Cours N°3

Concepts de Base

1. Définition de l'Informatique

- Le mot ***informatique***, proposé par l'ingénieur français Philippe DREYFUS, en 1962, est une contraction des mots ***information*** et ***automatique.***
- Définition accepté par l'Académie Française : ***"****Science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communication dans les domaines techniques, économiques et sociaux****".***
- L'informatique désigne l'ensemble des sciences et techniques en rapport avec le traitement de l'information.

1. Définition de l'Informatique (Suite)

- L'informatique n'est pas fondamentalement liée à l'utilisation des ordinateurs. Surtout elle se fonde sur des études théoriques de logique, de mathématiques, de linguistique, de grammaire formelle, de compilation et bien évidemment de structure d'ordinateur.

  À cet égard, Edsger Dijkstra (Mathématicien et informaticien néerlandais du XXe siècle) disait :
  " ***L'informatique n'est pas plus la science des ordinateurs que l'astronomie n'est celle des télescopes.*** "

2. Branches de l'Informatique

L'informatique est subdivisée en de nombreuses branches plus ou moins spécialisées dont on peut citer:

- **Informatique formelle ou analytique:** branche de l'informatique la plus proche des sciences exactes
- **Informatique systématique et logique:** qui étudie l'architecture des systèmes informatiques
- **Informatique physique et technologique:** qui s'attache à l'étude et à la réalisation des composants et sous-ensembles électroniques
- **Informatique méthodologique:** qui se rapporte aux recherches en méthodologie de la programmation
- **Informatique appliquée:** qui s'occupe concrètement de l'application de l'informatique dans les divers domaines de la vie économique, culturelle et sociale

3. Étapes de Résolution d'un Problème Informatique

Pour résoudre un problème informatique, il faut:

- ***Analyser ce problème:*** définir avec précision les résultats à obtenir, les informations dont on dispose, ...
- ***Déterminer les méthodes de résolution:*** il s'agit de déterminer la suite des opérations à effectuer pour obtenir à partir des données la solution au problème posé. Cette suite d'opérations constitue un ***algorithme***.
- ***Formuler l'algorithme définitif:*** cette étape doit faciliter la résolution sur ordinateur par l'expression de l'algorithme dans un formalisme adéquat.
- ***Traduire l'algorithme dans un langage de programmation adapté***.

4. Notion d'Algorithme (suite)

- **Exemples d'algorithmes :**

Exemple2: ***Tri d'un jeu de cartes suivant la couleur***

1) Prendre la première carte
2) La carte est-elle rouge?

   Si oui, poser la carte sur le premier tas

   Sinon, poser la carte sur le second tas
3) Reste-t-il des cartes?

   Si oui, prendre la carte suivante et continuer sous 2

   Sinon, fin du tri

4. Notion d'Algorithme (suite)

❑ Exemples d'algorithmes :

Exemple3: ***Calcul des racines d'un polynôme du 2nd ordre: a x² + b x + c = 0 , ∀ (a,b,c)≠0***

1) Saisir les valeurs de (a, b, c)

2) On calcule $\Delta = b^2 - 4ac$

Si $\Delta < 0$ alors pas de racine dans IR

Si $\Delta = 0$ alors racine double $x = -\frac{b}{2a}$

Si $\Delta > 0$ alors deux racines :

$$x_1 = \frac{-b - \sqrt{\Delta}}{2a} \qquad x_2 = \frac{-b + \sqrt{\Delta}}{2a}$$

4. Notion d'Algorithme (suite)

- Un *algorithme* est une suite d'actions qui, correctement exécutées donneront le résultat désiré (attendu).
- Un *algorithme* est le résultat de la décomposition d'un problème complexe en opérations élémentaires à exécuter en plusieurs étapes successives.
- Un *algorithme* est toujours exécuté par un *processeur*. Il peut être une personne, un dispositif électronique, mécanique ou un ordinateur. C'est toute entité en mesure de comprendre et d'exécuter les actions constituant un *algorithme.*
- L'ensemble des objets (éléments) nécessaires à la réalisation d'un travail décrit par un algorithme est appelé *environnement.*

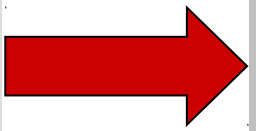
4. Notion d'Algorithme (suite)

❑ Définitions:

✓ Un algorithme est une séquence (suite) d'actions élémentaires, qui exécutées par un processeur bien défini réalisera un travail bien précis (demandé).

✓ Un algorithme est une suite de règles, de raisonnements ou d'opérations, qui transforment des grandeurs données (données d'entrée) en d'autres grandeurs (données de sortie).

Entrée:
Normalement, un algorithme possède une ou plusieurs données d'entrée [input data], c-à-d des valeurs qui sont connues avant son exécution et sur lesquelles l'algorithme est appliqué.



Sortie:
Un algorithme possède une ou plusieurs données de sortie [output data], c-à-d des valeurs produites par lui-même. Ces données sont en relation exactement spécifiée avec les données d'entrée.

4. Notion d'Algorithme (suite)

❑ Propriétés:

– L'algorithme doit tenir compte de tous les cas possibles. Il traite le cas général et les cas particuliers

– Il contient toujours un nombre fini d'actions

– L'ordre des actions est important (exécution séquentielle)

– Chaque action doit être définie avec précision, sans aucune difficulté

– Certaines actions peuvent être raffinées (décomposées)

– L'algorithme n'est pas nécessairement unique

– Il doit produire le résultat désiré

4. Notion d'Algorithme (suite)

❑ Formalisme algorithmique:

Un formalisme algorithmique est un ensemble de conventions (ou de règles) dans lequel on exprime toute solution d'un problème donné.

4. Notion d'Algorithme (suite)

❑ Structure générale d'un algorithme:

4. Notion d'Algorithme (suite)

- **Exemple 1 :** *Addition de deux nombres réels*

```
Algorithme Addition
Variables utilisées:
A, B, Somme : nombres Réels
    1) Début
    2) Lire (A,B)
    3) Somme=A+B
    4) Écriture (Somme)
    5) Fin
```

- **Exemple 2 :** *Calcul des racines d'un polynôme du 2ème ordre :*

$$a\,x^2 + b\,x + c = 0\ ,\ \forall\,(a,b,c)\neq 0$$

5. Notion d'Organigramme

❑ Définitions:

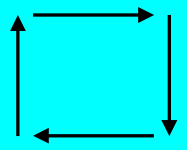
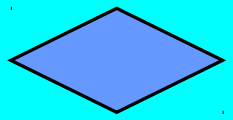
✓ Un *organigramme* est un schéma symbolique conventionnel qui illustre les étapes d'un algorithme et leurs relations.

✓ Nous utilisons l'*organigramme* parce qu'une représentation graphique aide à la compréhension.

✓ L'*organigramme* est un schéma fonctionnel qui présente les différentes parties d'un programme les unes à la suite des autres en utilisant des symboles graphiques pour visualiser l'exécution du programme et le cheminement des données.

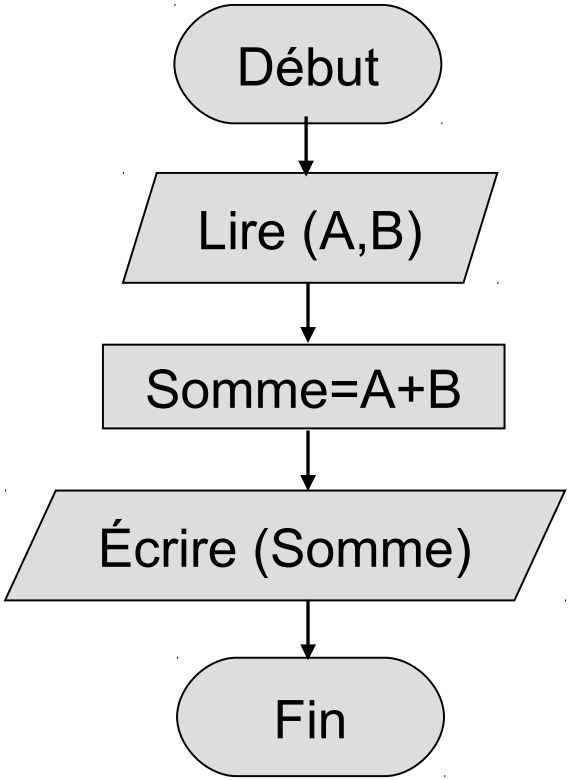
5. Notion d'Organigramme (suite)

❑ Principaux Symboles d'un Organigramme:

Noms	Symbole	Définition
Flèches		Elles indiquent le sens du traitement (haut, bas, gauche, droite).
Début / Fin		Ce symbole indique le début ou la fin de l'organigramme
Entrée / Sortie		Ce symbole indique les données d'entrées et de sorties
Boite de traitement		Elle indique un traitement spécifique qui peut être exécuté
Boite de décision (Test)		Elle permet d'envoyer le traitement sur un chemin ou sur un autre, selon le résultat du test

5. Notion d'Organigramme (suite)

- **Exemple 1 :** *Addition de deux nombres réels*

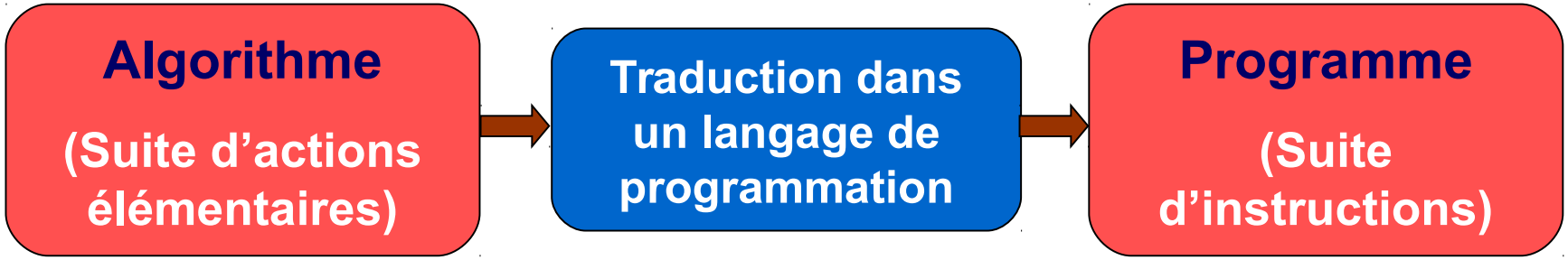


- **Exemple 2 :** *Calcul des racines d'un polynôme du 2nd ordre :*
  $a\,x^2 + b\,x + c = 0\ ,\ \forall\,(a,b,c)\neq 0$

6. Programmes et Langages de Programmation

❑ Notion de Programme:

Un programme est une séquence d'instructions écrites dans un langage de programmation traduisant un algorithme. Chacune de ses instructions spécifie l'opération que doit exécuter l'ordinateur.



6. Programmes et Langages de Programmation

❑ Langage de Programmation:

– Un langage de programmation est un langage artificiel comprenant un ensemble de caractères, de symboles et de mots régis par des règles qui permettent de les assembler, utilisé pour donner des instructions à une machine.

– Les langages de programmation permettent de définir les ensembles d'instructions effectuées par l'ordinateur lors de l'exécution d'un programme.

– Il existe plusieurs langages de programmation, la plupart d'entre eux étant réservés à des domaines spécialisés. Exemple: *Fortran, C, C++, Java, Html, Pascal …*

6. Programmes et Langages de Programmation

❑ Compilateur:

– Tout langage possède un compilateur ou du moins un interpréteur.

– Il sert à traduire le programme écrit avec le langage (programme source) en langage machine (codes) afin qu'il soit compris par l'ordinateur.

– Il permet aussi d'analyser le programme source pour détecter les erreurs de syntaxe commises par le programmeur.

6. Programmes et Langages de Programmation

❑ Du problème au résultat: