

Chapitre 1 : Introduction

1. Résolution d'un problème par ordinateur

L'ordinateur est une machine capable d'exécuter automatiquement une suite d'opérations. Pour résoudre un problème à l'aide d'un ordinateur, il faut :

1. Analyser ce problème : définir ce que j'ai comme données et ce que j'ai besoin comme résultat.
2. Déterminer la méthode de résolution : déterminer la suite des opérations à effectuer pour résoudre le problème posé. Plusieurs méthodes peuvent être trouvées pour résoudre un seul problème, il faut choisir la plus efficace.
3. Formuler l'algorithme définitif : représenter la méthode de résolution par un algorithme écrit en langage algorithmique, dit aussi un langage de description d'algorithme (LDA).
4. Traduire l'algorithme en un langage de programmation adapté.

2. Notion d'algorithme

Le mot algorithme provient du nom du célèbre mathématicien arabe : Muhammad ibn Musa Al Kwarizmi (780-850). Son nom donna au moyen-âge le nom "algorisme" qui devint algorithme avec Lady Ada Lovelace (1792-1871).

Un algorithme est une suite d'opérations élémentaires exécutées dans un ordre donné pour résoudre un problème ou accomplir une tâche. En tant que science, on parle de l'algorithmique.

3. Exemple

Soit le problème de calcul de la somme de deux nombres. Ce problème peut être résolu de la manière suivante:

A. Analyse

Pour le calcul de la somme:

- Nous avons besoin en entrée de deux valeurs : valeur1 et valeur2.

- En sortie, nous aurons la somme de ces deux valeurs.

B. Solution

Le calcul de la somme consiste à:

1. Avoir les deux valeurs (lire valeur1 et valeur2).
2. Additionner les deux valeurs.
3. Afficher le résultat (Ecrire la somme).

Cette forme de représentation d'un algorithme est dite "Enumération des étapes".

C. Ecriture en LDA

En informatique, on utilise un langage algorithmique pour écrire un algorithme ; et l'algorithme au-dessus devient.

Algorithme Somme ;

Variables

valeur1, valeur2, som : entier ;

Début

Lire (valeur1, valeur2) ;

som ← valeur1 + valeur2 ;

Ecrire (som) ;

Fin.

Comme illustré dans l'exemple précédent:

- Un algorithme commence par le mot Algorithme suivi de son nom et un point virgule. Généralement le nom de l'algorithme indique sa fonction.
- Le mot Variables précède la liste des variables manipulées dans l'algorithme et leurs types. Les variables du même type sont séparées par des virgules. Deux déclarations différentes sont séparées par un point virgule.
- Les opérations de l'algorithme sont prises entre les mots Début et Fin indiquant le début et la fin de l'algorithme. Ces opérations sont séparées par des points virgules.
- Le mot Lire permet la lecture à partir du clavier. Le mot Ecrire permet l'affichage à l'écran.
- Le symbole ← correspond à l'opération d'affectation. Le symbole + est utilisé pour indiquer l'addition.
- Un algorithme se termine par un point.

D'autres mots et symboles utilisés dans notre langage algorithmique seront découverts dans le reste de ce cours.

Remarque : Le langage algorithmique varie d'un document à un autre. Par exemple, le symbole d'affectation représenté dans notre cours par \leftarrow peut être représenté dans un autre document par $::=-$.

Un algorithme doit être :

- Lisible : clair et facile à comprendre.
- De haut niveau : indépendant du langage de programmation et du système d'exploitation.
- Précis : non ambigu.
- Concis : réduire le nombre d'opérations ainsi que l'espace occupé en mémoire.
- Structuré : organisé.

D. Ecriture en langage de programmation

Pour être exécuté sur un ordinateur, un algorithme doit être traduit en un langage compréhensible par la machine, i.e. un langage de programmation, tel que Pascal, C, Java, etc.

Les algorithmes de ce cours sont traduits en langage Pascal. Le programme correspondant à l'algorithme précédent est le suivant.

```
program Somme ;
var valeur1, valeur2, som : integer ;
begin
  read(valeur1, valeur2) ;
  som := valeur1 + valeur2 ;
  write(som) ;
end.
```

Il est possible d'améliorer le programme précédent de la manière suivante:

```
program Somme ;
var valeur1, valeur2, som : integer ;
begin
  write('Entrez la première valeur : ');
  readln(valeur1) ;
  write('Entrez la deuxième valeur : ');
  readln(valeur2) ;
  som := valeur1 + valeur2 ;
  write('Somme = ', som) ;
end.
```

Remarques et définitions :

- Dans un algorithme, on parle d'opérations. Dans un programme, on dit instructions. Un programme est donc une suite d'instructions.
- Langage machine ou code machine ou code binaire : instructions de base d'un ordinateur correspondant aux opérations élémentaires de son CPU, et pouvant être directement exécutées sans traduction. Ces instructions sont codées en binaire.
- Compilateur : c'est un logiciel qui transforme un programme écrit en un langage de programmation en un programme dont les instructions sont écrites en langage machine, et donc directement exécutables par l'ordinateur. Le fichier exécutable est parfois appelé fichier binaire car les instructions du langage machine sont codées directement par des nombres binaires. Le fichier contenant le programme d'entrée est appelé fichier source.
- Le langage le plus facile à convertir en code machine est l'assembleur. L'assembleur ou langage d'assemblage, est un langage de programmation dont les opérations correspondent au langage machine (c'est-à-dire aux instructions élémentaires du CPU) mais sont écrites en abréviations textuelles plutôt qu'en code binaire.

Par exemple, un processeur de la famille x86 reconnaît une instruction du type

```
10110000 01100001
```

En langage assembleur, cette instruction est représentée par un équivalent plus facile à comprendre pour le programmeur

```
movb $0x61,%al
```

```
(10110000 = movb %al
```

```
01100001 = $0x61)
```

Ce qui signifie : « écrire le nombre 97 (la valeur est donnée en hexadécimal : $61_{16} = 97_{10}$) dans le registre AL ».