

Cours 1:

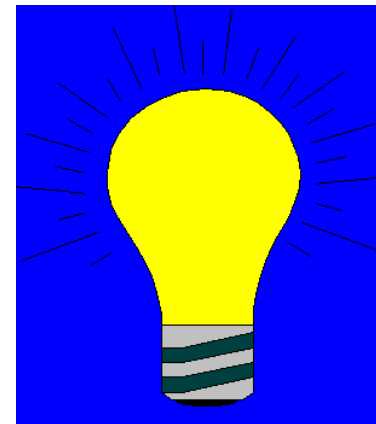
L'ingénierie, la résolution de problèmes, les ordinateurs et la programmation

Prof. Mohamed Ibrahim, PhD
University d'Ottawa

Qu'est-ce que l'ingénierie?

- Juridique:
 - Une profession autorégulatrice: ing.
 - Statut similaire à celui du droit et de la médecine
- Pratique:
 - Résolution de problèmes

Solutions



Aspects couverts

- Technologie

Pouvons-nous le faire?

- Économie

Peut-on faire au juste prix?

- Société

Pouvons-nous le faire sans danger?

- Communications

Comment pouvons-nous expliquer?

Disciplines de l'ingénierie

Génie chimique

- Des techniques principalement issues de la **chimie** mais aussi de la physique, de la biologie et des mathématiques.
- Les procédés continus pour faire des produits comme l'essence, les plastiques, les médicaments (biotechnologie), la nanotechnologie, les piles à combustible, etc.
 - Conversion des matières premières en produits plus intéressants.
- Garantir que les produits sont sans danger.

Génie civil

- Offrir à la société les **infrastructures**:
 - Bâtiments
 - Ponts
 - Villes
 - Autoroutes
 - Barrages
- Utilise des procédés chimiques pour fournir de l'eau propre, par exemple.
- Utilise les processus physiques pour construire des bâtiments et des ponts, par exemple .

Génie informatique

- Application de principes d'ingénierie pour construire des ordinateurs ou des systèmes informatiques et des dispositifs.
- Systèmes individuels aux grandes, de l'iPad aux superordinateurs.
- Co-design de matériel/logiciel et optimisation de performance.
- Traite aussi les réseaux informatiques (Internet), les communications informatiques, etc.

Génie électrique

- Utilise de l'électricité, de l'électronique et de l'électromagnétisme pour construire divers **systèmes et appareils**.
- Utilise également des procédés physiques et chimiques pour produire et utiliser l'électricité.
- **Électrique:**
 - Transmission, les moteurs et les machines
 - Sources d'énergie réutilisables et les technologies vertes
- **Electronique:**
 - Appareils audiovisuels, des amplificateurs, des systèmes de communication, les ordinateurs, etc.

Génie mécanique

- Utilise des processus scientifiques essentiellement physiques et matérielles (aussi une certaine chimie) pour **construire des machines, des matériaux et des structures**.
- Automobile, aéronautique, locomotives et navires.
- Moteurs, les systèmes énergétiques.
- Fabrication, la robotique, le contrôle.

Génie logiciel

- Discipline nouvelle d'ingénierie.
 - Très important en raison de demande énorme!
 - Souvent classé numéro 1 en termes de meilleures carrières en Amérique du Nord.
- Utilise des principes d'ingénierie pour **construire des systèmes et produits logiciels**
- De très petits logiciels (votre machine distributrice Coca-Cola) à de très grands systèmes (comme Amazon cloud).
- Le logiciel est intégré dans tous les systèmes modernes, même les voitures.
- L'application de principes d'ingénierie pour assurer la sécurité publique.

Résolution de problèmes

Résolution de problèmes

- Définir le problème.
- Collecter de l'information.
- Générer une solution possible.
- Mettre en œuvre la solution.
- Tester la solution.

Définir le problème

- La définition appropriée du problème est essentielle pour une bonne solution.
- Pas toujours facile de voir le vrai problème.
- Pour mieux comprendre le problème, il faut le diviser.

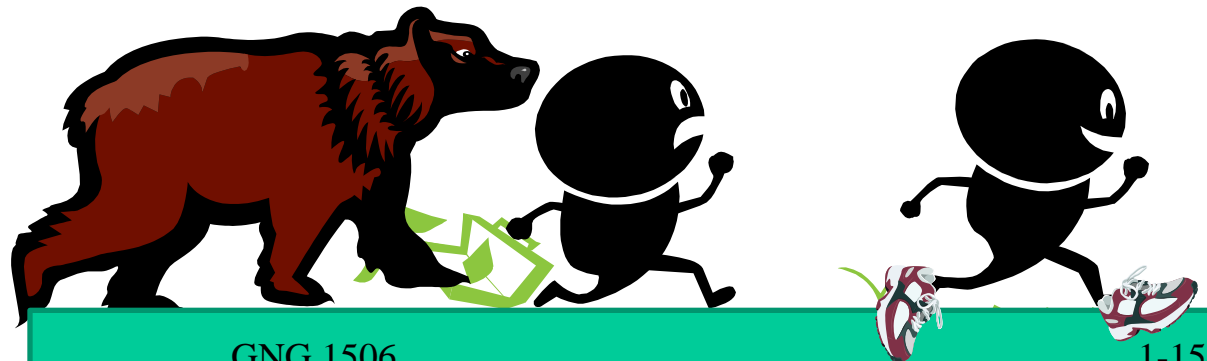


Attention au mauvais problème

- L'ours
- Longues files
- Ascenseurs lents

L'ours

- Mike et Jack marchent dans les bois quand soudainement un ours les attaque, les forçant à monter à un arbre. Comme l'ours commence à monter à l'arbre aussi, il devient évident que Mike et Jack doivent descendre et s'enfuir. Jack commence à mettre ses chaussures de sport:
- **Mike** : Vous êtes un idiot, Jack! Il est impossible avec ces chaussures de sport de courir plus vite que l'ours!
- **Jack** : En réalité, Mike, je ne dois pas courir vite que l'ours. Je dois courir juste plus vite que toi.



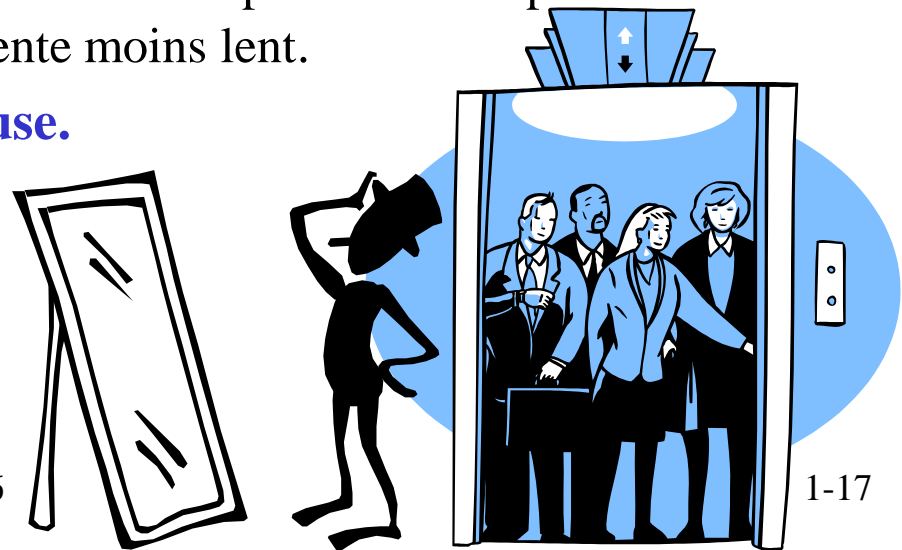
Longues files

- Un parc d'attractions reçoit beaucoup de plaintes que leurs files d'attente sont trop longues.
- Ils augmentent leur infrastructure de tour et le personnel pour accélérer les files d'attente, mais les plaintes continuent toujours.
- Un de leurs ingénieurs recommande la solution suivante (qui marche et réduit significativement le nombre de plaintes):
- À des distances spécifiques à la ligne, **mettre un panneau indiquant combien de minutes restent pour passer à l'avant de la ligne**. Mais au lieu de mettre la bonne valeur, on affiche une valeur qui est de 1,5 fois, ce qui manipule les attentes de la population.
- Problème semblable et solution pour longs feux de signalisation.



Ascenseurs lents

- La direction d'un immeuble de grande hauteur a reçu trop de plaintes d'occupants signalant que les ascenseurs sont lents. Ils demandent à la compagnie de l'ascenseur de l'aide.
- Après d'énormes dépenses, les ascenseurs fonctionnent plus rapidement mais les gens se plaignent toujours. La compagnie affirme que compte tenu de l'infrastructure de l'immeuble et le nombre d'occupants à chaque étage, les ascenseurs sont aussi vite qu'ils peuvent être.
- La direction demande alors à un de leurs occupants (un ingénieur) de l'aide qui, après l'étude du cas, recommande que des miroirs **soient installés autour des cadres de porte d'ascenseur**. Le nombre de plaintes diminue significativement après que les miroirs sont installés.
- Pourquoi? Parce que le vrai problème est l'impatience des gens. Donnez aux gens quelque chose à faire comme se regarder dans le miroir pour distraire pendant une courte période, ce qui semble rendre le temps d'attente moins lent.
 - **Le temps passe vite quand on s'amuse.**
 - **Plus on attend une chose, plus elle se fait attendre.**



Identification du problème

- Quel est le problème signalé?
 - Pourquoi c'est un problème?
 - Pourquoi que c'est un problème?
 - ...
- Il pourrait y avoir plusieurs solutions à un problème:
 - Ours: Concevoir et fabriquer une chaussure qui vous permet de courir plus vite qu'un ours!
 - Longues Files: Augmenter la capacité de tour et embaucher plus de personnel.
 - Ascenseur : Augmenter le nombre d'ascenseurs et leur vitesse.
 - Mais ces solutions sont toutes très chères!
- Un ingénieur considérera toujours des alternatives.

Collecte de l'information

- Quelles sont les contraintes?
- Quelles sont les lois de la physique?
- Quelles sont les équations mathématiques?

Générer une solution possible

- Il peut y avoir plusieurs solutions possibles
 - Il faut choisir une.
- Définir un algorithme.
- Algorithme
 - une liste des étapes à suivre pour résoudre au problème dans lequel l'ordre est important.
 - précise les actions à exécuter et l'ordre dans lequel ces actions doivent être exécutées.

Algorithme de la routine du matin

- Quitter du lit,
- Prendre une douche,
- S'habiller,
- Prendre le petit-déjeuner,
- Prendre la voiture pour aller travailler.

Algorithme de la routine du matin mal ordonnée

- Quitter du lit
- S'habiller
- Prendre une douche
- Prendre le petit-déjeuner
- Prendre la voiture pour aller travailler
 - Arrive au travail mouillé!

Techniques de génération de solution

- Remue-méninges
 - L'expérience aide
- Inversion
 - Utilisation du bas vers le haut.
- Recherche dans le Web!
 - Beaucoup de solutions existent déjà et peuvent être trouvées si vous le cherchez.
 - Prudent : est-ce qu'ils sont fiables?
 - Tout sur le Web n'est pas correct.

Mettre en œuvre la solution

- Noter les règles.
- Construire la machine (matériel).
- Écrire le programme (logiciel).

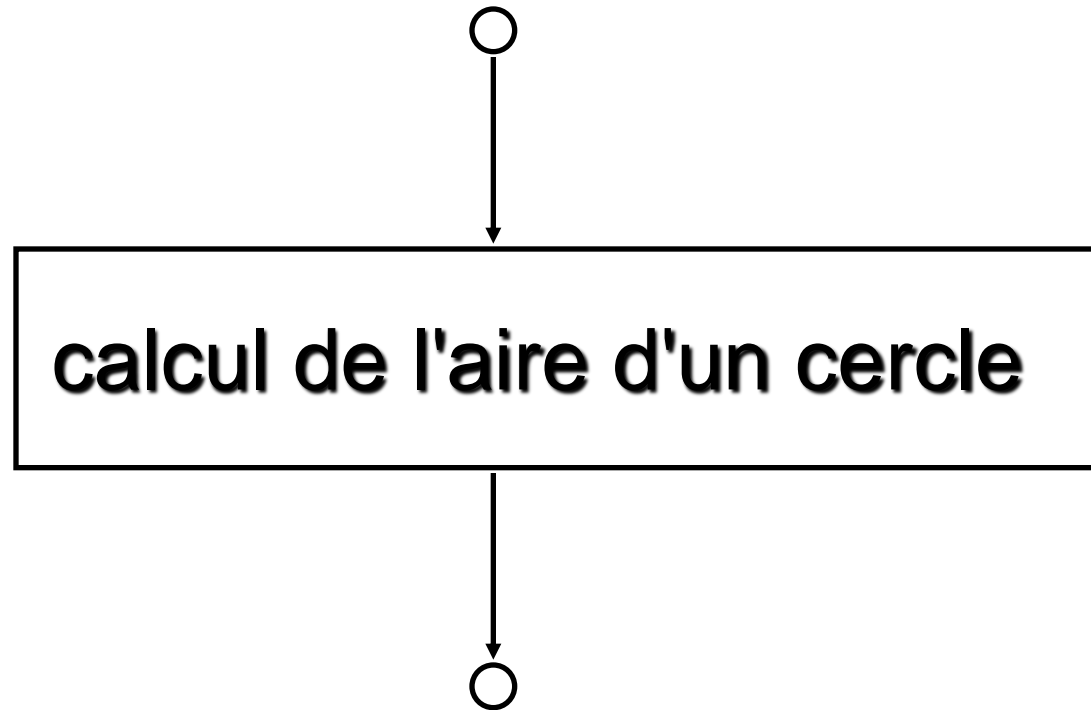
Tester la solution

- Vérifier que la solution fonctionne dans de nombreuses situations différentes.
- Essayer différentes entrées.
- Tester avec les mauvaises entrées.

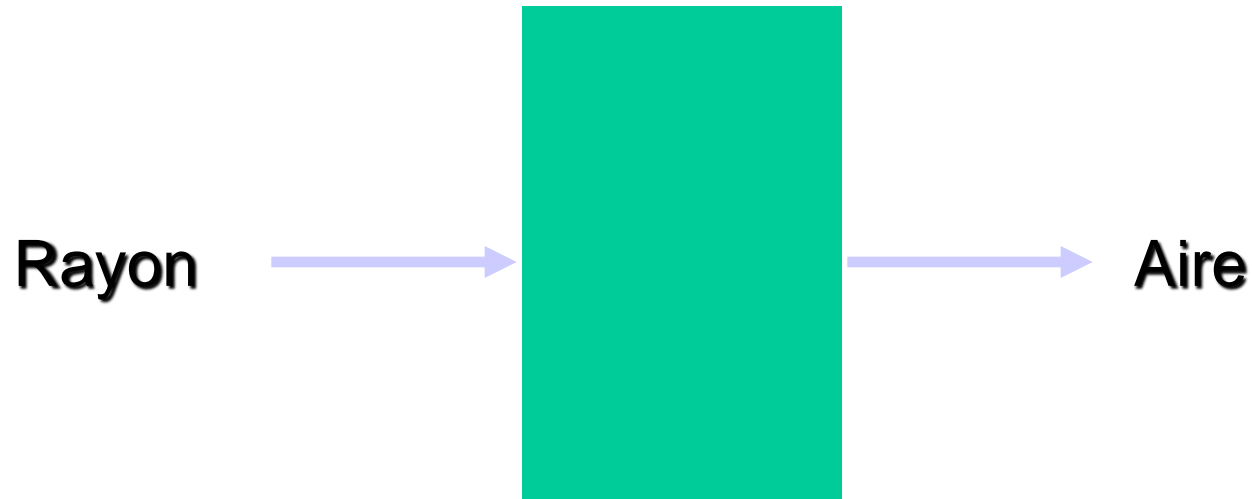
Exemples

Définition du problème 1

- ◆ **Comment calculer l'aire d'un cercle?**



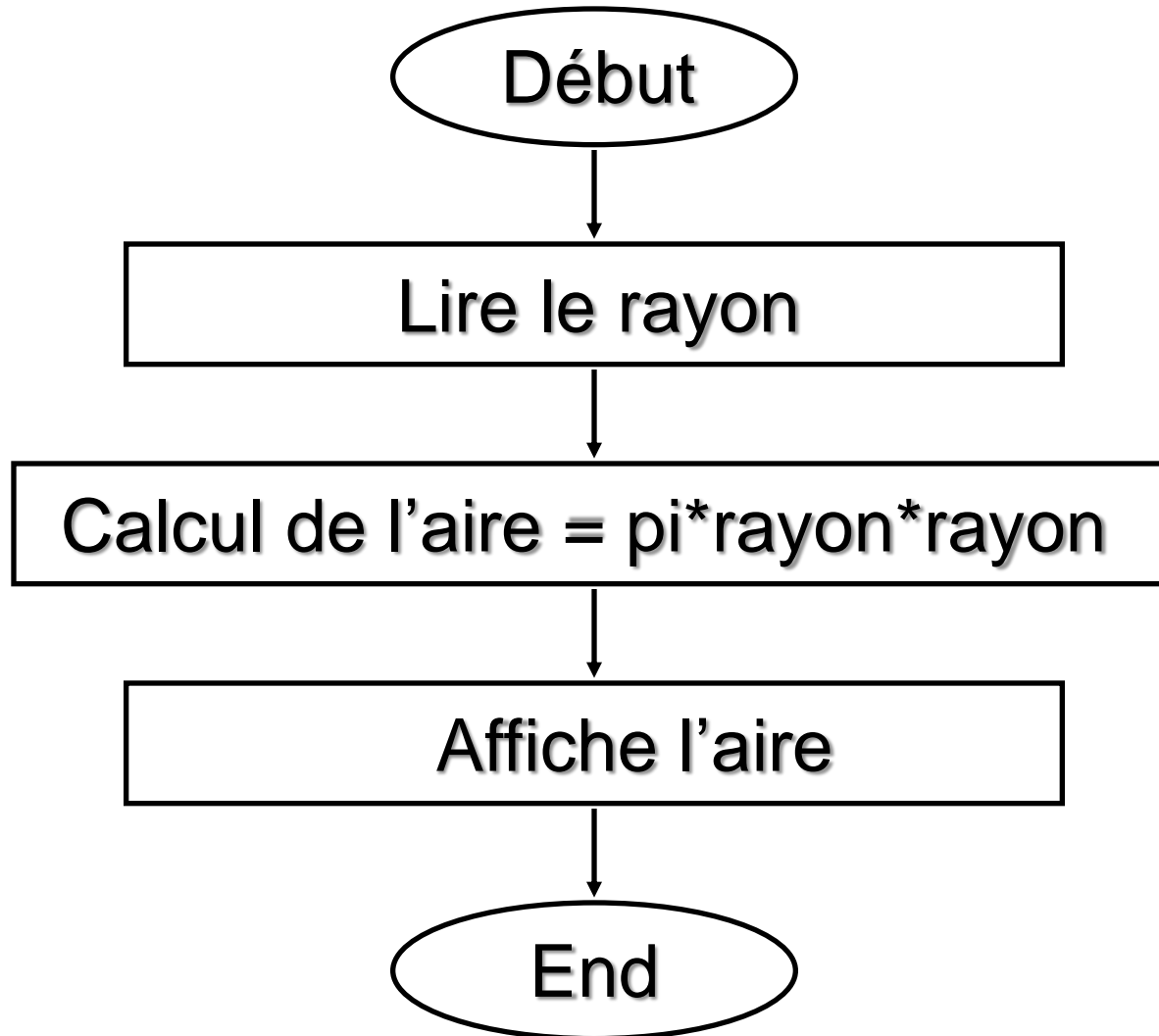
Description d'entrée-sortie



Collecte de l'information

- Mathématiques: $a = \pi r^2$
- aire = $\pi \times \text{rayon} \times \text{rayon}$
- aire = $\text{pi} * \text{rayon} * \text{rayon}$

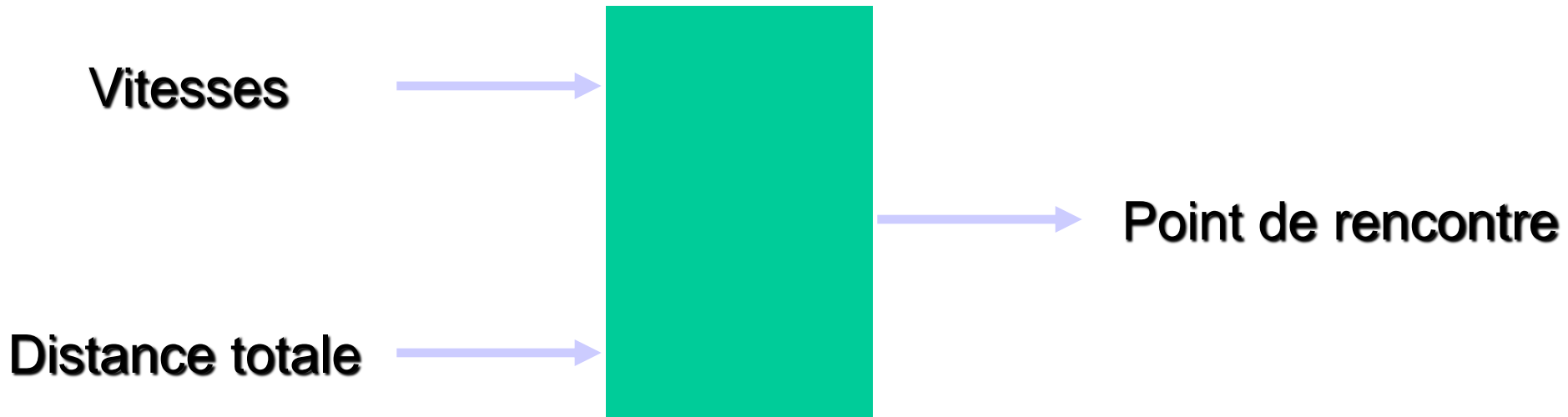
Algorithme



Définition du problème 2

- Un train de voyage quitte Montréal à 10 heures vers Toronto à 100 km/h.
- Dans le même temps, un train quitte Toronto en direction de Montréal à 150km/h.
- Si la distance entre Montréal et Toronto est 550 km, où les trains se croisent-ils?

Description d'entrée-sortie



Collecte de l'information (1/2)

- distance = vitesse * temps
- Pour le train qui part de Montréal:

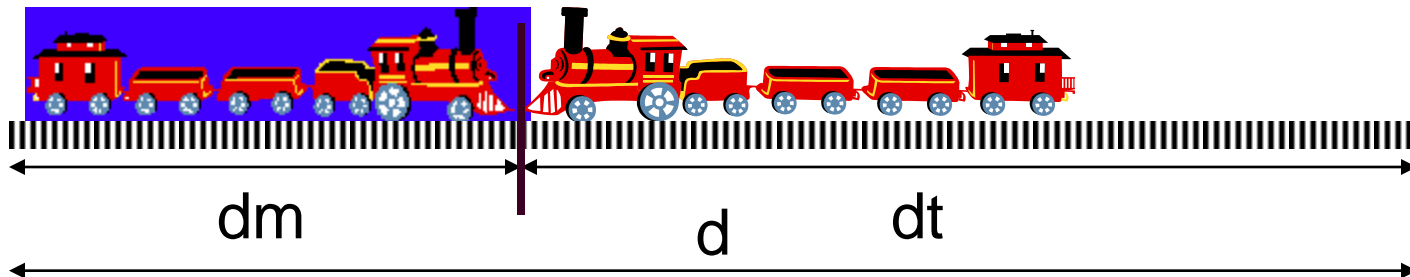
$$d_m = v_m * t \rightarrow dm = vm * t$$

- Pour le train qui part de Toronto:

$$dt = vt * t$$

- La distance totale de Montréal à Toronto est :

$$d = dm + dt = 550 \text{ Km}$$



Collecte de l'information (2/2)

- Les trains se croisent après avoir voyagé pendant le même temps t (même heure départ):

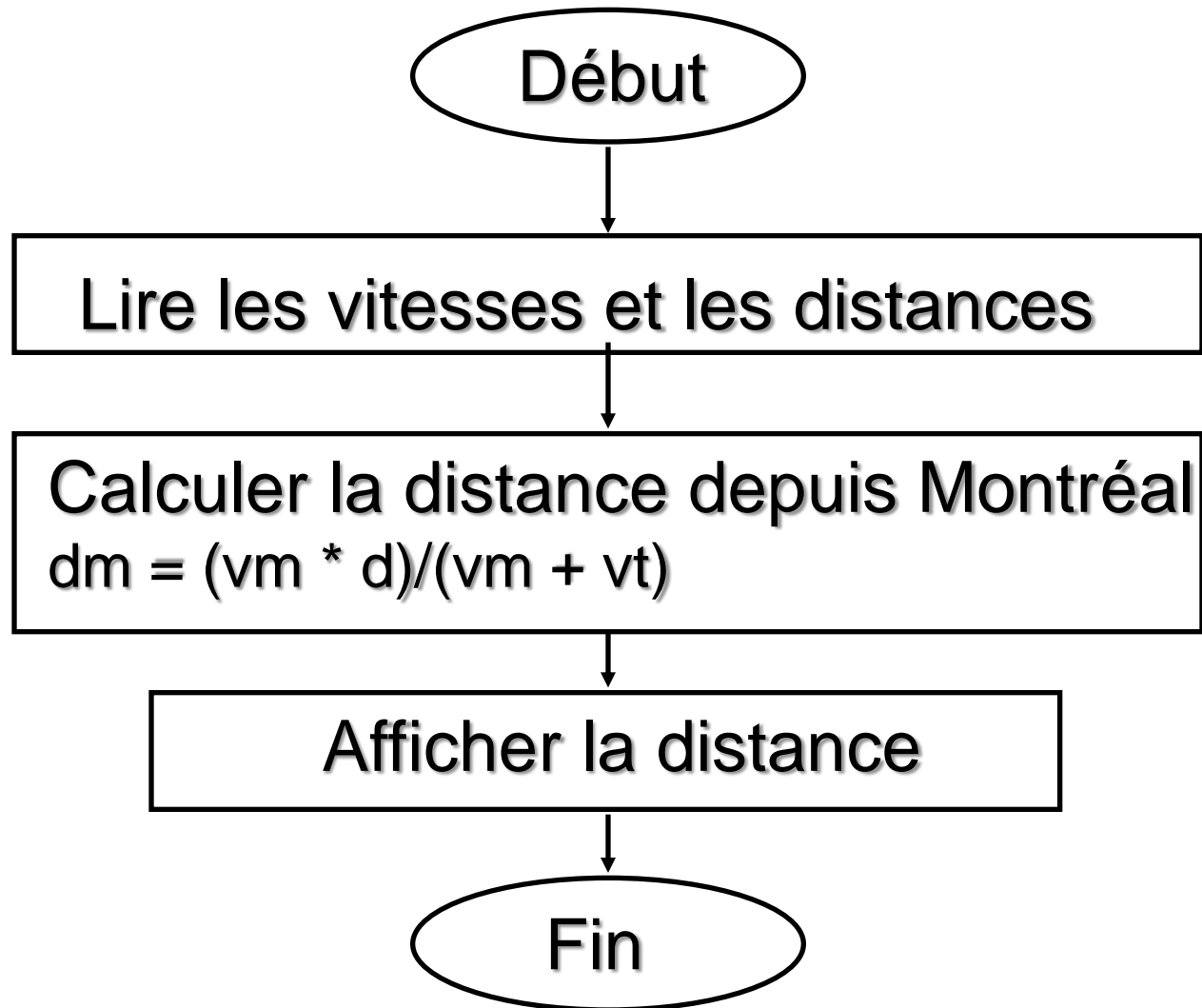
$$d = d_m + d_t = (v_m + v_t) * t$$

$$t = \frac{d}{(v_m + v_t)}$$


$$d_m = v_m * t$$

$$d_m = \frac{(v_m * d)}{(v_m + v_t)}$$

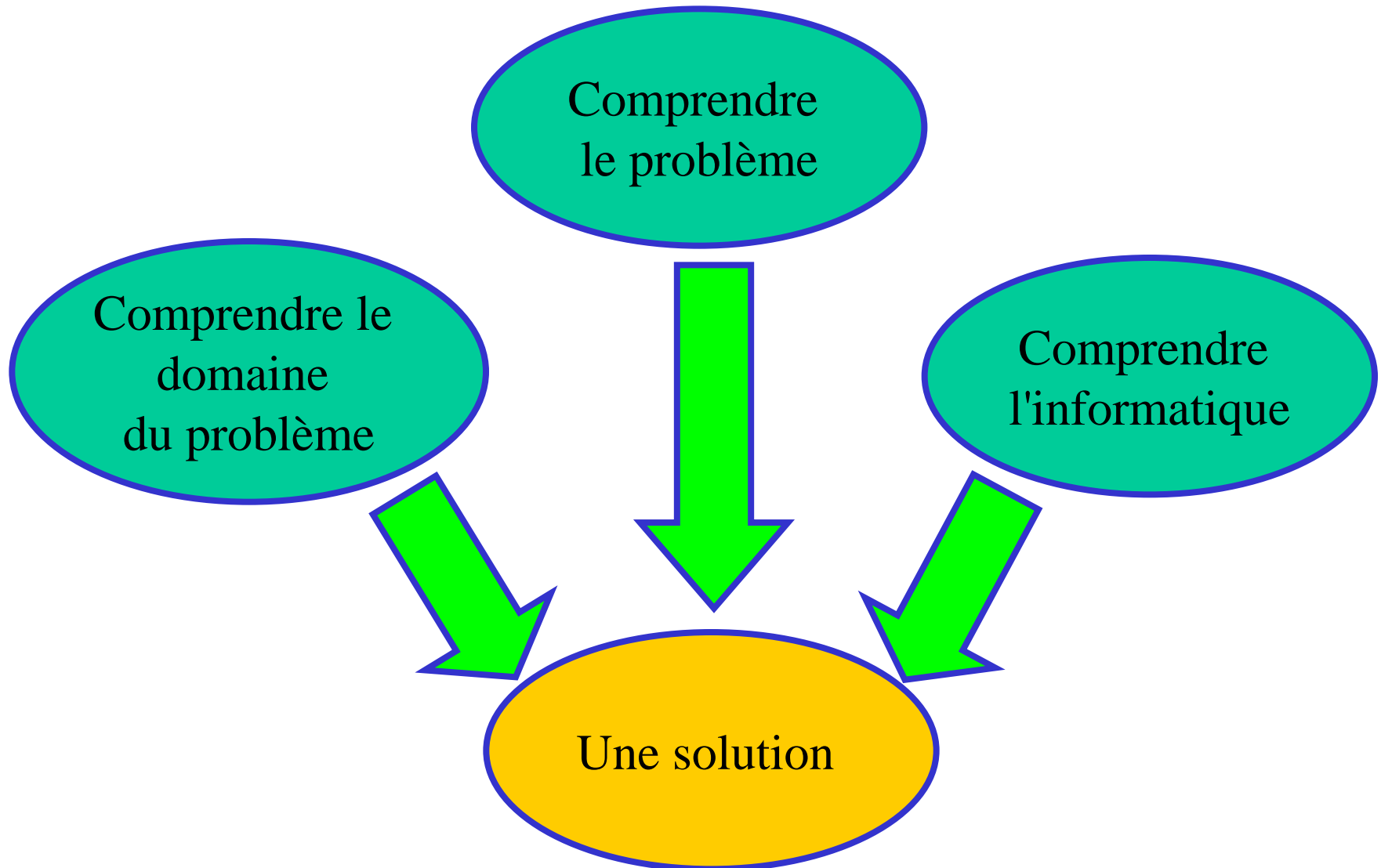
Algorithme



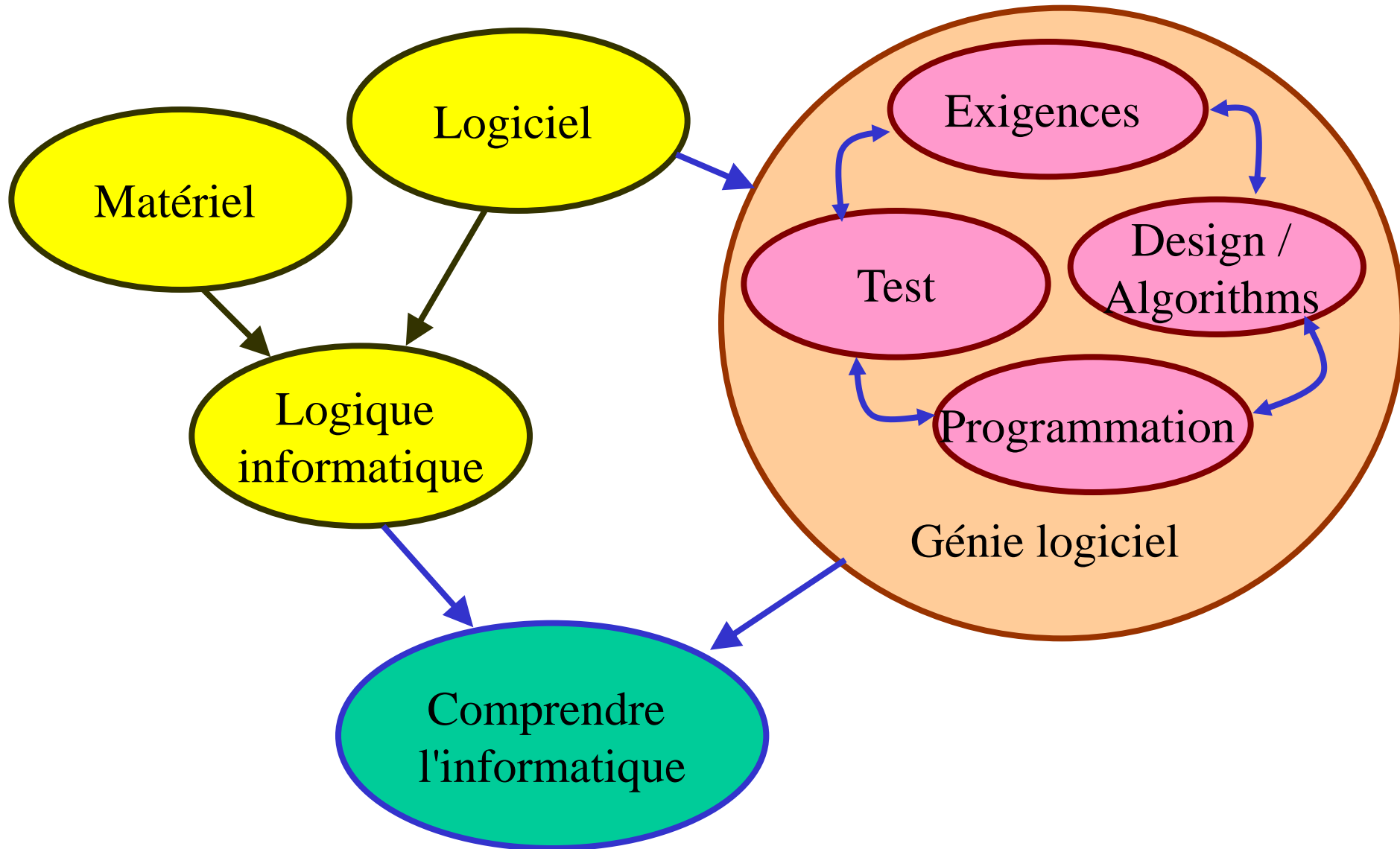
Vérification

- $dm = (100 * 550)/(100 + 150) = 220\text{km}$
- $dt = d - dm = 550 - 220 = 330\text{km}$
- $t = dm/vm = 220/100 = 2.2\text{h}$
- $dt = vt * t = 150 * 2.2 = 330\text{km}$  Vérification

Résoudre des problèmes avec l'informatique



Quelques concepts du système informatique



Introduction à l'informatique

- Qu'est-ce qu'un ordinateur?
 - Un ordinateur est un appareil qui est capable de faire des calculs et des décisions logiques à des vitesses beaucoup plus rapide que les humains.
 - Exemple: Un PC en 1996 pourrait faire environ 10^7 additions/sec tandis qu'un super-ordinateur pourrait faire environ 10^{11} additions/sec.
 - Un ordinateur fonctionne sur des données selon les directives ou des commandes qui sont répertoriés dans un programme. Le programme est écrit par un être humain (en général) à l'aide d'un langage de programmation.
 - L'ordinateur (dispositif matériel) et le programme (logiciel ou commandes) sont nécessaires pour un ordinateur afin d'accomplir une tâche utile.

Organisation d'un ordinateur (1/2)

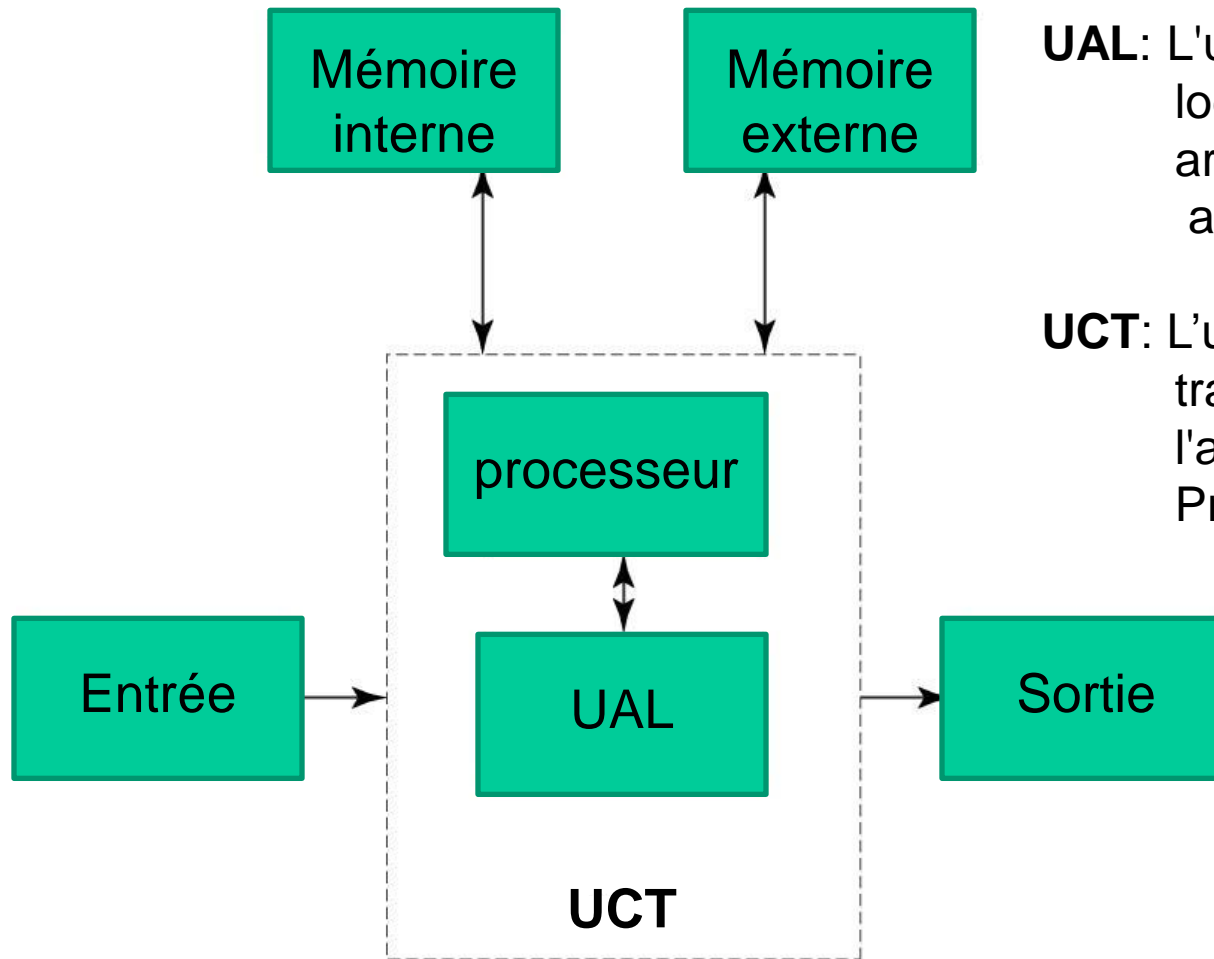
L'organisation logique de la plupart des ordinateurs à usage général est le suivant:

- **Unité d'entrée**
 - Obtenir des informations comme des données et des programmes utilisant des dispositifs comme un clavier, DVDs, des disques durs et des cartes d'accès réseau.
- **Unité de sortie**
 - Rendre disponible au monde extérieur les résultats de calculs utilisant des dispositifs comme un écran, des haut-parleurs, un accès réseau, etc.
- **Unité de mémoire (RAM).**
 - À faible capacité de stockage par rapport à un disque dur mais a une vitesse d'accès rapide.
 - Le stockage est temporaire, c'est-à-dire les données sont perdues lorsque l'alimentation de l'unité de mémoire est désactivée.
 - Habituellement, cette mémoire contient les programmes en exécution ainsi que les données d'entrée nécessaires et des données de sortie.

Organisation d'un ordinateur (1/2)

- **Unité centrale de traitement (UCT).**
 - Section administrative de l'ordinateur qui coordonne le fonctionnement des autres sections. Elle Synchronise le fonctionnement de l'ordinateur.
- **Unité arithmétique et logique (UAL).**
 - Contient les mécanismes de traitement pour effectuer des opérations arithmétiques de base telles que l'addition, la soustraction, la multiplication et la division ainsi que les mécanismes nécessaires à la prise de décision logique.
 - Elle fait partie de l'UCT.
- **Unité de stockage secondaire.**
 - Cette unité a de grande capacité stockage (exemple: le disque dur).
 - L'accès est lente par rapport à la mémoire RAM.
 - Elle contient des programmes et des données qui ne sont pas en usage immédiat.

Organisation interne d'un ordinateur

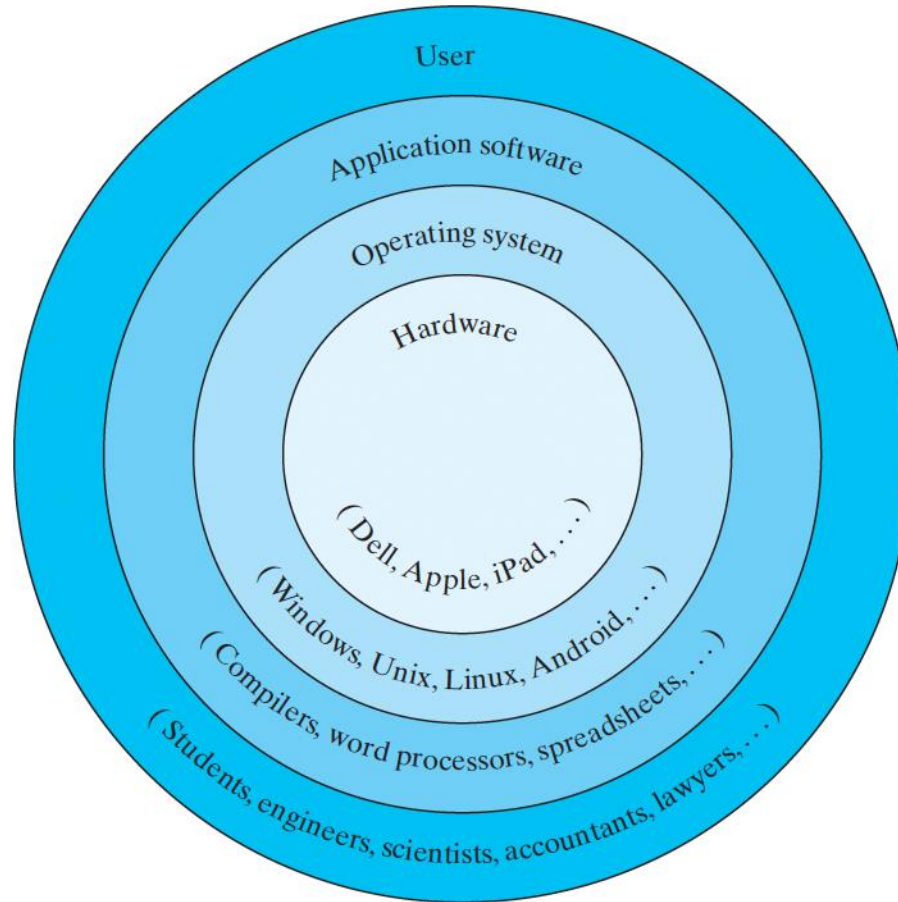


UAL: L'unité arithmétique et logique (ou bien ALU, arithmetic logic unit en anglais)

UCT: L'unité centrale de traitement (ou CPU de l'anglais Central Processing Unit)

Engineering Problem Solving with C, Fourth Edition

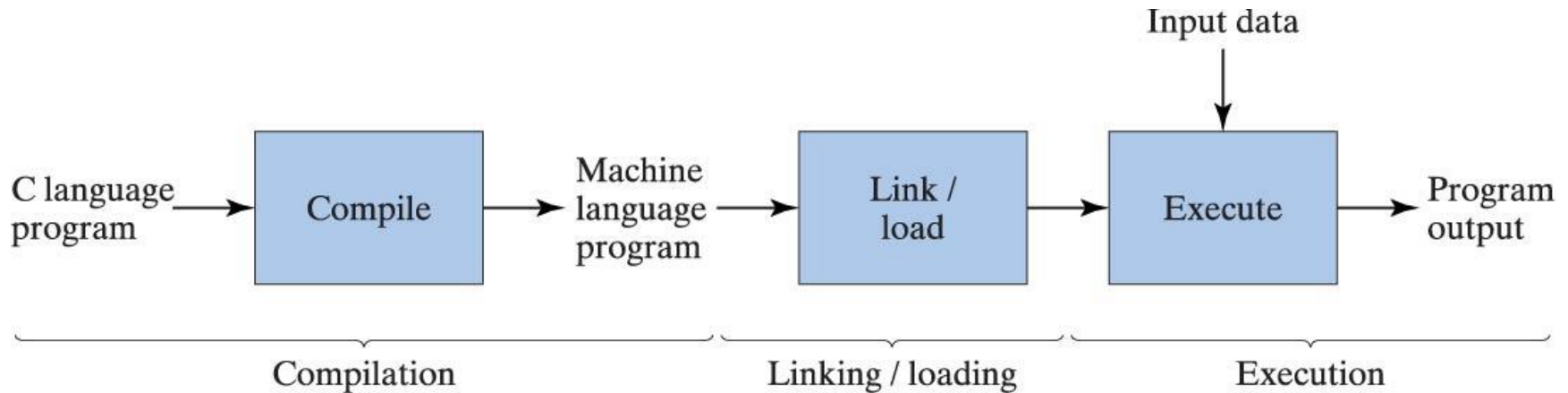
Interface logicielle de l'ordinateur



Engineering Problem Solving with
C, Fourth Edition

Programme

Compilation/Linkage/Exécution



Engineering Problem Solving with C, Fourth Edition

Qu'est-ce que un système de traitement informatique?

- Le traitement des **données** selon un algorithme - ou la résolution de problèmes!
- **La connaissance est divisée dans deux catégories**
 - **Déclaratif**
 - ❖ Les déclarations de faits!
 - ❖ Quelles sont les choses!
 - **Impératif**
 - ❖ Une description de la façon de faire quelque chose (c'est une recette).
 - ❖ Ceci est le calcul (traitement)!
- Exemple: racine carrée!

Langage de programmation

- **Pour la mise en œuvre des algorithmes, nous avons besoin d'un langage.**
 - Dans ce cours GNG1506, nous allons utiliser le langage C.
- **Syntaxe**
 - Quelles sont les expressions légales dans le langage.
 - Exemple: Elle, il, etc.
- **Sémantique**
 - Quelles expressions font du sens.
 - Qu'est-ce que le programme fait (ce qui se passe quand je lance un programme)
 - L'exemple d'une sémantique incorrecte (bien que la syntaxe soit correcte)
 - Mon bureau est Alex!

Types du langage de programmation (1/2)

Trois classifications ou catégories de langages de programmation

- **Code machine**

- Le seul langage que le matériel informatique peut exécuté **directement**.
- **Tous** les programmes et les données doivent être traduits en langage machine.
- Le langage dépend de la machine.
- Code machine est généralement difficile à manipuler car elle est essentiellement constituée de nombres.
 - Par exemple: le code de machine pour ajouter des variables x et y et stocker le résultat dans z:

FC 0900

F3 0902

7C 0904

Types du langage de programmation(2/2)

• Langage Assembleur

- Les commandes numériques de code de la machine sont associés à des abréviations ou des mnémoniques.
- Un programme est écrit en utilisant les abréviations et un assembleur traduit le programme en code machine. Le langage assembleur est trop lourd à manipuler pour écrire des programmes importants et complexes.
 - Voici un exemple d'un code assembleur pour ajouter deux variables x et y et stocker le résultat dans z:
 - LOAD x
 - ADD y
 - STORE z

• Langues de haut niveau

- Composé d'un grand nombre de commandes puissantes.
- Les commandes sont descriptives, mathématiques.
- Un **compilateur** traduit le programme écrit dans un langage de haut niveau à un programme en code machine.
 - Voici un exemple d'un code C pour ajouter deux variables x et y et stocker le résultat dans z: $z = x + y$;

Cycle de vie d'un programme

- La programmation utilisant un langage de haut niveau comme C, implique de nombreuses itérations:
 - **Editeur de texte**: `program1.c`
 - **Prétraitement** exécute les directives du préprocesseur qui s'appliquent au code de programme avant la compilation. Les directives du préprocesseur communs comprennent la substitution et l'insertion de code.
 - **Compilation** du code à l'aide du compilateur de langage requis. Cela se traduit par le programme de langage de haut niveau écrit par le programmeur de langage machine. Un fichier de langage machine est créé de la forme: `program1.obj`
 - **Lier** le code objet avec les bibliothèques nécessaires (mathématique, entrée/sortie et la manipulation de chaînes de caractère). L'éditeur de liens crée un fichier exécutable qui peut être chargé sur la machine; le nom du fichier exécutable a généralement la forme: `Program1.exe`
 - **Chargement** du code exécutable dans la mémoire.
 - **L'exécution** du programme par l'ordinateur.

Langues populaires de haut niveau

1. **COBOL** (COmmon Business Oriented Language)
2. **FORTRAN** (FORmula TRANslation)
3. **BASIC** (Beginner All-purpose Symbolic Instructional Code)
4. **Pascal** (named for Blaise Pascal)
5. **Ada** (named for Ada Lovelace)
6. **Java** (très généralement utilisé)
7. **Visual Basic** (langage visuel développé par Microsoft)
8. **Delphi** (langage visuel similaire à Pascal développé par Borland)
9. **C++** (langage orienté objet basé sur C)
10. **C#** (très similaire à Java, créé par Microsoft)
11. **C** (nous utilisons dans ce cours)

Pseudo-code

- Les instructions de pseudocode consistent en **texte de langue naturelle (anglaise, française, etc.)** pour décrire l'action désirée pour être pris par l'ordinateur. e.g.

Assigne 2 à var1

- Une ligne de pseudocode correspond normalement à une instruction de langage de programmation.
 - Parfois, une seule ligne de pseudocode correspond à plusieurs instructions en langage de programmation.
- Les blocs d'instructions (une séquence d'instructions) sont importantes lors de la création des structures de codage (décision et le bouclage des structures).
- Dans les notes de cours, le **pseudocode est écrit avec la police *Script***.