



# Initiation à la programmation

Mohsine Eleuldj

Département Génie Informatique, EMI

[eleuldj@emi.ac.ma](mailto:eleuldj@emi.ac.ma)

# Initiation à la programmation



## Objectif

- Introduction aux algorithmes et aux notions de base de la programmation
- Programmation en Java sous Eclipse

## Contenu

- Introduction
- Conception des algorithmes
- Environnement de développement Java
- Programmation orientée objet Java
- Bibliothèques des classes Java

## Evaluation des connaissances :

- Comptes rendus (10%) + TP (20%) + Contrôle (70%)

# Chapitre I : Introduction



- 1 Terminologie
- 2 Structure d'un ordinateur
- 3 Langages de programmation
- 4 Systèmes d'exploitation
- 5 Exécution d'un programme

# Terminologie

---

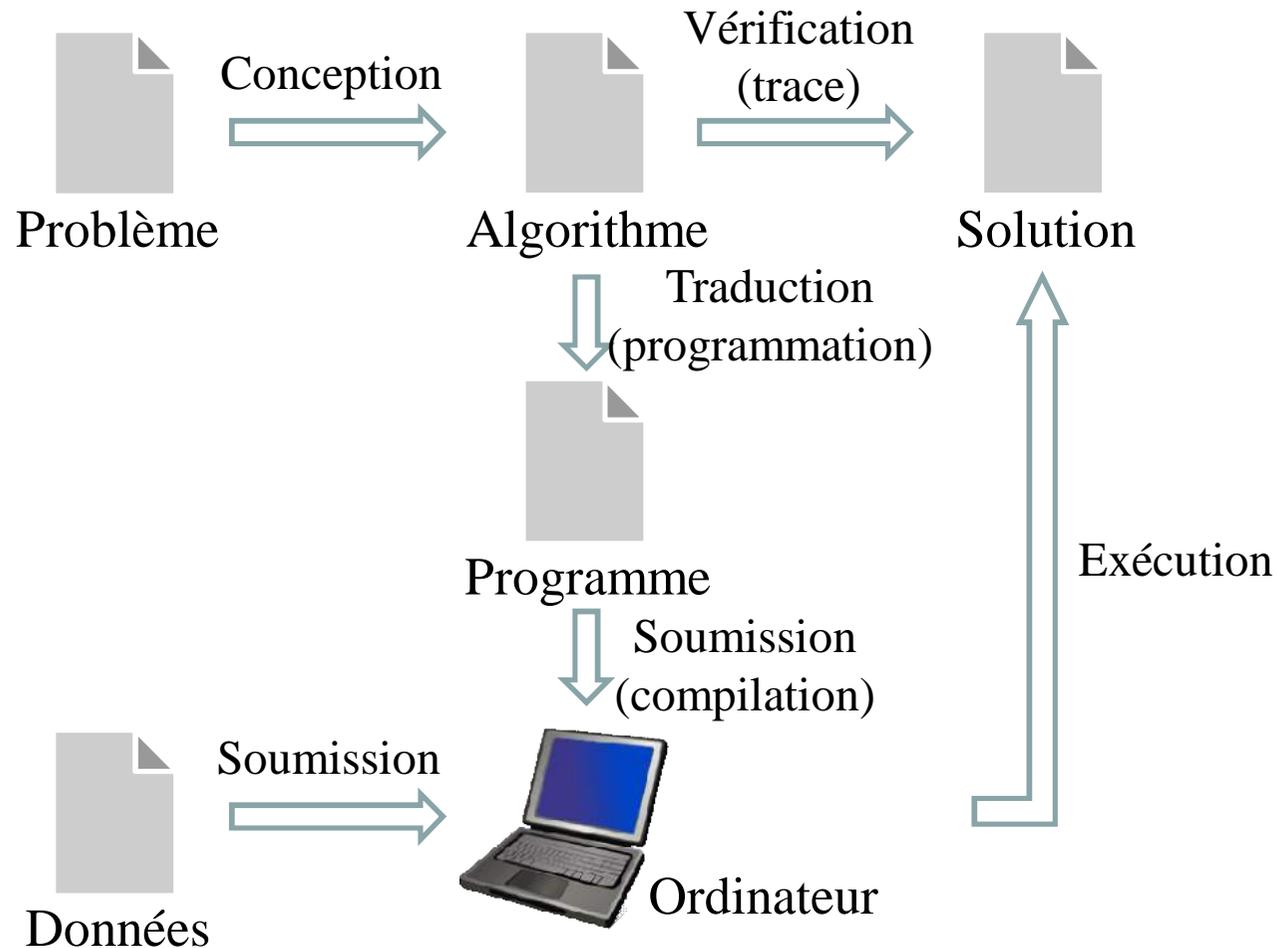
**Algorithme** : suite de prescriptions précises qui indiquent l'exécution, dans ordre bien déterminé, d'une succession d'opérations en vue de la résolution d'un problème.

**Programme** : ensemble d'ordres ou instructions qui agit sur les données pour produire des résultats.

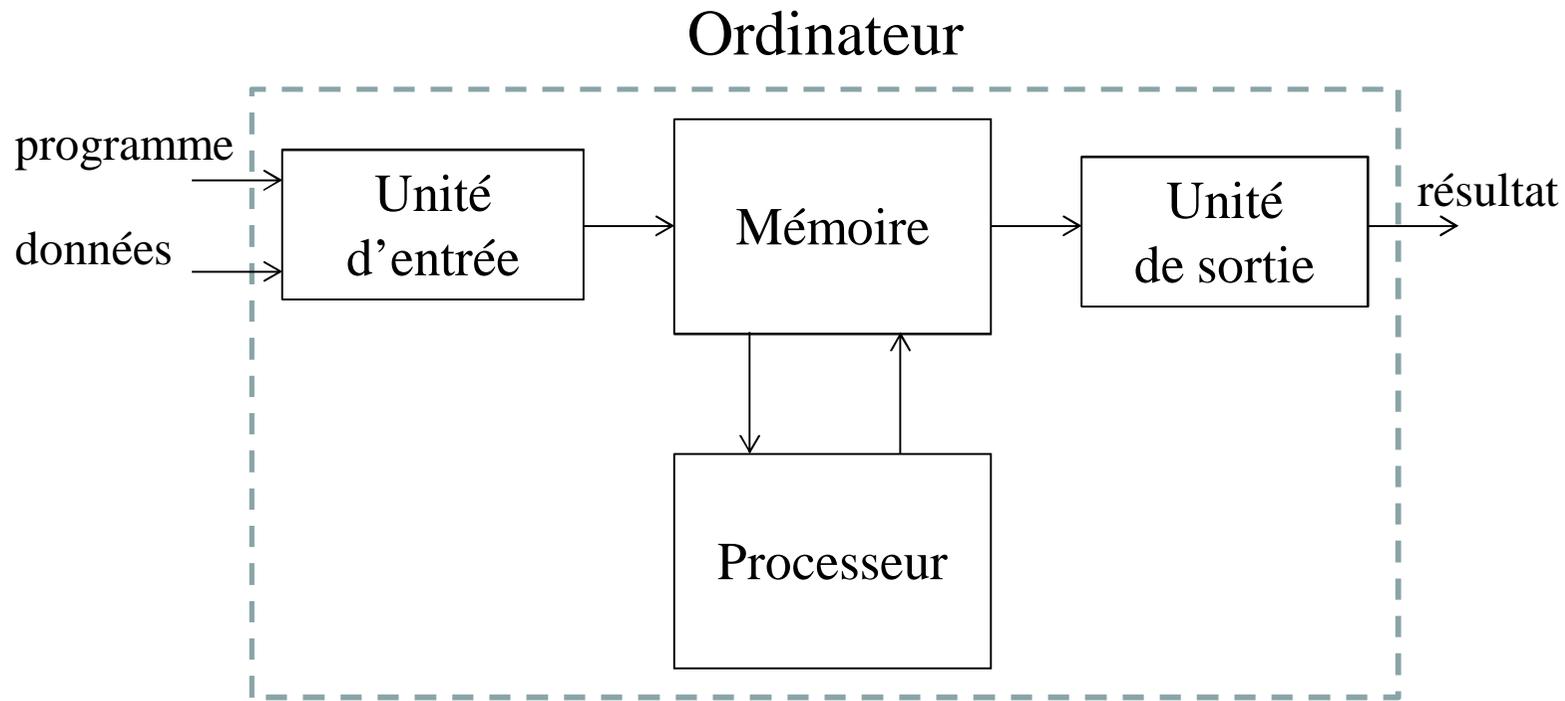
**Ordinateur** : machine électronique capable d'exécuter un programme.

**Informatique** : traitement automatique de l'information à l'aide d'un ordinateur.

# Méthodologie de développement de programmes



# Structure d'un ordinateur

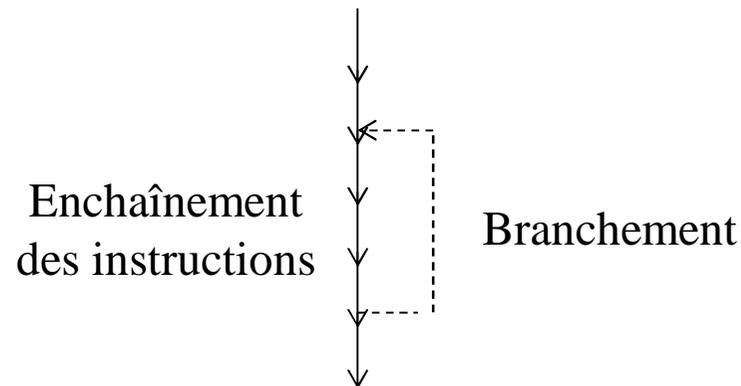


Automacité :

faculté d'exécuter des millions d'instructions sans l'intervention humaine

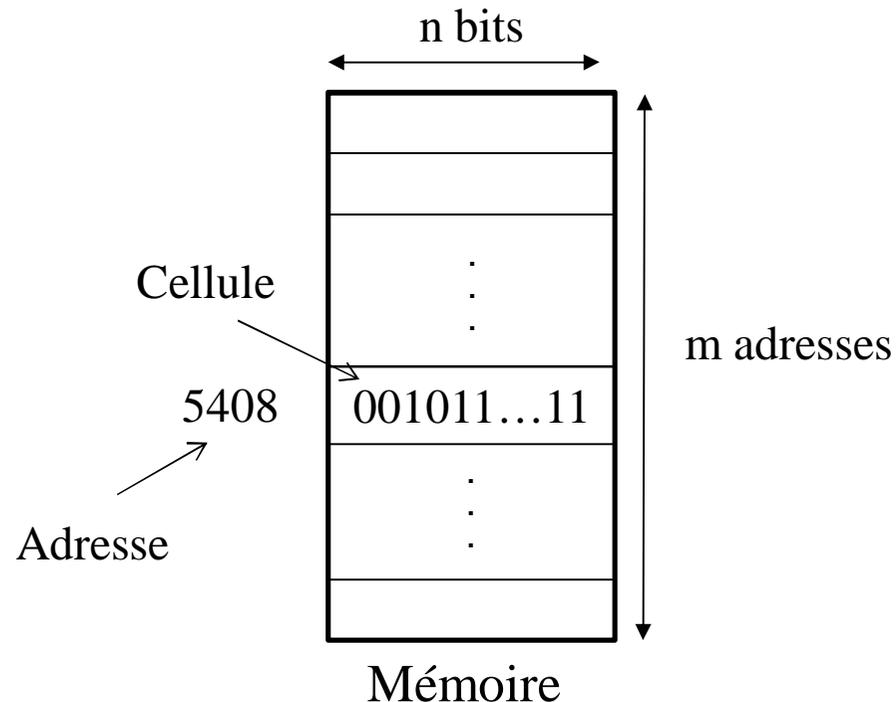
# Types d'instructions

- Instruction arithmétique : addition, soustraction, multiplication,...
- Instruction de transfert : mouvement de l'information d'un emplacement à un autre (entre le processeur et la mémoire)
- Instruction de branchement : rupture de l'exécution séquentielle des instructions



- Instruction d'entrée/sortie : lecture ou écriture (caractère, nombre,...)
- Commande de gestion des périphériques (imprimante, disque,...)

# Mémoire principale

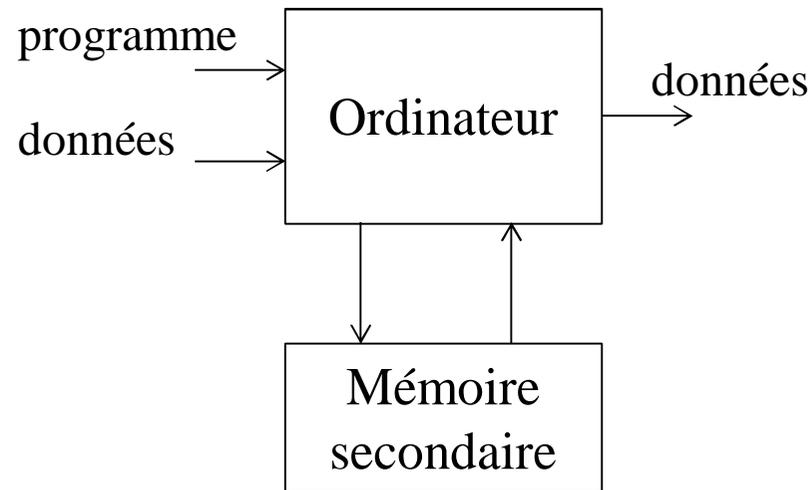


Mémoire : ensemble de cellules ayant chacune une adresse unique et contenant des bits

Rôle : emmagasiner les informations + pouvoir les accéder (lecture et écriture)

Caractéristiques : volatile + capacité limitée (quelques Giga octets) + coût élevé

# Mémoire secondaire



Support : magnétique ou optique (biologique dans le stade de la recherche)

Temps d'accès : dépend de la localisation de l'information dans le support

Caractéristiques : permanente + capacité (quelques Téra octets) + coût bas

# Configuration de mon système

---

Processeur : Intel i7-2760QM

Mémoire cache 6 Mo

Disque dur : 698 Go

Lecteur DVD/CD-ROM

Carte réseau

claviers PS/2

Bluetooth

contrôleur audio vidéo,

lecteur carte à puce

...

Fréquence : 2,40 Ghz

Mémoire RAM : 4 Go

4 Cœurs et 8 Threads

Ports COM et LPT

carte graphique

contrôleur de bus USB

Souris

Moniteur

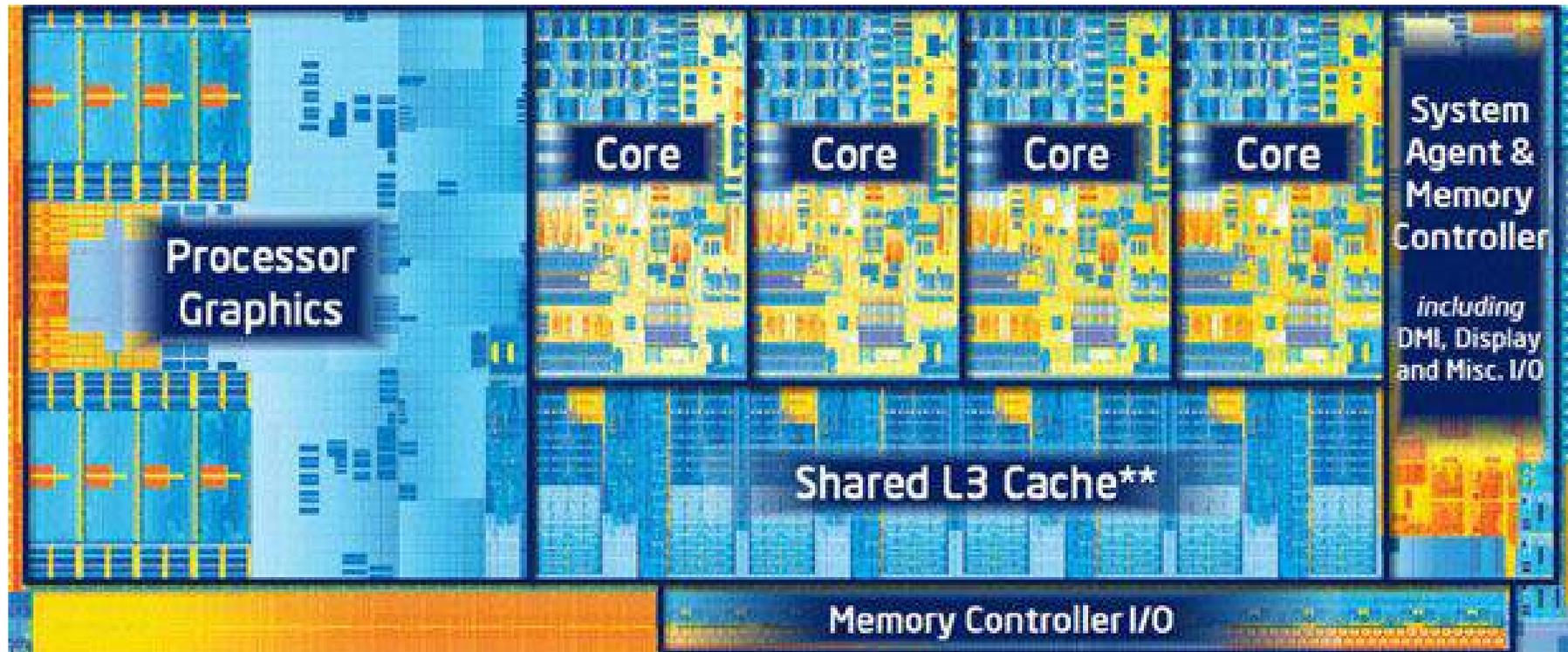
Exercice : Déterminer la configuration de votre système

# Configuration de mon système

Series [Intel Core i7](#) Codename Sandy Bridge Clock Rate 2400 - 3500 MHz Level 1 Cache 256 KB Level 2 Cache 1024 KB Level 3 Cache 6144 KB Number of Cores / Threads 4 / 8 Max. Power Consumption (TDP = Thermal Design Power) 45 Watt Transistor Count 995 Million Manufacturing Technology 32 nm Die Size 216 mm<sup>2</sup> Max. Temperature 100 °C Socket FCBGA1224, FCPGA988 Features HD Graphics 3000, DDR-1333 Memory Controller, HyperThreading, AVX, Quick Sync, Virtualization GPU [Intel HD Graphics 3000](#) (650 - 1300 MHz) 64 Bit 64 Bit support Hardware Virtualization VT-x, VT-d Starting Price \$378 U.S. Announcement Date 09/03/2011 ...

<http://www.notebookcheck.net/Intel-Core-i7-2760QM-Notebook-Processor.59920.0.html>

# Intel Core i7-3770K



Caractéristiques : fréquence = 3,4 GHz, technologie de fabrication = 22 nm et nombre de transistors = 1,4.

# Langages de programmation

Définition : Outil à l'aide duquel le programmeur écrit des programmes exécutables sur un ordinateur

Exemples : FORTRAN, COBOL, Pascal, Ada, C, Java,...

```
.....  
101101101111  
111011110101  
01100011011  
.....
```

langage binaire  
(langage machine)

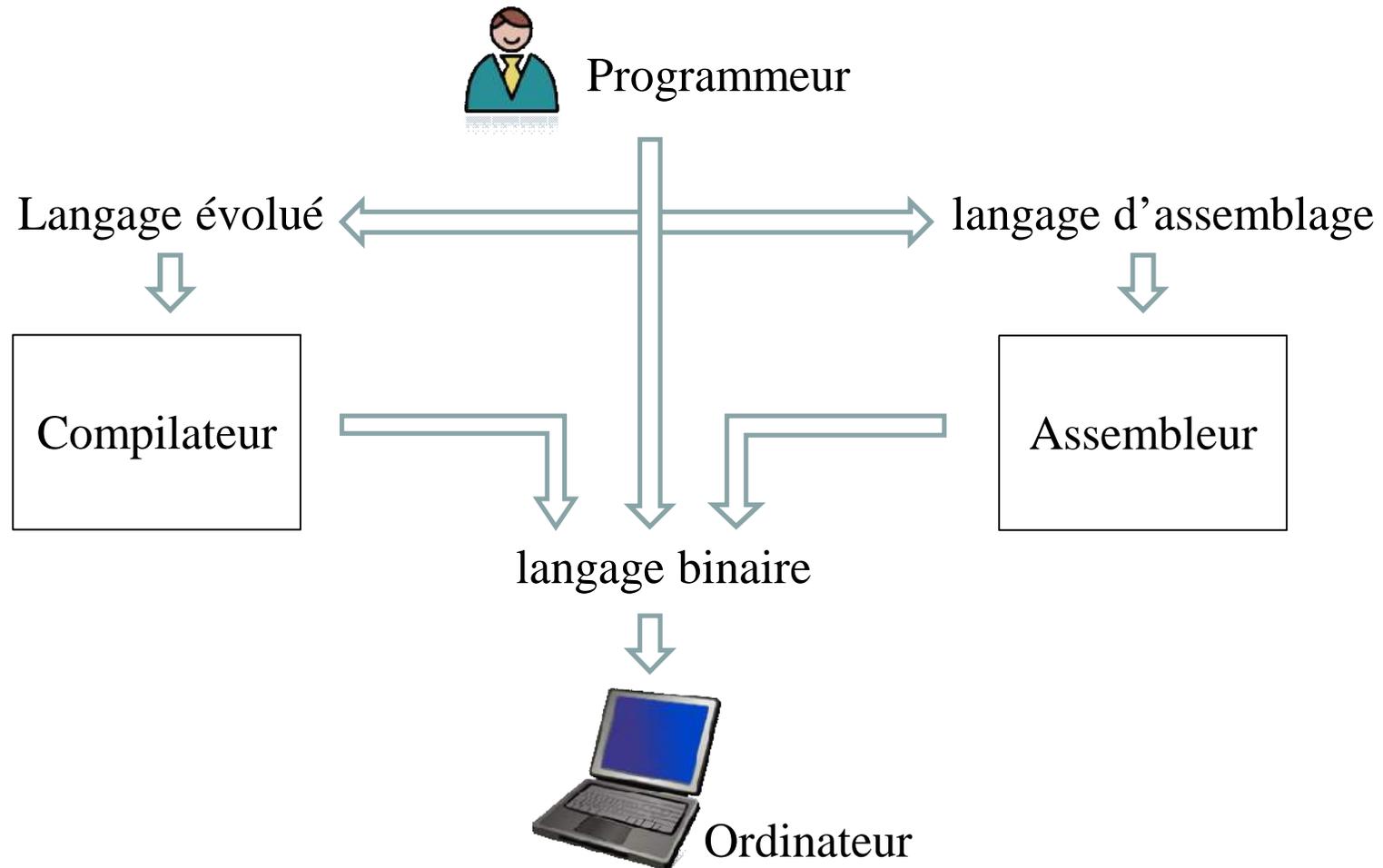
```
.....  
ADD R1,R2  
MULT R4,13  
.....
```

Langage symbolique  
(langage d'assemblage)

```
.....  
Indice = i+3  
System.out  
.....
```

Langage évolué  
(haut niveau)

# Choix du langage de programmation



# Systeme d'exploitation

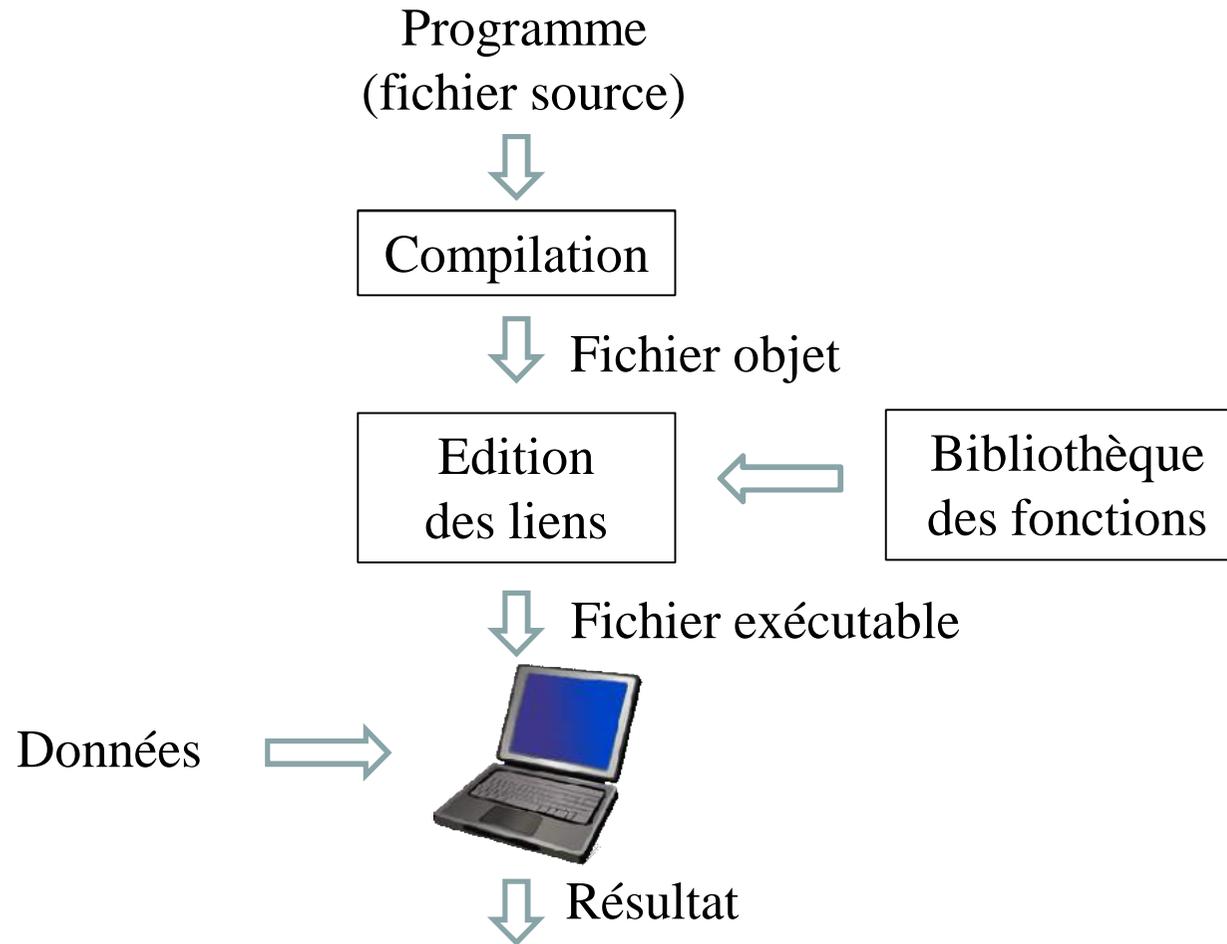
Définition : ensemble de programmes qui permettent de gérer l'utilisation (exploitation) de la machine (Operating System)

Composantes : compilateurs, assembleur, fonctions d'entrée/sortie, mémoire virtuelle, multiprogrammation...

Rôle : faciliter au programmeur l'utilisation de la machine et ses périphériques et assurer le bon fonctionnement et l'utilisation optimale.

Exemples : Unix, Linux, Windows, SunOS, ...

# Exécution d'un programme



# Chapitre II : Conception des algorithmes



1 Notion d'algorithme

2 Structures de contrôle

Enchaînement

Sélection binaire

Sélection multiple

3 Structures répétitives

Boucle tant que

Boucle répéter ... jusqu'à

Boucle pour

4 Structure de données tableau

5 Etapes de résolution d'un problème

# Notion d'algorithme

Origine d'après les historiens des mathématiques en 825 du livre

جعفر محمد ابن موسى الخوارزمي في كتابه "الجبر و المقابلة"

Définition : Un algorithme est une suite de prescriptions précises qui indiquent d'exécuter, dans un ordre déterminé, une succession d'opérations en vue de la résolution d'un problème.

Exemples

Résolution d'une équation du second degré

Guide d'utilisation d'un appareil ménager

Recette de cuisine sans interprétation ni interprétation

Pseudo-code (ou langage algorithmique) : langage informel qui vise à représenter un algorithme indépendamment de l'agent (humain ou machine) qui sera appelé à l'exécuter (intermédiaire entre un langage naturel et un langage de programmation)

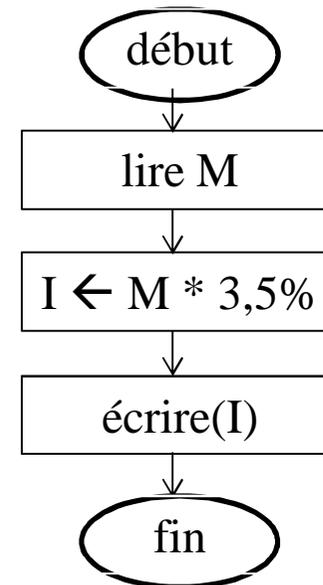
Organigramme : diagramme, qui est une autre représentation d'un algorithme, composé de rectangles, losanges, flèches, ...

# Enchainement

Problème 1 : Calculer l'intérêt  $I$  d'un montant  $M$  sachant que le taux d'intérêt est de 3,5%

*1 début*  
*2 lire M*  
*3  $I \leftarrow M * 3,5\%$*   
*4 écrire (I)*  
*5 fin*

Pseudo\_code



Organigramme

# Sélection binaire

Problème 2 : Calculer l'intérêt  $I$  d'un montant  $M$  sachant que le taux d'intérêt est de 3,5% si  $M < 10\ 000$  DH et de 4% sinon.

1 *début*

2 *lire M*

3 *si  $M < 10\ 000$  alors*

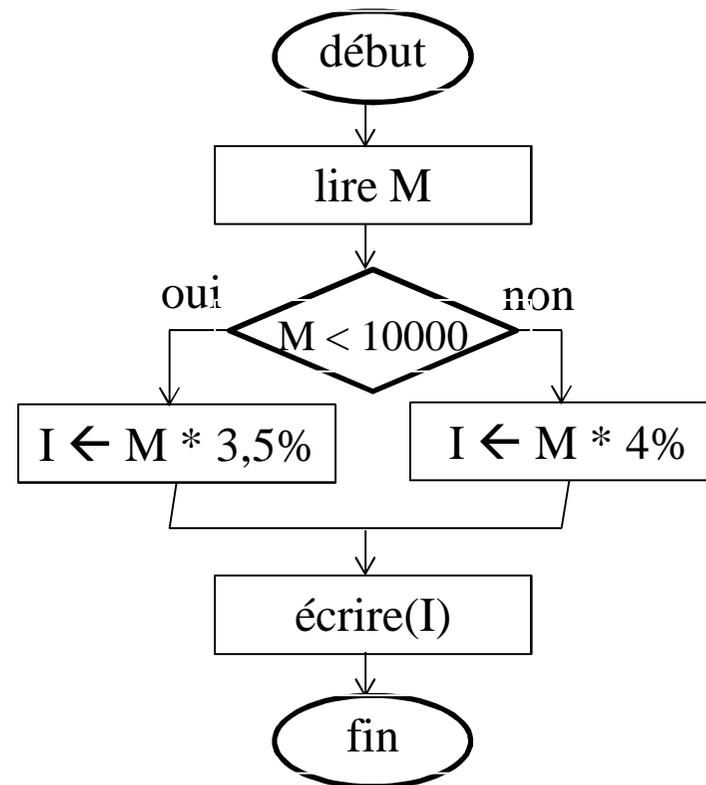
4  $I \leftarrow M * 3,5\%$

5 *sinon*

6  $I \leftarrow M * 4\%$

7 *écrire(I)*

8 *fin*



# Sélections binaires imbriquées

Problème 3 : Calculer l'intérêt  $I$  d'un montant  $M$  sachant que le taux d'intérêt annuel dépend de la période de dépôt  $P$ . Il est de 3,5% si  $P = 1$  an, de 4% si  $P = 2$  ans, de 4,5% si  $P = 3$  ans et de 5% si  $P \geq 4$  ans.

*lire  $M, P$*

*si  $P = 1$  alors*

*$I \leftarrow M * 3,5\%$*

*sinon*

*si  $P = 2$  alors*

*$I \leftarrow M * 4\%$*

*sinon*

*si  $P = 3$  alors*

*$I \leftarrow M * 4,5\%$*

*sinon*

*$I \leftarrow M * 5\%$*

*écrire( $I$ )*

*lire  $M, P$*

*si  $P = 1$  alors*

*$I \leftarrow M * 3,5\%$*

*sinon si  $P = 2$  alors*

*$I \leftarrow M * 4\%$*

*sinon si  $P = 3$  alors*

*$I \leftarrow M * 4,5\%$*

*sinon*

*$I \leftarrow M * 5\%$*

*écrire ( $I$ )*

# Sélection multiple

Problème 4 : Calculer l'intérêt  $I$  d'un montant  $M$  sachant que le taux annuel est de 3,5% si  $P=1$  an, de 4% si  $P=2$  ans, de 4,5% si  $P=3$  ans et de 5% si  $P=4$  ans.

*lire  $M, P$*

*choisir  $P$  de*

$$1 : I \leftarrow M * 3,5\%$$

$$2 : I \leftarrow M * 4\%$$

$$3 : I \leftarrow M * 4,5\%$$

$$4 : I \leftarrow M * 5\%$$

*écrire( $I$ )*

Exercice :

- Dessiner l'organigramme de l'algorithme du problème 3.
- Quelle est la différence entre les algorithmes du problème 3 et 4 ?
- Qu'arrive-t-il si  $P$  est négatif ou supérieur à 4 ?

# Boucle tant que

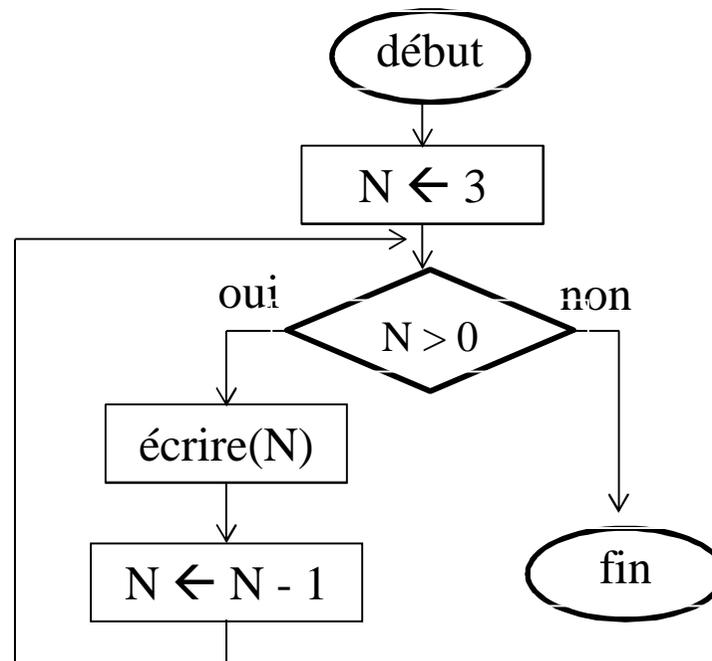
Forme générale

**tant que** <condition> **faire**  
<traitement>

Problème 5 : imprimer les valeurs  $N$  ( $1 \leq N \leq 3$ ) en ordre décroissant.

1 *début*  
2  $N \leftarrow 3$   
3 ***tant que***  $N > 0$  ***faire***  
4 *écrire*( $N$ )  
5  $N \leftarrow N - 1$   
6 *fin*

Exercice : Faire la trace.



# Trace de l'algorithme 5

Opération	N	$N > 0$	Ecrire(N)
1	-	-	-
2	3	-	-
3	3	oui	-
4	3	-	3
5	2	-	-
3	2	oui	-
4	2	-	2
5	1	-	-
3	1	oui	-
4	1	-	1
5	0	-	-
3	0	non	-
6	-	-	-

# Exercice 1



Problème 6 : On dispose d'une liste de nombres positifs tel que le dernier nombre est 0. Calculer la moyenne arithmétique de ces nombres.

- a) Ecrire un algorithme en pseudo-code basé sur la boucle tant que.
- b) Dessiner l'organigramme de l'algorithme.
- c) Faire la trace de cet algorithme pour : 14,5 12 15 13 0

# Boucle répéter ... jusqu'à

Forme générale

**répéter**

<traitement>

**jusqu'à** <condition>

Reprenons le problème 5 en utilisant la boucle  
répéter...jusqu'à.

1 *début*

2  $N \leftarrow 3$

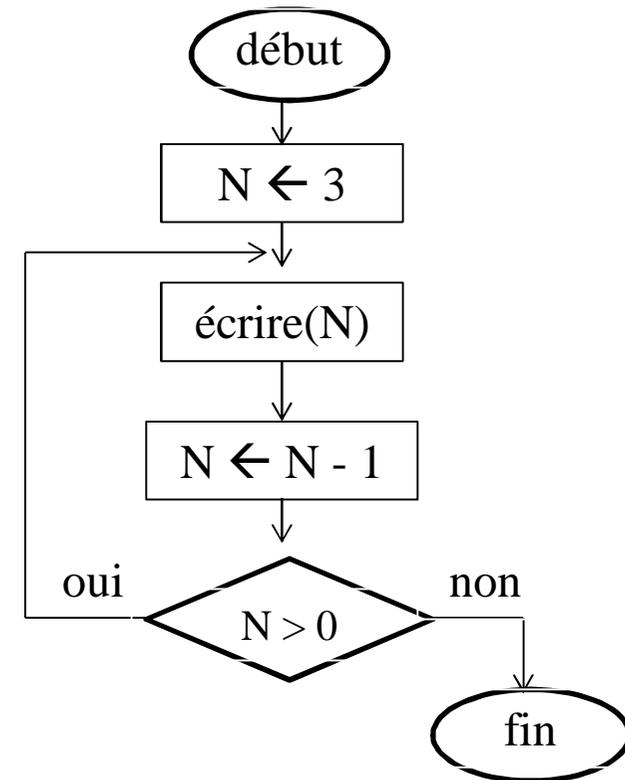
3 *répéter*

4 *écrire(N)*

5  $N \leftarrow N - 1$

6 *jusqu'à*  $N=0$

7 *fin*



# Exercice 2



Reprenons le problème 6 qui consiste à calculer la moyenne arithmétique d'une liste de nombres positifs dont le dernier élément est 0.

- a) Ecrire l'algorithme en pseudo-code basé sur la boucle répéter...jusqu'à.
- b) Dessiner l'organigramme de l'algorithme.
- c) Faire la trace de ce algorithme pour la liste : 14 12 15 13 0
- d) Faire la trace de ce algorithme pour la liste : 0

# Boucle pour

Forme générale

**pour** v=valeur1 **à** valeur2 **faire**  
<traitement>

Reprenons le problème 5 en utilisant la boucle pour.

*1 début*

*2 **pour** N = 3 **à** 1 **faire***

*3 écrire(N)*

*4 fin*

Exercice : Comparer les trois types de boucles (structures répétitives).

# Exercice 3

Reprenons le problème 6 qui consiste à calculer la moyenne arithmétique d'une liste de nombres positifs sachant que la liste est composée de  $N$  éléments.

- a) Ecrire l'algorithme en pseudo-code basé sur la boucle pour.
- b) Faire la trace de ce algorithme pour : 14 12 15 13
- c) Comparer les trois algorithmes du problème 6.

Problème 7 : Considérons le calcul du quotient  $Q$  et du reste  $R$  de la division de deux nombres positifs  $A$  et  $B$  en utilisant l'addition et la soustraction uniquement (la division entière ne doit pas être utilisée).

- a) Décrire l'algorithme.
- b) Faire la trace pour  $A = 9$  et  $B = 4$
- c) Que se passe-t-il si :  $A = 0$  et  $B \neq 0$ ;  $A > 0$  et  $B = 0$ ;  $A = 0$  et  $B = 0$ ;  $A < B$

# Exercice 4

Problème 8 : Trouver toutes les valeurs de  $N$  ( $1 \leq N \leq 100$ ) tel que :

$$1 + 2 + \dots + N = 3(N - 1)$$

Problème 9 : Déterminer le plus grand élément dans une suite de nombres positifs

Problème 10 : Soit l'algorithme suivant:

*lire*  $A, \varepsilon$

$X \leftarrow A$

***répéter***

$X \leftarrow (X + A/X)/2$

***jusqu'à***  $|X^2 - A| < \varepsilon$

*écrire*  $X$

- Faire la trace pour  $A = 9$  et  $\varepsilon = 0,1$ .
- Que calcule cet algorithme ?

# Structure de donnée tableau

Définition : un tableau est une structure de données qui peut contenir plusieurs données de même type. Un élément du tableau est accédé en utilisant la variable du tableau suivie du numéro de l'élément appelé indice

Exemple : Soit T un tableau de 10 entiers

$T(5) \leftarrow 26$  permet d'affecter la valeur 26 au 5<sup>ème</sup> élément.

Problème 11 : Considérons que le tableau Notes contient N notes. Calculer la moyenne arithmétique de ces notes.

- a) Ecrire l'algorithme en pseudo-code.
- b) Faire la trace de ce algorithme pour Notes=(14 ,12,15,13).

Problème 12 : calculer le produit de deux matrices d'ordre n.

- a) Ecrire l'algorithme en pseudo-code.
- b) Faire la trace de cet algorithme pour deux matrices d'ordre 2.

# Etapes de résolution d'un problème

## 1 Définition précise du problème

généralement les problèmes sont mal définis et représentent des ambiguïtés.  
Il faudrait prévoir les cas susceptibles de se produire et préciser le traitement correspondant (données, traitement, résultats et erreurs)

## 2 Méthode de développement

procédurale  
orientée objet

## 3 Conception des algorithmes

## 4 Validation des algorithmes

les algorithmes doivent répondre aux spécifications (trace)

## 5 Codification

expression (traduction) des algorithmes en langage de programmation  
dépendamment des langages disponibles et de la nature du problème.

## 6 Mise au point

correction des erreurs de syntaxe et de sémantique (à l'aide des jeux d'essai).

# Solutions

Algorithme 8

$S \leftarrow 0$

***pour**  $N$  à 100 **faire***

$S \leftarrow S + N$

***si**  $S = 3(N - 1)$  **alors***

*écrire( $N$ , vérifie l'équation)*

***sinon***

*écrire( $N$ , ne vérifie pas l'équation)*

Algorithme 9

*lire  $A$  et  $B$*

$Q \leftarrow 0$

***tant que**  $A \geq B$  **faire***

$A \leftarrow A + B$

$Q \leftarrow Q + 1$

$R \leftarrow A$