



Introduction

Cet ouvrage propose une approche pédagogique du langage C. Il s'appuie sur des exemples et des exercices corrigés.

L'expérience de l'enseignement montre qu'un exemple bien choisi est toujours beaucoup plus efficace qu'une longue explication. Mais l'un n'empêche pas l'autre !

L'ensemble des chapitres offrent une description complète du langage. Bien entendu, une présentation exhaustive peut "noyer" le lecteur dans une multitude de détails techniques. Pour pallier cet inconvénient, chaque chapitre souligne l'essentiel des thèmes abordés.

A qui s'adresse cet ouvrage ?

Ce livre convient aux débutants aussi bien qu'aux programmeurs confirmés. Toutefois, il est souhaitable de posséder un minimum de connaissances en informatique, comme savoir installer son ordinateur personnel.

Les encadrés

INFO

Ces rubriques attirent l'attention du lecteur sur des points intéressants.

ATTENTION

Ces rubriques mettent en garde contre des problèmes ou des erreurs susceptibles de se produire.

ASTUCE

Ces rubriques donnent des conseils ou des astuces au programmeur.



Les exemples

Les exemples donnés dans cet ouvrage sont développés en respectant le langage C normalisé. Ils sont donc exploitables dans n'importe quel environnement de programmation.

Leur objectif est double :

- Montrer immédiatement comment utiliser la syntaxe vue dans le chapitre.
- Montrer comment utiliser le concept énoncé, dans un programme plus complet. Dans ce cas, l'exemple est plus conséquent, le concept étudié n'étant qu'un élément de l'ensemble.

Cet ouvrage est conçu pour faire évoluer le lecteur vers un niveau de programmation de plus en plus complexe.

Les exemples du début sont volontairement simples et pourront sembler sans rapport les uns avec les autres. Ils permettent en réalité de constituer une collection de techniques de programmation de base sur les différentes données élémentaires (entiers, réels, caractères, etc.), qui seront bien utiles lorsque seront abordées les données structurées.

Dès que le niveau de programmation atteint est suffisant et que les grands mécanismes ont été vus (tests, boucles, sous-programmes), un exemple plus poussé est utilisé comme fil conducteur. Il est enrichi à mesure de l'apprentissage de nouvelles techniques.

Les exemples présentés respectent des conventions typographiques.

La police utilisée est à espacement fixe. Cette présentation permet de distinguer la présence ou l'absence d'espaces dans le programme, ce qui peut être important.

Chaque listing de programme est ensuite suivi par le résultat de son exécution. Les commandes saisies à ce niveau sont précédées par l'invite système (\$).

I NFO

Le caractère \$ est la forme courante de l'invite utilisée dans les systèmes UNIX. Tous les exemples ont été développés dans cet environnement, plus précisément sous AIX et Linux.

Le Listing I.1 présente un exemple de cette convention typographique.

Listing I.1 : Le premier programme

```
▼ /* premier_prog.c          */
▼ /* Mon premier programme */
▼
▼ #include <stdio.h>
▼
▼ main()
```



```
▼ {  
▼ printf("Bonjour\n") ;  
▼ }
```

Une fois le programme du Listing I.1 traduit en binaire exécutable (compilé), son exécution produit le résultat suivant :

```
▼ $ premier_prog  
▼ Bonjour
```

Les explications nécessaires à la compréhension de ce premier programme C, ainsi que les commandes de compilation (traduction en binaire exécutable) sont fournies dans les chapitres suivants. Il ne s'agit ici que d'un exemple de présentation typographique.

Les exercices corrigés

De nombreux exercices sont proposés (dont vous trouverez la solution sur le CD-ROM de l'ouvrage).

Les exemples montrent l'utilisation des techniques de programmation, mais rien ne remplace l'apprentissage par la manipulation.

Il convient de noter que chacun possède sa façon de programmer. Les erreurs des uns ne sont pas forcément celles des autres. Un bon apprentissage consiste souvent à apprendre par le biais de la compréhension de ses erreurs. Le lecteur a tout le loisir de se former au travers de ces exercices corrigés.

Les conventions de l'ouvrage

L'emploi de polices différentes permet de distinguer les syntaxes C ou de faire ressortir un nouveau terme ou un nouveau concept :

- Les instructions du langage C apparaissent dans une police à espacement fixe ; exemple : `main()`.
- Les termes nouveaux ou importants sont en italique ; exemple : *compilation*.
- Les syntaxes C nouvelles ou importantes sont en police à espacement fixe et en gras, comme la syntaxe suivante : `car=getchar() ;`.

Le contenu de l'ouvrage

L'apprentissage d'un langage consiste tout d'abord à assimiler les différents types de données (les types simples et les types structurés), puis à étudier les instructions qui les manipulent (instructions simples et composées).



L'étude des données structurées nécessite la connaissance des boucles (instructions composées). L'ordre logique qui en découle est d'étudier d'abord les types simples, puis les instructions, et en dernier les données structurées.

Cet ouvrage se divise en huit grandes parties :

- La **Partie I** porte sur les concepts fondamentaux. Elle décrit l'environnement de programmation (outils et principes) et la structure du langage. Elle regroupe les Chapitres 1 et 2.
- La **Partie II** présente les données de types simples. Elle est constituée des Chapitres 3, 4, 5, 6, 7 et 8.
- La **Partie III** décrit les instructions simples et les instructions composées. Elle contient les Chapitres 9 et 10.
- La **Partie IV** aborde les données de types structurés. Elle est constituée des Chapitres 11, 12, 13, 14 et 15.
- La **Partie V** présente les sous-programmes, la compilation séparée et les bibliothèques. Elle est constituée des Chapitres 16 et 17.
- La **Partie VI** porte sur les pointeurs et l'allocation dynamique de la mémoire. Elle est constituée des Chapitres 18 et 19.
- La **Partie VII** donne des exemples de programmation avancée, comme l'accès au système et aux répertoires. Elle présente également les instructions du pré-processeur. Les Chapitres 20 et 21 la constituent.

Voici en détail les thèmes abordés dans chacun des chapitres :

- Le **Chapitre 1** présente l'environnement de programmation. Il aborde les notions de langage binaire et de langage évolué. Il montre comment l'utilisation d'un langage évolué comme le C permet le développement de logiciels "portables" sur tous les ordinateurs. Enfin, il décrit les étapes et les outils de la conception.
- Le **Chapitre 2** porte sur la structure d'un programme C et les éléments de ce langage. Il explique les notions de variables, de constantes et d'instructions. Il décrit la syntaxe propre au langage C en présentant les caractères spéciaux et les mots réservés. Enfin, il présente quelques directives de compilation.
- Le **Chapitre 3** présente les instructions d'Entrée/Sortie liées aux types simples.
- Le **Chapitre 4** décrit le type entier. Il présente sa structure et la manipulation des données entières.
- Le **Chapitre 5** porte sur le type réel. Il décrit la représentation binaire des données, leurs limites ainsi que les opérateurs et les fonctions associés.
- Le **Chapitre 6** porte sur le type caractère. Il présente les tables de codage, les opérateurs et les fonctions usuels.
- Le **Chapitre 7** aborde le type booléen, sa représentation binaire et les deux méthodes de gestion, logique ou bits à bits.



- Le **Chapitre 8** présente le type énumération, et montre comment utiliser ce type particulier.
- Le **Chapitre 9** aborde les instructions simples.
- Le **Chapitre 10** décrit les instructions composées. Elles sont réparties en deux catégories : les instructions conditionnelles (les tests) et les instructions répétitives (les boucles).
- Le **Chapitre 11** présente les tableaux. Ce type de données permet la gestion d'une grande quantité d'informations.
- Le **Chapitre 12** présente les chaînes de caractères, les tableaux de chaînes et leur traitement.
- Le **Chapitre 13** présente les structures et les tableaux de structures. Il montre comment utiliser ces données pour gérer des informations multiples.
- Le **Chapitre 14** porte sur les unions.
- Le **Chapitre 15** aborde les fichiers texte et les fichiers binaires. Des exemples montrent comment coupler la gestion des tableaux de structures avec les fichiers.
- Le **Chapitre 16** détaille les procédures et les fonctions. Il décrit le passage de paramètres et la notion de variable locale ou globale.
- Le **Chapitre 17** complète le Chapitre 16 avec la compilation séparée et la construction de bibliothèques.
- Le **Chapitre 18** porte sur les pointeurs. Il présente les différents types de pointeurs et les fonctions d'allocation de la mémoire.
- Le **Chapitre 19** aborde la gestion dynamique de la mémoire. Il propose une solution de rechange aux tableaux, la liste chaînée d'enregistrements. Les tableaux dynamiques sont également étudiés.
- Le **Chapitre 20** décrit les outils d'interfaçage avec le système, tels que la récupération des arguments de la ligne de commande et des variables d'environnement, les fonctions de gestion des fichiers et des répertoires, le traitement de la date et de l'heure système. Il fournit également quelques exemples de programmation des signaux et des processus.
- Le **Chapitre 21** détaille les instructions du préprocesseur.