ensemble de commandes et mots clés nécessaires pour l'écriture d'un programme afin qu'il soit compréhensible par l'ordinateur

**Exemples:** Pascal, C, C++, Delphi, JAVA, Python....

#### 3.4. Langage algorithmique

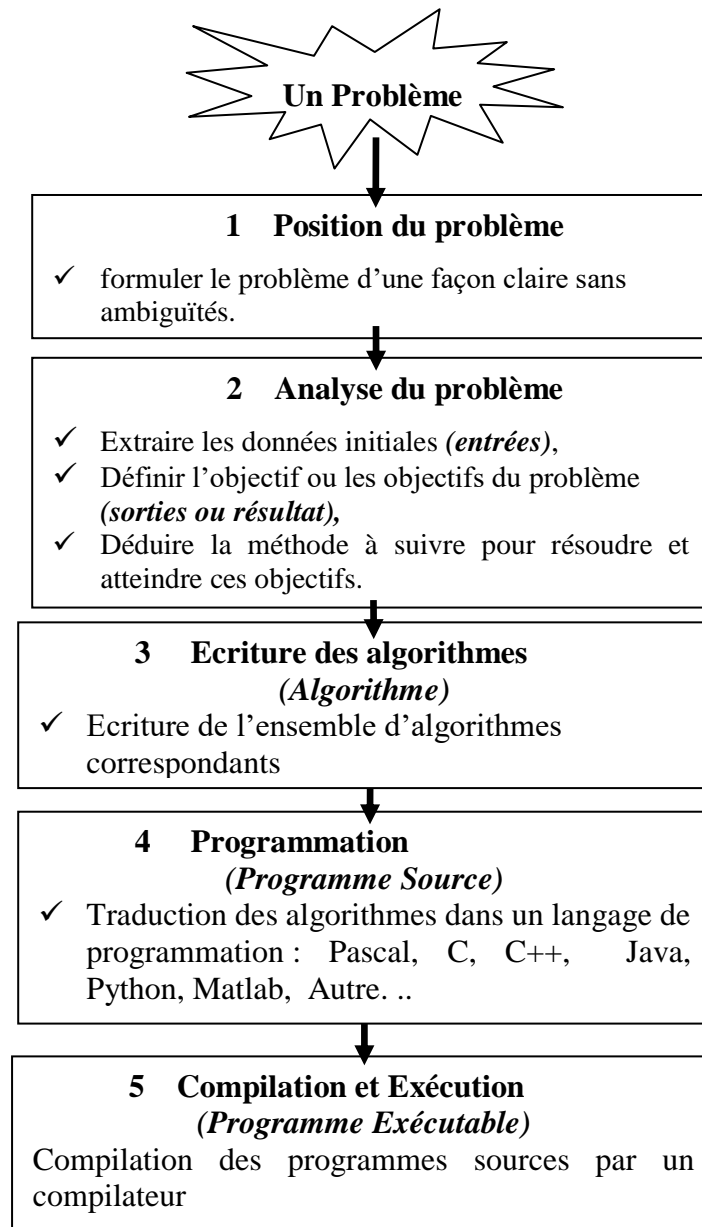
Le "**langage algorithmique**" est un pseudo langage qui tient compte des caractéristiques de la machine, tout en étant plus souple qu'un langage de programmation. C'est donc un compromis entre un langage naturel et un langage de programmation caractérisé par une liste de mots clés. Un mot clé est un identificateur qui a une signification particulière (debut, si, Fin....).

- Un **programme** est (donc) la traduction d'un algorithme dans un langage accepté par la machine (langage de programmation).
- Le traitement de l'information par l'ordinateur consiste à faire élaborer, par cette machine, des informations appelées *résultats*, à partir d'informations appelées *données*.



#### 4. Etapes de résolution d'un problème

La réalisation d'un programme exécutable par un ordinateur, nécessite le suivi d'une démarche constituée d'un ensemble d'étapes.



##### 4.1. Première étape : Position du problème

Le problème est souvent posé par un demandeur de solution informatique. Parfois, ces demandeurs ne savent plus exprimer leurs besoins avec précision. L'objectif de cette étape est bien formuler le problème pour pouvoir le résoudre correctement. Cet élève rencontre le problème suivant ?



Comment je peux trouver  
la surface d'un rectangle

**Q :** Aider cet élève à exprimer son problème ?

**R :** Le problème est : « Ecrire un programme qui permet de lire la longueur et la largeur d'un rectangle et afficher sa surface ».

#### 4.2. Deuxième étape : Spécification et analyse des problèmes

L'objectif de cette étape est bien comprendre l'énoncé du problème, déterminer les formules de calculs, les règles de gestion,...etc.

L'analyse des problèmes s'intéresse aux éléments suivants :

- Les données nécessaires aux traitements (entrées),
- Les résultats souhaités (sorties),
- Les traitements (actions réalisés pour atteindre le résultat).

##### Analyse du problème précédent:

- Données d'entrées:  
Connaitre la valeur de **la longueur L**.  
Connaitre la valeur de **la largeur G**.
- Résultats ou Sorties : Objectifs : Calcul la surface S
- Méthode : Selon la règle : La surface = longueur \* largeur    (**S = L \* G**)

#### 4.3. Troisième étape : Ecriture de l'algorithme

Après avoir terminé l'analyse, il faut mettre les instructions dans leur ordre logique d'exécution. On obtient **un algorithme**. « Un algorithme est une suite structurée et finie d'actions ou d'instructions permettant de résoudre un problème ».

##### L'algorithme relatif au calcul de la surface :

```
Algorithme calcul_surface
Variables L ,G ,S : real ;
Début
  Lire (L);
  Lire (G);
  S ← L * G ;
  Ecrire ('la surface est',S);
Fin
```

Un algorithme utilise les conventions suivantes :  
Le verbe « Lire » utilisé pour la saisie des données.  
Le verbe « Ecrire » utilisé pour l'affichage du résultat.  
Le signe « ← » utilisé pour mettre une valeur dans une variable.

#### 4.4. Quatrième étape : Ecriture du programme

Dans cette étape on va traduire l'algorithme dans un langage de programmation pour obtenir un programme, ou « **code source** » du Programme qui est stocké dans un ou plusieurs **fichiers** sur l'ordinateur. Pour la traduction, on utilise le langage de programmation Pascal et le langage C pour obtenir deux programmes.

**Programme 1 en Pascale**

```

Program calcule_surface;
  Uses Wincrt ;
  Var L ,G ,S : reel ;
  Begin
    Read (L);
    Read (G);
    S := L*G ;
    Writeln ('La surface est',S);
  End.
    
```

**Programme 2 en Langage C**

```

int main( )
{
  float L, G, S ;
  scanf("%f", &L) ;
  scanf("%f", &G) ;
  S = L*G ;
  printf("La surface est = %f\n", S) ;
  return 0 ;
}
    
```


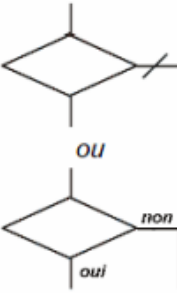


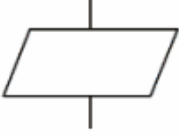


**4.5. Cinquième étape : Compiler et exécuter le programme : (voir TP)**

Dans cette étape le code source du programme est traduit *en langage machine* ( suite de 0 et 1) en utilisant un Logiciel spéciale appelé *compilateurs*, on obtient alors *le code machine du Programme*, ce code machine est ensuite exécuté directement par l'ordinateur.

**5. Les organigrammes**

Un organigramme est la représentation graphique d'un algorithme. Ils utilisent des symboles et des flèches pour illustrer la séquence des étapes impliquées dans une tâche. Voici les éléments clés des organigrammes :

**Les symboles normalisés :**

SYMBOLE	DÉSIGNATION	SYMBOLE	DÉSIGNATION
	début ou fin d'un algorithme		<b>Test ou Branchement conditionnel</b>  décision d'un choix parmi d'autres en fonction des conditions
	symbole général de « traitement » opération sur des données, instructions, ... ou opération pour laquelle il n'existe aucun symbole normalisé		<b>sous-programme</b>  appel d'un sous-programme
	entrée / sortie		<b>Liaison</b> Les différents symboles sont reliés entre eux par des lignes de liaison. Le cheminement va de haut en bas et de gauche à droite. Un cheminement différent est indiqué à l'aide d'une flèche
	commentaire		



**Exemple :**

Etablir un organigramme qui lit deux nombres entiers A et B, puis les comparer.

### L'analyse :

**Les entrées :** deux nombres entiers A et B

**Les sorties :** le nombre le plus grand (soit A soit B)

### Les cas possible :

1.  $A > B$  afficher le nombre A
2.  $A < B$  afficher le nombre B
3.  $A = B$  les deux nombre sont égaux

### L'organigramme :

