

Chapitre 1

Notions de base

1. Informatique

Le mot informatique est utilisé pour désigner des activités très différentes les unes des autres. Lorsque quelqu'un vous dit "je fais de l'informatique" vous n'êtes pas pour autant renseignés sur son activité ? Est-il secrétaire utilisateur de traitement de texte, est-il comptable utilisateur d'un logiciel de comptabilité, est-il physicien utilisateur des services d'un centre de calcul, est-il informaticien programmeur, analyste, administrateur de réseau, ingénieur système ?

Ce cours s'adresse à de futurs utilisateurs éclairés des systèmes informatiques qui, à défaut d'écrire eux même leurs programmes, auront besoin de communiquer de manière précise avec des informaticiens et donc de connaître la discipline de l'intérieur. Ce cours est donc conçu comme une initiation à l'**analyse** et à la **programmation**, il n'est pas l'apprentissage à l'utilisation de logiciel.

Avant d'écrire les programmes qui résoudront les problèmes, nous écrirons des algorithmes. On précise ce qu'est un **algorithme**.

2. Algorithme

Dans le sens général, un algorithme peut prendre de nombreuses formes :

- recette de cuisine
- notice de montage
- feuille de calcul d'impôt
- protocole expérimental
- ...

Dans tous les cas, il s'agit de la description, faite par un individu A, d'un enchaînement d'actions destinées à être exécutées par un individu B. Pour que B puisse exécuter sans erreur les tâches commandées par A, il faut que la description faite par A soit claire, complète, et sans ambiguïté. Cette description est une forme d'algorithme.

Pour nous, l'individu B, celui qui exécute, doit pouvoir être remplacé par un ordinateur. Il faut alors que les consignes soient données dans un langage interprétable par l'ordinateur. L'ensemble des ces consignes, écrite dans un langage de programmation, s'appelle un programme. Il est exécuté par l'ordinateur.

Comme il est difficile et peu souhaitable (pour des raisons qui seront développées plus tard) d'écrire directement un programme pour un ordinateur, on écrit des algorithmes, qui sont l'expression formalisée des solutions aux problèmes que l'on doit résoudre. Que veut dire formaliser ? Par exemple au lieu de poser la question "quel nombre faut-il ajouter à 7 pour obtenir 15, on peut formaliser la question sous la forme de l'équation $7 + x = 15$.

Voici l'exemple d'un algorithme qui ressemble à ce que les étudiants écriront dans quelques semaines.

3. Exemple

Algorithme Intérêts

/ calcule le nombre d'années à attendre pour obtenir une somme donnée avec un dépôt annuel identique chaque année. Ces données sont saisies au clavier par l'utilisateur. Les montants sont en euros */*

/ Déclarations */*

Constantes

/ identificateur = valeur rôle */*
 TAUX = 4.5 */* taux annuel */*

Variables

/ identificateur : type rôle */*
 depot : réel */* montant du dépôt annuel */*
 somme : réel */* somme courante */*
 sommeFinale : réel */* somme souhaitée */*
 nbAn : entier */* nombre d'années */*
 interet : réel */* montant de l'intérêt annuel */*

/ Instructions */*

Début

1 somme ← 0 */* on part d'une somme nulle */*
 2 nbAn ← 0 */* on part d'un nombre d'années nul */*

/ entrée des données de l'utilisateur */*

3 écrire("montant du dépôt annuel : ")
 4 lire(depôt)
 5 écrire("montant de la somme souhaitée : ")
 6 lire(sommeFinale)

/ traitement */*

7 **tantque** (somme < sommeFinale) **faire**
 8 somme ← somme + depot */* premier jour de l'année */*
 9 nbAn ← nbAn + 1
 10 interet ← (somme * TAUX) / 100
 11 somme ← somme + interet */* dernier jour de l'année */*
 12 **fantantque**

/ affichage des résultats */*

13 écrire("il faudra attendre : ", nbAn, " années et vous recevrez : ", somme,
 " euros")

Fin

On suppose que :

- le dépôt est identique chaque année
- le dépôt est effectué le premier jour de l'année
- l'intérêt est ajouté le dernier jour de l'année
- on s'intéresse à la somme possédée le dernier jour de l'année

Remarque : Normalement un algorithme ne comporte pas de numéros de lignes. Ceux qui ont été introduits dans cet exemple ne sont là que pour faciliter la simulation qui sera faite plus loin..

4. Description des éléments d'un algorithme

Un algorithme se compose de 2 parties :

- les **déclarations**
- les **instructions**

Dans chaque partie, des **commentaires** doivent faciliter la compréhension du lecteur.

4.1. Les déclarations

C'est la partie de déclaration des **constantes**, des **variables**, ...

On pourrait la comparer à la mise en place du dispositif expérimental d'une expérience de chimie : préparer la verrerie adéquate pour recevoir les solutions,

4.1.1. Variables

Une **variable** sert à mémoriser une valeur. C'est une **zone de mémoire**, physique dans un ordinateur, virtuelle dans un algorithme. Sa **valeur peut changer** au cours du temps. Pour manipuler une variable, on lui donne un nom : son **identificateur**. Lorsqu'on utilise l'identificateur d'une variable, c'est en fait pour manipuler la valeur de la variable. Un identificateur doit commencer par une lettre, il se compose de lettres, de chiffres et/ou du tiret souligné. On pourra adopter la règle suivante pour les identificateurs de variables : elles sont composées de minuscules, sauf aux ruptures de mots qui seront en majuscule (ex : sommeFinale).

Une variable est aussi caractérisée par son **type**, c'est à dire **une contrainte sur les valeurs** qu'elle peut prendre. Si on déclare une variable de type entier, on ne pourra pas y mettre un réel. Dans un algorithme il faut veiller à la compatibilité des types de variables et des valeurs qu'on met dans les variables. Dans une machine, le type d'une variable correspond à la place qui est réservée dans la mémoire ranger les futures valeurs. On conçoit qu'il faudra réserver une plus grande place pour un réel que pour un entier.

Le **rôle** (qui est un commentaire) est indispensable pour indiquer à quoi sert la variable à moins que celui-ci ne soit explicitement exprimé par l'identificateur.

4.1.2. Constantes

Une **constante** correspond à une valeur littérale. A chaque fois que la constante apparaît dans un algorithme, elle est remplacée par sa valeur. On pourra adopter la règle suivante pour les identificateurs de constantes : elles sont composées de majuscule avec un tiret bas pour les rupture de mots (ex : TAUX_ANNUEL).

4.1.3. Types

Au début on travaillera avec 4 types :

Type	en algorithmique	en machine
entier	élément de \mathbb{Z}	élément d'un sous ensemble de \mathbb{Z}
réel	élément de \mathbb{R}	élément d'un sous ensemble de \mathbb{Q}
caractère	signe, lettre, chiffre ...	Idem – délimiteur : '
chaîne	suite de caractères	Idem – délimiteur : "

Lorsqu'on veut manipuler directement des valeurs de chacun de ces types, il faut prendre certaines précautions :

- pour les entiers ou réels, pas de problème : 13, 15.78
- pour les caractères (chaînes) il peut y avoir ambiguïté : en effet si j'écris N, je désigne l'identificateur d'une variable et non le caractère donc je dois écrire 'N' pour désigner le caractère. **Les valeurs de type caractère ou chaîne doivent être entourées par des apostrophes pour les distinguer des identificateurs.**

4.2. Les instructions

Elles constituent la partie active de l'algorithme. Le chapitre suivant définira précisément ce que sont les instructions. Ici on ne présente qu'un exemple d'exécution de l'algorithme précédent. Les instructions de l'algorithme s'exécutent séquentiellement au cours du temps et modifient certaines valeurs des variables. On peut comparer une instruction à un événement qui agit et modifie l'état du système sur lequel il agit. L'état du système est constitué par l'ensemble des valeurs des variables.

Chaque numéro de ligne fait référence à la ligne de l'algorithme. Il est convenu que, si la valeur d'une variable est modifiée par l'instruction courante, la nouvelle valeur de la variable est inscrite dans la case correspondante.

L'exemple ci-dessous présente une exécution de l'algorithme pour laquelle l'utilisateur a donné

- 800 pour le dépôt
- 2600 pour la somme à obtenir

	depot	somme	sommeFinale	nbAn	interet	Ecran	Expr. bool.
1		0				Montant du dépôt annuel 800 Montant de la somme souhaitée 2600	
2				0			
3							
4	800.00						
5							
6			2600.00				
7							Vrai
8		800.00					
9				1			
10					36		
11		836.00					
12 7							
8		1636.00					Vrai
9				2		Il faudra attendre : 3 années et vous recevrez 2622.55 euros...	
10					73.62		
11		1709.62					
12 7							
8		2509.62					Vrai
9				3			
10					112.93		
11		2622.55					
12 7							Faux
13							