

Claude Delannoy

5^e édition

Programmer en langage C

Cours et exercices corrigés

EYROLLES

Table des matières

Table des matières	V
Avant-propos	1
1 Généralités sur le langage C	3
1 Présentation par l'exemple de quelques instructions du langage C	3
1.1 Un exemple de programme en langage C	3
1.2 Structure d'un programme en langage C	5
1.3 Déclarations	5
1.4 Pour écrire des informations : la fonction printf	6
1.5 Pour faire une répétition : l'instruction for	7
1.6 Pour lire des informations : la fonction scanf	7
1.7 Pour faire des choix : l'instruction if	8
1.8 Les directives à destination du préprocesseur	9
1.9 Un second exemple de programme	10
2 Quelques règles d'écriture	12
2.1 Les identificateurs	12
2.2 Les mots-clés	12
2.3 Les séparateurs	13
2.4 Le format libre	13
2.5 Les commentaires	14
3 Création d'un programme en langage C	15
3.1 L'édition du programme	15
3.2 La compilation	15
3.3 L'édition de liens	16
3.4 Les fichiers en-tête	16
2 Les types de base du langage C	17
1 La notion de type	17
2 Les types entiers	18
2.1 Leur représentation en mémoire	18
2.2 Les différents types d'entiers	19
2.3 Notation des constantes entières	19
3 Les types flottants	20
3.1 Les différents types et leur représentation en mémoire	20
3.2 Notation des constantes flottantes	20

4	Les types caractères	21
4.1	La notion de caractère en langage C	21
4.2	Notation des constantes caractères	22
5	Initialisation et constantes	23
6	Autres types introduits par la norme C99	24
3	Les opérateurs et les expressions en langage C	25
1	L'originalité des notions d'opérateur et d'expression en langage C	25
2	Les opérateurs arithmétiques en C	27
2.1	Présentation des opérateurs	27
2.2	Les priorités relatives des opérateurs	27
3	Les conversions implicites pouvant intervenir dans un calcul d'expression	29
3.1	Notion d'expression mixte	29
3.2	Les conversions d'ajustement de type	29
3.3	Les promotions numériques	30
3.4	Le cas du type <code>char</code>	31
4	Les opérateurs relationnels	33
5	Les opérateurs logiques	35
6	L'opérateur d'affectation ordinaire	37
6.1	Notion de lvalue	38
6.2	L'opérateur d'affectation possède une associativité de droite à gauche	38
6.3	L'affectation peut entraîner une conversion	38
7	Les opérateurs d'incrément et de décrémentation	39
7.1	Leur rôle	39
7.2	Leurs priorités	40
7.3	Leur intérêt	41
8	Les opérateurs d'affectation élargie	41
9	Les conversions forcées par une affectation	42
10	L'opérateur de cast	43
11	L'opérateur conditionnel	44
12	L'opérateur séquentiel	45
13	L'opérateur <code>sizeof</code>	47
14	Récapitulatif des priorités de tous les opérateurs	48
	Exercices	49
4	Les entrées-sorties conversationnelles	51
1	Les possibilités de la fonction <code>printf</code>	52
1.1	Les principaux codes de conversion	52
1.2	Action sur le gabarit d'affichage	52
1.3	Actions sur la précision	53
1.4	La syntaxe de <code>printf</code>	54
1.5	En cas d'erreur de programmation	55
1.6	La macro <code>putchar</code>	56

2	Les possibilités de la fonction scanf	56
2.1	Les principaux codes de conversion de scanf	57
2.2	Premières notions de tampon et de séparateurs	57
2.3	Les premières règles utilisées par scanf	57
2.4	Imposition d'un gabarit maximal	58
2.5	Rôle d'un espace dans le format	59
2.6	Cas où un caractère invalide apparaît dans une donnée	59
2.7	Arrêt prématuré de scanf	60
2.8	La syntaxe de scanf	61
2.9	Problèmes de synchronisation entre l'écran et le clavier	61
2.10	En cas d'erreur	62
2.11	La macro getchar	64
	Exercices	65
5	Les instructions de contrôle	67
1	L'instruction if	68
1.1	Blocs d'instructions	68
1.2	Syntaxe de l'instruction if	69
1.3	Exemples	69
1.4	Imbrication des instructions if	70
2	Instruction switch	72
2.1	Exemples d'introduction de l'instruction switch	72
2.2	Syntaxe de l'instruction switch	76
3	L'instruction do... while	77
3.1	Exemple d'introduction de l'instruction do... while	78
3.2	Syntaxe de l'instruction do... while	79
3.3	Exemples	80
4	L'instruction while	80
4.1	Exemple d'introduction de l'instruction while	81
4.2	Syntaxe de l'instruction while	81
5	L'instruction for	82
5.1	Exemple d'introduction de l'instruction for	82
5.2	Syntaxe de l'instruction for	84
6	Les instructions de branchement inconditionnel : break, continue et goto	86
6.1	L'instruction break	86
6.2	L'instruction continue	87
6.3	L'instruction goto	88
	Exercices	90
6	La programmation modulaire et les fonctions	93
1	La fonction : la seule sorte de module existant en C	94
2	Exemple de définition et d'utilisation d'une fonction en C	95

3	Quelques règles	97
3.1	Arguments muets et arguments effectifs	97
3.2	L'instruction <code>return</code>	98
3.3	Cas des fonctions sans valeur de retour ou sans arguments	99
3.4	Les anciennes formes de l'en-tête des fonctions	100
4	Les fonctions et leurs déclarations	101
4.1	Les différentes façons de déclarer (ou de ne pas déclarer) une fonction	101
4.2	Où placer la déclaration d'une fonction	102
4.3	À quoi sert la déclaration d'une fonction	102
5	Retour sur les fichiers en-tête	103
6	En C, les arguments sont transmis par valeur	104
7	Les variables globales	105
7.1	Exemple d'utilisation de variables globales	106
7.2	La portée des variables globales	106
7.3	La classe d'allocation des variables globales	107
8	Les variables locales	107
8.1	La portée des variables locales	108
8.2	Les variables locales automatiques	108
8.3	Les variables locales statiques	109
8.4	Le cas des fonctions récursives	110
9	La compilation séparée et ses conséquences	110
9.1	La portée d'une variable globale - la déclaration <code>extern</code>	111
9.2	Les variables globales et l'édition de liens	112
9.3	Les variables globales cachées - la déclaration <code>static</code>	112
10	Les différents types de variables, leur portée et leur classe d'allocation	113
10.1	La portée des variables	113
10.2	Les classes d'allocation des variables	113
10.3	Tableau récapitulatif	114
11	Initialisation des variables	115
11.1	Les variables de classe statique	115
11.2	Les variables de classe automatique	115
12	Les arguments variables en nombre	116
12.1	Premier exemple	116
12.2	Second exemple	118
13	Cas particulier de la fonction <code>main</code>	116
	Exercices	120
7	Les tableaux et les pointeurs	121
1	Les tableaux à un indice	121
1.1	Exemple d'utilisation d'un tableau en C	121
1.2	Quelques règles	123
2	Les tableaux à plusieurs indices	124
2.1	Leur déclaration	124
2.2	Arrangement en mémoire des tableaux à plusieurs indices	124

3	Initialisation des tableaux	125
3.1	Initialisation de tableaux à un indice	125
3.2	Initialisation de tableaux à plusieurs indices	126
4	Notion de pointeur – Les opérateurs * et &	127
4.1	Introduction	127
4.2	Quelques exemples	128
4.3	Incrémentement de pointeurs	129
5	Comment simuler une transmission par adresse avec un pointeur	130
6	Un nom de tableau est un pointeur constant	132
6.1	Cas des tableaux à un indice	132
6.2	Cas des tableaux à plusieurs indices	133
7	Les opérateurs réalisables sur des pointeurs	134
7.1	La comparaison de pointeurs	134
7.2	La soustraction de pointeurs	135
7.3	Les affectations de pointeurs et le pointeur nul	135
7.4	Les conversions de pointeurs	135
7.5	Les pointeurs génériques	136
8	Les tableaux transmis en argument	137
8.1	Cas des tableaux à un indice	137
8.2	Cas des tableaux à plusieurs indices	139
9	Utilisation de pointeurs sur des fonctions	141
9.1	Paramétrage d'appel de fonctions	141
9.2	Fonctions transmises en argument	142
	Exercices	144
8	Les chaînes de caractères	145
1	Représentation des chaînes	146
1.1	La convention adoptée	146
1.2	Cas des chaînes constantes	146
1.3	Initialisation de tableaux de caractères	147
1.4	Initialisation de tableaux de pointeurs sur des chaînes	148
2	Pour lire et écrire des chaînes	149
3	Pour fiabiliser la lecture au clavier : le couple gets sscanf	151
4	Généralités sur les fonctions portant sur des chaînes	153
4.1	Ces fonctions travaillent toujours sur des adresses	153
4.2	La fonction strlen	153
4.3	Le cas des fonctions de concaténation	154
5	Les fonctions de concaténation de chaînes	154
5.1	La fonction strcat	154
5.2	La fonction strncat	155
6	Les fonctions de comparaison de chaînes	156
7	Les fonctions de copie de chaînes	157
8	Les fonctions de recherche dans une chaîne	158

9	Les fonctions de conversion	158
9.1	Conversion d'une chaîne en valeurs numériques	158
9.2	Conversion de valeurs numériques en chaîne	159
10	Quelques précautions à prendre avec les chaînes	159
10.1	Une chaîne possède une vraie fin, mais pas de vrai début	159
10.2	Les risques de modification des chaînes constantes	160
11	Les arguments transmis à la fonction main	161
11.1	Comment passer des arguments à un programme	161
11.2	Comment récupérer ces arguments dans la fonction main	162
	Exercices	164

9 Les structures et les énumérations **165**

1	Déclaration d'une structure	166
2	Utilisation d'une structure	167
2.1	Utilisation des champs d'une structure	167
2.2	Utilisation globale d'une structure	167
2.3	Initialisations de structures	168
3	Pour simplifier la déclaration de types : définir des synonymes avec typedef	169
3.1	Exemples d'utilisation de typedef	169
3.2	Application aux structures	169
4	Imbrication de structures	170
4.1	Structure comportant des tableaux	170
4.2	Tableaux de structures	171
4.3	Structures comportant d'autres structures	172
5	À propos de la portée du modèle de structure	173
6	Transmission d'une structure en argument d'une fonction	174
6.1	Transmission de la valeur d'une structure	174
6.2	Transmission de l'adresse d'une structure : l'opérateur ->	175
7	Transmission d'une structure en valeur de retour d'une fonction	177
8	Les énumérations	177
8.1	Exemples introductifs	177
8.2	Propriétés du type énumération	178
	Exercices	180

10 Les fichiers **181**

1	Création séquentielle d'un fichier	182
2	Liste séquentielle d'un fichier	184
3	L'accès direct	185
3.1	Accès direct en lecture sur un fichier existant	186
3.2	Les possibilités de l'accès direct	187
3.3	En cas d'erreur	188

4	Les entrées-sorties formatées et les fichiers de texte	189
5	Les différentes possibilités d'ouverture d'un fichier	191
6	Les fichiers prédéfinis	192
	Exercices	193
11	La gestion dynamique de la mémoire	195
1	Les outils de base de la gestion dynamique : malloc et free	196
1.1	La fonction malloc	196
1.2	La fonction free	198
2	D'autres outils de gestion dynamique : calloc et realloc	199
2.1	La fonction calloc	199
2.2	La fonction realloc	200
3	Exemple d'application de la gestion dynamique : création d'une liste chaînée ..	200
	Exercice	203
12	Le préprocesseur	205
1	La directive #include	205
2	La directive #define	206
2.1	Définition de symboles	206
2.2	Définition de macros	208
3	La compilation conditionnelle	211
3.1	Incorporation liée à l'existence de symboles	211
3.2	Incorporation liée à la valeur d'une expression	212
13	Les possibilités du langage C proches de la machine	215
1	Compléments sur les types d'entiers	216
1.1	Rappels concernant la représentation des nombres entiers en binaire	216
1.2	Prise en compte d'un attribut de signe	217
1.3	Extension des règles de conversions	217
1.4	La notation octale ou hexadécimale des constantes	217
2	Compléments sur les types de caractères	218
2.1	Prise en compte d'un attribut de signe	218
2.2	Extension des règles de conversion	219
3	Les opérateurs de manipulation de bits	220
3.1	Présentation des opérateurs de manipulation de bits	220
3.2	Les opérateurs bit à bit	220
3.3	Les opérateurs de décalage	221
3.4	Exemples d'utilisation des opérateurs de bits	222
4	Les champs de bits	222
5	Les unions	224

Annexes	Les principales fonctions de la bibliothèque standard	227
1	Entrées-sorties (stdio.h)	228
1.1	Gestion des fichiers	228
1.2	Écriture formatée	228
	Les codes de format utilisables avec ces trois fonctions	229
1.3	Lecture formatée	231
	Règles communes à ces fonctions	232
	Les codes de format utilisés par ces fonctions	233
1.4	Entrées-sorties de caractères	234
1.5	Entrées-sorties sans formatage	236
1.6	Action sur le pointeur de fichier	236
1.7	Gestion des erreurs	237
2	Tests de caractères et conversions majuscules-minuscules (ctype.h)	237
3	Manipulation de chaînes (string.h)	238
4	Fonctions mathématiques (math.h)	239
5	Utilitaires (stdlib.h)	241
	Correction des exercices	243
	Chapitre 3	243
	Chapitre 4	244
	Chapitre 5	244
	Chapitre 6	248
	Chapitre 7	250
	Chapitre 8	252
	Chapitre 9	254
	Chapitre 10	256
	Chapitre 11	259
	Index	261