



新手学编程系列

打开C语言程序设计大门的金钥匙

新手学 C语言

王欣惠 胡艳 邵国红 等 编著

10.5

小时多媒体视频讲解

DVD-ROM

- ◎ 由浅入深：从基本概念开始讲解，逐步深入到实际开发
- ◎ 实例丰富：讲解知识点时穿插了193个实例，有较强的实用性
- ◎ 面向就业：提供了常见面试题，帮助读者了解入职面试的相关知识
- ◎ 案例典型：提供了3个取材于实际项目的案例，提高读者开发水平
- ◎ 视频教学：提供了10.5小时多媒体教学视频，学习起来更加直观



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn



新手学编程系列

新手学 C语言

王欣惠 胡艳 邵国红 等 编著

10.5

小时多媒体视频讲解



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

C 语言是当代最优秀的程序设计语言之一, 它具有效率高、可移植性强、可对硬件编程及易于调试与维护的特点。C 语言既可用于系统软件的开发, 也适合应用软件的开发。

本手册由三篇组成, 基础篇、提高篇和实例篇, 详细介绍了 C 语言的基础知识, 数据类型、运算符、表达式和输入输出函数, 选择结构设计, 循环结构设计, 数组, 函数的基本使用, 函数调用, 指针, 结构体、公用体和枚举, 位运算, 文件等内容。最后列举了 3 个完整实例, 帮助读者巩固所学内容, 方便读者模拟实践。

本手册由具有丰富教学经验的一线教师编写, 通俗易懂, 适合 C 语言的初学者、从事软件开发的程序员、大中专院校学生及社会相关培训班学员阅读。

本光盘内容为实例源代码、语音视频教学及电子教案(PPT)。

本光盘及配套手册由北京希望电子出版社独家发行, 未经出版者书面许可, 任何单位和个人不得擅自摘抄、复制光盘和本手册的部分或全部内容, 并以任何方式进行传播。

需要技术支持的读者, 请与北京清河 6 号信箱(邮编: 100085)发行部联系, 电话: 010-62978181(总机)转发行部、010-82702675(邮购), 传真: 010-82702698, E-mail: tbd@bhp.com.cn。

新手学 C 语言/王欣惠, 胡艳, 邵国红等编著. —北京希望
电子出版社, 2010

(新手学编程系列)

ISBN 978-7-89498-989-5

I. 新… II. ①王… ②胡… ③邵… III. C 语言—程序设计

责任编辑: 周凤明 / 责任校对: 马 君

责任印刷: 双 青 / 封面设计: 一品设计

北京希望电子出版社 出版

北京市海淀区上地三街 9 号金隅嘉华大厦 C 座 611

邮政编码: 100085

<http://www.bhp.com.cn>

北京四季青双青印刷厂印刷

北京希望电子出版社发行 各地新华书店经销

*

2010 年 1 月第 1 版

2010 年 1 月第 1 次印刷

印数: 1—3 000

开本: 787mm×1092mm 1/16

印张: 28

字数: 646 千字

定价: 45.00 元(配 1 张 DVD)

前言

C 语言是近年来在国内外得到迅速推广和使用的第三代程序设计语言之一，是当代最优秀的程序设计语言之一。早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们所认识，到了八十年代，C 开始进入其他操作系统，并很快在各类计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

C 语言是一种结构化语言，它层次清晰，便于按模块化的方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强，它不仅拥有丰富的运算符和数据类型，便于实现各类复杂的数据结构，它还可以直接访问内存的物理地址，进行位 (bit) 一级的操作。由于 C 语言实现了对硬件的编程操作和系统处理能力，可以直接实现对系统硬件和外部接口的控制，因此 C 语言集高级语言和低级语言的功能于一体。它既可用于系统软件的开发，也适合于应用软件的开发，著名的 UNIX 操作系统就是用 C 语言开发的。此外，C 语言还具有效率高、可移植性强等特点，因此它有广泛的应用领域。当然，C 语言不是万能的，不能取代其他的高级语言。但是，一个真正的程序设计员应该首先要学会 C 语言，有 C 语言的基础，学习其他语言也会起到事半功倍的效果。

本手册的作者是来自教学一线的有着丰富教学经验的教师。作者精心编写了本手册，目的是使编程语言的初学者在学习程序设计的道路上一帆风顺。C 语言的功能强、使用灵活、概念复杂、规则多，使用易出错，很多初学者认为学习起来非常困难。作者结合自己多年的经验，为广大读者介绍了自顶向下逐步求精的结构化程序设计技术，将难点进行分散，并由浅入深地介绍 C 语言的各个技术内容，力求使初学者学习起来更轻松，使立志于学习程序设计和 C 语言的初学者有更大的收获。

由于作者的水平有限，存在的错误希望得到广大读者和专家的指正，在适当的时机再进行修订，以便跟上计算机科学技术的发展。

特点

本手册由浅入深地讲解了 C 语言的基础知识和编程技巧，重点介绍了自顶向下的逻辑设计和模块化的结构化编程方法，并通过大量的程序示例向读者介绍了各种知识的应用方法。通过大量的实践，使读者快速的掌握所学知识。在每章的最后都安排相关的内容应用实例，更进一步强化了知识的学习。另外，每章还特别选编了经典的面试题，每个题目都进行了详细的分析，为想进入 C 语言编程领域的人员应对面试提供了有力的保障。

本手册的特点主要体现在以下几个方面。

- ❑ 内容的编排采用循序渐进的方式，适合初级、中级读者逐步掌握程序设计的基本方法，C 语言设计的基本知识和精髓。
- ❑ 精心设计结构，每一小节尽量只介绍一个知识点，使每学习一节就会有一个小的提

高，这样读者小步的上台阶，学习起来不吃力。

- 结合作者多年的教学经验和毕业生的反馈信息，深入浅出地介绍 C 语言知识，并在每章添加了面试题部分。通过精选的面试题，指出使用 C 语言时可能存在的问题和误解，并且有针对性地进行解析，是帮助新手走出困境的很好的参考，也为参加面试的人员提供了很好的参考。
- 本手册在介绍各种关键知识、技术和算法时，采用了大量浅显易懂的例子，方便读者自己进行实践和演练。在实例篇中，对于每个实例的每个算法都提供了详细分析和完整的开发示例代码，可供读者直接使用或者根据自己的实际情况进行调整。本手册中的实例、源代码都附在随书光盘中，方便读者使用。
- 本手册所选的例题适合初学者。考虑到初学者一般都没有太多的硬件知识，在例题的选用和最后的实例篇中都没有涉及到硬件知识，使一般的读者更易掌握和理解。
- 本手册课后有大量的练习题并附有答案，以便作者学习和检验。

适合的读者

- 学习程序设计的新手。
- 使用 C 语言进行程序设计的人员。
- 希望提高应用 C 语言技能的人员。
- 学习程序设计中典型算法的人员。
- 具备一定的 C 语言理论知识但是缺乏实践的程序设计人员。
- 希望了解软件设计特别是应用软件设计、开发基本思想的人员。
- 正在考虑建立基本程序设计思想的人员。
- 希望通过学习 C 语言而提高计算机高级语言应用能力的人员。
- 大中专院校和培训学校的学生。

本书第 1~7 章和第 15 章由胡艳执笔，第 8~10 和第 12~14 章由王欣惠执笔，第 11 章由邵国红执笔。其他参与编写和资料整理的人员有卜庆玲、冯曼菲、匡妍娜、雷成健、李小波、刘浩然、刘会神、马震、齐志华、舒军、孙大林、王辉、王沛、王石、王晓悦、熊英、张杰、袁福庆、赵显琼、韩延峰、李刚、张佳楠、张金霞、孔鹏、张昆、陈刚、姚志鹏等。在此一并表示感谢。封面作者署名排名不分先后，特此声明。

因编者水平有限，书中错漏之处在所难免，恳请广大读者指正，相关问题请与 martt0656@163.com 联系。

编著者

目 录

第 1 篇 基础篇

第 1 章 认识 C 语言	2
1.1 程序设计语言	2
1.1.1 程序设计语言的发展	2
1.1.2 C 语言的发展	3
1.1.3 C 语言的特点	4
1.2 结构化程序设计	5
1.2.1 程序设计的概念	5
1.2.2 算法	5
1.2.3 程序的三种基本结构	9
1.2.4 结构化程序设计方法	11
1.3 C 程序的特点	13
1.3.1 C 程序实例	13
1.3.2 C 程序的结构特点	14
1.3.3 标识符和关键字	15
1.4 C 程序的编译、连接和执行	17
1.5 编译和运行 C 程序的一般步骤	19
1.5.1 使用 Turbo C 2.0 开发	19
1.5.2 使用 Visual C++ 6.0 开发	21
1.6 常见面试题	23
1.7 小结	24
1.8 习题	24
第 2 章 数据类型、运算符、表达式和 输入输出函数	25
2.1 常量	25
2.1.1 常量的定义	26
2.1.2 符号常量	26
2.2 变量	27
2.2.1 变量定义	27
2.2.2 变量命名	27
2.2.3 变量的赋值和初始化	28
2.2.4 变量的作用域	28
2.2.5 使用 extern 声明变量	30

2.3 数据类型	30
2.3.1 整型	31
2.3.2 实型	32
2.3.3 字符型	34
2.4 运算符和表达式	36
2.4.1 运算符和表达式	36
2.4.2 赋值运算符和赋值表达式	37
2.4.3 算术运算符和算术表达式	38
2.4.4 关系运算符和关系表达式	39
2.4.5 逻辑运算符和逻辑表达式	40
2.4.6 条件运算符和条件表达式	42
2.4.7 逗号运算符和逗号表达式	43
2.4.8 运算符的优先级	43
2.5 不同类型数据之间的转换	43
2.5.1 隐式转换	44
2.5.2 显示转换	44
2.6 数据的输入	46
2.6.1 使用 scanf 函数实现数据输入	46
2.6.2 使用 getchar 函数实现字符输入	48
2.7 数据的输出	50
2.7.1 使用 printf 函数实现数据 格式输出	50
2.7.2 使用 putchar 函数实现字符输出	52
2.8 顺序结构程序设计举例	53
2.9 常见面试题	54
2.10 小结	55
2.11 习题	55
第 3 章 选择结构设计	57
3.1 选择结构简介	57
3.2 使用 if 实现选择结构设计	57
3.2.1 if 结构语法	58
3.2.2 if 结构程序举例	58
3.3 使用 if-else 实现选择结构设计	60

3.3.1 if-else 结构语法.....	60	4.8 循环结构程序设计举例.....	98
3.3.2 if-else 结构程序举例.....	61	4.9 常见面试题.....	101
3.4 使用多重 if 实现选择结构设计.....	62	4.10 小结.....	102
3.4.1 多重 if 结构语法.....	62	4.11 习题.....	102
3.4.2 多重 if 结构程序举例.....	63	第 5 章 数组.....	104
3.5 使用 if 嵌套实现选择结构设计.....	64	5.1 数组简介.....	104
3.5.1 if 嵌套结构语法.....	64	5.2 一维数组.....	105
3.5.2 if 和 else 配对问题.....	65	5.2.1 一维数组的定义.....	105
3.5.3 if 嵌套结构程序举例.....	65	5.2.2 一维数组的存储.....	105
3.6 使用 switch 实现选择结构设计.....	66	5.2.3 一维数组的初始化.....	105
3.6.1 switch 结构语法.....	66	5.2.4 一维数组元素的引用.....	106
3.6.2 switch 语句的注意事项.....	68	5.2.5 一维数组使用举例.....	106
3.6.3 switch 的执行过程.....	68	5.3 二维数组.....	108
3.6.4 switch 结构程序举例.....	69	5.3.1 二维数组的定义.....	108
3.7 switch 和多重 if 的比较.....	71	5.3.2 二维数组的存储.....	108
3.8 选择结构程序设计举例.....	73	5.3.3 二维数组的初始化.....	109
3.9 常见面试题.....	77	5.3.4 二维数组元素的引用.....	110
3.10 小结.....	78	5.3.5 二维数组使用举例.....	110
3.11 习题.....	78	5.4 多维数组.....	112
第 4 章 循环结构设计.....	80	5.4.1 多维数组的定义.....	112
4.1 循环结构简介.....	80	5.4.2 多维数组的存储.....	112
4.2 使用 while 循环.....	81	5.4.3 多维数组的初始化.....	112
4.2.1 while 循环语法及含义.....	81	5.4.4 多维数组元素的引用.....	113
4.2.2 使用 while 循环程序举例.....	82	5.5 字符数组和字符串.....	113
4.3 使用 do-while 循环.....	84	5.5.1 字符数组定义.....	113
4.3.1 do-while 循环语法及含义.....	84	5.5.2 字符数组初始化.....	114
4.3.2 使用 do-while 循环程序举例.....	85	5.5.3 字符数组引用.....	115
4.4 使用 for 循环.....	87	5.5.4 输入字符串.....	116
4.4.1 for 循环语法及含义.....	87	5.5.5 输出字符串.....	117
4.4.2 for 循环的注意事项.....	88	5.5.6 常用字符串处理函数.....	118
4.4.3 使用 for 循环程序举例.....	89	5.5.7 字符串使用举例.....	126
4.5 循环嵌套.....	91	5.6 数组应用举例.....	127
4.5.1 循环嵌套原则.....	91	5.7 常见面试题.....	131
4.5.2 使用循环嵌套程序举例.....	92	5.8 小结.....	132
4.6 三种循环的比较.....	93	5.9 习题.....	132
4.7 跳转语句.....	95	第 2 篇 提高篇.....	
4.7.1 使用 break 语句实现程序跳转.....	95	第 6 章 函数的基本使用.....	136
4.7.2 使用 continue 语句实现程序跳转.....	96	6.1 函数简介.....	136
4.7.3 使用 goto 语句实现程序跳转.....	97		

6.2 函数的定义	137	7.2.3 使用多维数组作函数参数	179
6.3 函数的参数和返回值	139	7.3 函数的嵌套调用	181
6.3.1 函数的形参和实参	139	7.3.1 函数嵌套调用的概念	181
6.3.2 函数参数的注意事项	140	7.3.2 函数嵌套调用举例	181
6.3.3 函数的返回值	141	7.4 函数的递归调用	182
6.4 调用函数	143	7.4.1 函数递归调用概念	182
6.4.1 调用函数的形式	143	7.4.2 函数递归调用举例	184
6.4.2 调用函数的方法	144	7.5 外部函数和内部函数	186
6.4.3 调用函数的注意事项	146	7.5.1 使用外部函数	186
6.4.4 函数声明	146	7.5.2 使用内部函数	187
6.5 函数之间的参数传递方式	148	7.6 函数调用应用举例	188
6.5.1 使用传值方式	148	7.7 常见面试题	191
6.5.2 使用传地址方式	149	7.8 小结	192
6.6 生存期和作用域	150	7.9 习题	192
6.6.1 变量的生存期	150	第8章 指针	194
6.6.2 变量的作用域	152	8.1 指针简介	194
6.7 局部变量和全局变量	153	8.1.1 直接访问内存	195
6.7.1 使用局部变量	154	8.1.2 间接访问内存	195
6.7.2 使用全局变量	155	8.2 指针的定义和引用	196
6.7.3 变量的屏蔽	157	8.2.1 指针的概念	196
6.8 变量的存储类型和作用域	157	8.2.2 指针变量的定义	196
6.8.1 变量的存储类型	158	8.2.3 指针运算符	197
6.8.2 使用自动型变量	159	8.2.4 指针的引用	197
6.8.3 使用静态变量	160	8.3 指针的运算	200
6.8.4 使用外部变量	163	8.3.1 指针的赋值运算	200
6.8.5 使用寄存器变量	164	8.3.2 指针的算术运算	201
6.8.6 存储类型小结	165	8.3.3 指针的关系运算	203
6.9 函数的基本使用举例	165	8.4 数组的指针	204
6.10 常见面试题	170	8.4.1 指向数组元素的指针变量的 定义和赋值	204
6.11 小结	171	8.4.2 使用指针引用数组元素	205
6.12 习题	172	8.4.3 使用指向数组的指针作为 函数参数	208
第7章 函数调用	173	8.4.4 使用指向多维数组的指针变量	211
7.1 使用 main 函数	173	8.5 指针数组	214
7.1.1 main 函数的定义	173	8.5.1 指针数组的概念	215
7.1.2 main 函数的类型	174	8.5.2 使用指针数组	215
7.1.3 main 函数传递参数	175	8.6 指针和字符串	217
7.2 数组与函数	176	8.6.1 字符指针	217
7.2.1 使用数组元素作函数实参	176		
7.2.2 使用数组名作函数参数	177		

8.6.2 使用字符指针	218	9.6.5 链表的查询	267
8.6.3 使用字符串指针作函数参数	220	9.6.6 插入新结点	268
8.6.4 字符串指针和字符数组的 区别与联系	224	9.6.7 链表的删除	269
8.7 函数和指针	226	9.7 结构体应用举例	271
8.7.1 使用函数指针调用函数	226	9.8 共用体	276
8.7.2 使用指向函数的指针作 函数的参数	227	9.8.1 共用体的概念	276
8.8 指向指针的指针	229	9.8.2 共用体的声明	276
8.8.1 指向指针的指针的定义	229	9.8.3 共用体变量成员的引用	279
8.8.2 使用指向指针的指针	230	9.8.4 使用共用体应注意的问题	280
8.9 指针应用举例	232	9.9 枚举	282
8.10 常见面试题	234	9.9.1 枚举类型的定义	282
8.11 小结	235	9.9.2 枚举类型变量的声明	282
8.12 习题	236	9.9.3 枚举类型变量的引用	283
第9章 结构体、共用体和枚举	239	9.9.4 使用枚举应注意的问题	284
9.1 结构体的概念和定义	239	9.10 自定义类型	286
9.1.1 结构体概念	239	9.11 常见面试题	287
9.1.2 结构体的定义	240	9.12 小结	289
9.2 结构体变量的定义、引用和初始化	241	9.13 习题	292
9.2.1 结构体变量的定义	241	第10章 位运算	296
9.2.2 结构体变量的引用	243	10.1 位运算简介	296
9.2.2 结构体变量的初始化	245	10.1.1 字节和位	296
9.3 结构体数组	247	10.1.2 原码的运算	297
9.3.1 结构体数组定义	247	10.1.3 反码的运算	297
9.3.2 结构体数组初始化	249	10.1.4 补码的运算	298
9.3.3 结构体数组引用	250	10.2 使用位运算符	299
9.4 指向结构体类型的指针	253	10.2.1 按位与运算符及其运算	299
9.4.1 指向结构体类型变量的指针	253	10.2.2 按位或运算符及其运算	300
9.4.2 指向结构体类型数组的指针	254	10.2.3 按位异或运算符及其运算	301
9.5 结构体变量和函数	256	10.2.4 按位取反运算符及其运算	302
9.5.1 结构体变量作函数参数	256	10.2.5 左移运算符及其运算	302
9.5.2 结构体变量作函数的返回值	258	10.2.6 右移运算符及其运算	303
9.6 动态存储分配	260	10.3 位运算举例	304
9.6.1 动态存储分配概念	260	10.4 位段	306
9.6.2 链表的概念	260	10.4.1 位段的概念和定义	306
9.6.3 动态存储分配函数——malloc 函数和 free 函数	262	10.4.2 位段的引用	308
9.6.4 链表的创建	265	10.5 常见面试题	309
		10.6 小结	309
		10.7 习题	310
		第11章 编译预处理	312

11.1 “文件包含”语句.....	312	12.4.2 使用 fseek 函数实现文件 位置指针随机移动.....	346
11.1.1 文件包含形式	312	12.5 常见面试题.....	348
11.1.2 使用文件包含	313	12.6 小结.....	349
11.1.3 使用文件包含注意事项.....	314	12.7 习题.....	350
11.2 宏定义	314	第 3 篇 实例篇	
11.2.1 符号常量的定义	314	第 13 章 24 点扑克牌游戏.....	354
11.2.2 不带参数的宏定义	315	13.1 游戏描述.....	354
11.2.3 带参数的宏定义	317	13.2 游戏解析.....	354
11.2.4 使用宏定义的注意事项.....	319	13.2.1 处理用户输入的表达式.....	354
11.3 条件编译.....	320	13.2.2 对后缀表达式求值.....	356
11.3.1 使用#ifndef 形式实现条件编译....	320	13.3 两个重要函数的实现.....	356
11.3.2 使用#ifdef 形式实现 条件编译	321	13.4 “24 点扑克牌游戏”的实现	359
11.3.3 使用#if 形式实现条件编译	321	13.4.1 栈的基本操作函数的实现	360
11.4 常见面试题.....	322	13.4.2 随机发四张扑克牌函数的实现....	361
11.5 小结	322	13.4.3 检查输入表达式正确性 的函数的实现.....	362
11.6 习题	322	13.4.4 main 函数的实现.....	362
第 12 章 文件	327	13.5 小结.....	364
12.1 文件概述	327	第 14 章 学生成绩管理系统.....	365
12.1.1 C 语言文件.....	327	14.1 需求分析.....	365
12.1.2 文件类型指针	329	14.2 系统设计.....	365
12.1.3 文件处理过程	330	14.3 系统基本设计思路.....	366
12.2 打开和关闭文件.....	330	14.3.1 系统设计方法.....	366
12.2.1 使用 fopen 函数实现文件打开... 330		14.3.2 程序基本流程.....	366
12.2.2 使用 fclose 函数实现文件关闭... 332		14.3.3 程序流程的代码实现.....	367
12.3 读写文件	333	14.4 各主要功能模块的实现.....	369
12.3.1 使用 fgetc/fputc 函数实现 字符输入/输出	333	14.4.1 从指定的磁盘文件中读入 学生信息.....	369
12.3.2 使用 fgets/fputs 实现字符串 输入/输出	336	14.4.2 将数据存入磁盘文件.....	370
12.3.3 使用 fscanf/fprintf 函数实现将 格式化数据输入/输出文件	339	14.4.3 打印所有同学的成绩信息.....	371
12.3.4 使用 fread/fwrite 函数实现将 数据块输入/输出文件	341	14.4.4 排序学生成绩并输出.....	372
12.3.5 使用 feof/ferror/clearerr 函数 实现文件状态检查	343	14.4.5 查询学生成绩信息.....	373
12.4 文件定位	345	14.4.6 增加学生成绩.....	375
12.4.1 使用 rewind 函数实现位置 指针移动到文件头	345	14.4.7 删除学生成绩.....	376
		14.5 小结.....	377
		第 15 章 “俄罗斯方块”游戏的实现	378
		15.1 游戏描述.....	378

15.2 游戏解析	379
15.2.1 方块形状的形成	379
15.2.2 控制方块运动的函数	381
15.2.3 游戏画面绘制	381
15.3 俄罗斯方块的程序实现	382
15.3.1 准备工作	382
15.3.2 游戏初始化	385
15.3.3 绘制游戏画面	386
15.3.4 切换游戏级别	391
15.3.5 计算游戏得分	392
15.3.6 方块移动控制	393
15.3.7 方块的消除	395
15.3.8 游戏开始结束	399
15.3.9 main 函数	399
15.3.10 其他函数	401
15.4 小结	403
附录	404
附录 A Turbo C 中常见错误提示的 中英文对照	404
附录 B ASCII 编码	407
附录 C Turbo C 中各种数据类型	411
附录 D 函数间传递参数时类型的 自动转换	411
附录 E 主要库函数的介绍	411
附录 F C 标准库介绍	413
F.1 诊断<assert.h>	414
F.2 字符类别测试<ctype.h>	414
F.3 错误处理<errno.h>	415
F.4 浮点算术运算常量<float.h>	415
F.5 整型常量<limits.h>	416
F.6 地域环境<locale.h>	416
F.7 数学函数<math.h>	417
F.8 非局部跳转<setjmp.h>	418
F.9 信号<signal.h>	419

F.10 可变参数表<stdarg.h>	421
F.11 公共定义<stddef.h>	421
F.12 输入输出<stdio.h>	422
F.13 实用函数<stdlib.h>	423
F.14 字符串函数<string.h>	424
F.15 日期与时间函数<time.h>	425
附录 G C 语言专用字符函数	425
G.1 isalnum 检查字符是否是 字母或数字	426
G.2 isalpha 检查字符是否是字母	427
G.3 isascii 检查字符是否是 ASCII 码	427
G.4 iscntrl 检查字符是否是控制字符	428
G.5 isdigit 检查字符是否是数字字符	429
G.6 isxdigit 检查字符是否是 十六进制数字字符	430
G.7 isgraph 检查字符是否是可 打印字符 (不含空格)	430
G.8 isprint 检查字符是否是可 打印字符 (含空格)	431
G.9 ispunct 检查字符是否是 标点字符	432
G.10 islower 检查字符是否是 小写字母	432
G.11 isupper 检查字符是否是 大写字母	433
G.12 isspace 检查字符是否是空格符	434
G.13 toascii 将字符转换为 ASCII 码	434
G.14 tolower 将大写字母转换为 小写字母	435
G.15 toupper 将小写字母转换为 大写字母	436

第 1 篇 基础篇

第 1 章 认识 C 语言

第 2 章 数据类型、运算符、表达式和输入输出函数

第 3 章 选择结构设计

第 4 章 循环结构设计

第 5 章 数组

第 1 章 认识 C 语言

自 1946 年世界上第一台电子计算机问世以来，计算机科学技术的发展十分迅猛，计算机被广泛应用于人类的生产和生活的各个领域，推动了社会的进步与发展。计算机是由硬件系统和软件系统两大部分组成的，硬件是物质基础，而软件就是计算机的灵魂，没有软件的计算机仅仅是一台“机器”，是什么也不能干的，有了软件它才能灵动起来，成为一台真正的“电脑”。所有的软件，都是用计算机语言，也就是程序设计语言编写的。作为程序设计语言中的一颗璀璨明珠——C 语言，也就是本手册所要向大家讲述的，希望大家通过学习能很好地掌握的这门语言。

本章讲述的知识点包括：

- 算法。
- C 语言程序的 3 种基本结构。
- C 语言程序的特点。
- 使用 Turbo C 2.0 开发 C 语言程序。

1.1 程序设计语言

程序设计语言是人和计算机交流信息的工具，是软件的重要组成部分。本节将简要介绍程序设计语言的发展历史、结构化程序设计的基本要素以及 C 语言的特点和使用方法，为以后各章的学习奠定基础。

1.1.1 程序设计语言的发展

计算机语言也称程序设计语言（Program Language），即编写计算机程序所用的语言。计算机语言可分为：机器语言、汇编语言和高级语言。高级语言是接近于人类习惯使用的自然语言和数学语言的计算机程序设计语言。它独立于计算机，用户可以不了解机器指令，也可以不必了解机器的内部结构和工作原理，就能用高级语言编写程序。高级语言的通用性好、易学习、易使用、不受机器型号的限制，而且易于交流和推广。

早期的电脑都是直接采用机器语言编写程序的，即用“0”和“1”为指令代码来编写程序，这种程序读写都很困难，编程的效率极低。为了方便编程，随即出现了汇编语言，使用汇编语言编程虽然提高了效率，但所编的程序仍然不够直观简便。从 1954 年起，电脑界逐步开发了一批像 FORTRAN、COBOL 和 ALGOL 的最初的“高级语言”，这些高级语言采用英文词汇、符号和数字，遵照一定的规则来编写程序。

到 20 世纪 60 年代中期，美国达特默斯学院的约翰·凯梅尼（J.Kemeny）和托马斯·卡茨（T.Kurtz）认为，像 FORTRAN 那样的语言都是为专业人员设计的，没有办法普及。于是，

他们在简化 FORTRAN 的基础上，研发出一种“初学者通用符号指令代码”（Beginner's All purpose Symbolic Intruction Code），简称 BASIC 语言。由于 BASIC 语言易学易用，很快就成为最流行的电脑语言之一，几乎所有小型电脑和个人电脑都在使用它。经过不断改进后，它一直沿用至今，出现了像 QBASIC、VB 等新一代的 BASIC 版本。之后到 1971 年，瑞士联邦技术学院的尼克劳斯·沃尔斯（N.Wirth）教授发明了另一种简单明晰的电脑语言，这就是以帕斯卡的名字命名的 PASCAL 语言。PASCAL 语言语法严谨，层次分明，程序易写，具有很强的可读性，也是第一个结构化的编程语言。

1983 年度的图林奖则授予了 AT&T 贝尔实验室的两位科学家邓尼斯·里奇（D.Ritchie）和他的协作者肯·汤姆森（K.Thompson），以表彰他们共同开发的著名电脑语言 C。C 语言的设计理念是“Keep It Simple, Stupid”，因而程序员可以轻易掌握整个 C 语言的逻辑结构，而不用一天到晚翻手册写代码。于是，众多的程序员倒向了 C 语言的怀抱，C 语言迅速并广泛地传播开来。C 语言现在是当今软件工程师最宠爱的语言之一。

1.1.2 C 语言的发展

C 语言最早可以追溯到 ALGOL60。ALGOL60 是面向问题的高级语言，产生于 20 世纪 60 年代，但由于其自身的局限性，它并不适用于编写系统程序，于是，1963 年剑桥大学推出了 CPL 语言，虽然解决了一些问题，比 ALGOL60 更加接近硬件了，但是其规模较大，难以实现。于是剑桥大学继续改写 CPL，于 1967 年推出了 BCPL 语言，它主要是对 CPL 语言进行了简化。

至 1970 年，英国的贝尔实验室进一步简化了 BCPL，并以 BCPL 的第一个字母命名了该语言，但由于 B 语言太过简单了，导致功能十分有限。1972 年英国贝尔实验室的 D.Ritchie 一个人完成了 C 语言的定义。C 语言不仅保持了 B 语言的精练、更接近硬件的特点之外，还克服了 B 语言过于简单、功能不足的缺点。1973 年 K.Thompson 和 D.Ritchie 合作，利用 C 语言改写了 UNIX，奠定了操作系统的基础，更在 1978 年美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。随后 C 语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的应用，成为当代最优秀的程序设计语言之一。

随着微型计算机的普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准研究所（ANSI）为 C 语言制定了一套 ANSI 标准，这也成了现行的 C 语言标准版本。

如今，在微机上，有许多性能良好的商品 C 语言系统可用。包括 Borland 公司早期的 Turbo C 和后续的 Borland C/C++ 系列产品；Microsoft（微软）公司的 Microsoft C 和后续的 Visual C/C++ 系列产品等。各种工作站系统大都采用 UNIX 和 Linux，C 语言是它们的标准系统开发语言。

注意：除了上面介绍的产品外，还有很多廉价或免费的 C 语言系统。其他微型机也有多种 C 语言系统。各种大型计算机上也有自己的 C 语言系统。

1.1.3 C 语言的特点

C 语言是一种通用的、面向过程的程序语言。它具有高效、灵活、功能丰富、表达力强和较高的移植性等特点，在程序员中备受青睐。下面只介绍了 C 语言最容易理解的一般特点，由于 C 语言的这些优点，使 C 语言应用面很广。

1. C 语言数据类型丰富，运算符方便

C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。这些数据类型能用来实现各种复杂的数据结构（如链表、树、栈等）的运算，尤其是指针类型数据，使用起来比 PASCAL 更为灵活、多样；C 语言的运算符包含的范围很广，共有 34 种运算符。C 把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

2. 语言简洁、紧凑，使用方便、灵活

C 语言的一个语句可完成多项操作。一个表达式也可以完成多项操作。书写简练，源程序短，因而输入程序工作量小。

3. 面向结构化程序设计的语言

结构化语言的显著特点是代码、数据的模块化，C 程序是以函数形式提供给用户的，这些函数调用都很方便。C 语言具有多种条件语句和循环控制语句（如 if/else 语句、switch 语句、while 语句、do/while 语句、for 语句、break 语句和 continue 语句等），程序完全结构化。

4. C 语言能进行位（bit）操作

C 语言能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此，C 语言既具有高级语言的功能，又具有低级语言的许多功能，可用来编写系统软件，也可用来编写应用软件。C 语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言。有人把 C 称为“高级语言中的低级语言”或“中级语言”，意为兼有高级语言和低级语言的特点。

5. 生成目标代码质量高，程序执行效率高

C 程序一般只比汇编程序生成的目标代码的执行效率低 10%~20%，却比其他高级语言的执行效率高。

6. 移植性好

C 语言的移植性好。主要表现在只要对这种语言稍加修改，便可以适应不同型号的机器或各类操作系统。

注意：许多大型软件都是用 C 语言编写的，这主要是由于 C 的可移植性好和硬件控制能力高，表达和运算能力强。许多以前只能用汇编语言处理的问题，现在可以改用 C 语言来处理了。读者现在也许还不能深刻理解这些特点，待学习完本手册以后再回顾一下，就会有比较深刻的体会。

1.2 结构化程序设计

结构化程序设计是一个程序设计人员应具备的基本知识。要想实现一个完整合理的计算机程序，就应当采用结构化的程序设计方法进行程序设计，并且用计算机语言表示出来。因此，语言本身只是工具，要想很好地掌握C语言，首先就要对结构化程序设计进行深入的学习和理解。本节将介绍结构化程序设计的概念和方法，以及结构化程序设计的灵魂要素——算法。

1.2.1 程序设计的概念

什么是程序设计？程序设计就从著名计算机科学家沃思（Nikiklaus Wirth）提出的一个公式说起：

程序设计=数据结构+算法

数据结构即非数值计算程序设计问题中的计算机操作对象，以及它们之间的关系和操作。算法是对特定问题求解步骤的一种描述，是对指令的有序序列。形象地说，程序设计就像盖房子，数据结构就像砖和瓦，而算法就是设计图纸。若想盖好房子，首先必须有原料（数据结构），但是这些原料不会自动地盖起想要的房子；要做的是必须按照设计图纸（算法）上的说明一砖一瓦地砌起来；这样才能拥有梦想中的房子。程序设计也一样，在我们使用的编译工具中，有各种功能语句或基本结构，它们不会自动排列成需要的程序代码。你得按照程序规定的功能去编写，而程序功能的实现就是算法的具体体现。通俗地说就是“必须按照特定的规则，把特定的功能语句和基本结构按照特定的顺序排列起来，形成一个有特定功能的程序”。

数据结构是程序设计这座大厦的基础，没有基础，无论设计多么高明，这座大厦都不可能建造起来。算法则是程序设计的思想，是它的灵魂，没有灵魂的程序不能叫程序，只是一堆杂乱无章的符号而已。程序设计的基本目标是用算法对问题的数据进行处理，从而获得所期望的效果。

也就是说，数据结构和算法是一个程序设计人员所应具备的基本知识。算法是灵魂，数据结构是加工对象，算法是解决“做什么”和“怎么做”的工具。程序中的操作语句，实际上就是算法的体现。显然，不了解算法就谈不上程序设计。由于算法的重要性，本节就先来介绍算法的初步知识。

1.2.2 算法

做任何事情都要有一定的步骤。这些步骤都是按照一定的顺序进行的，缺一不可，顺序乱了错了也不行。算法就是程序设计的基本思想方法和步骤。

1. 算法概念

算法是对要解决一个问题或要完成一项任务所采取的方法和步骤的描述，包括需要什么数据（输入什么数据、输出什么结果）、采用什么结构、使用什么语句以及如何安排这些语

句等。

2. 算法实例

下面将举两个简单的算法实例。

【示例 1.1】演示两个变量交换数据的过程。

已知变量 x 和 y 中分别存放了数据，现在要交换其中的数据。为了达到交换的目的，需要引进一个中间变量 m ，其算法如下：

- 1 将 x 中的数据送给变量 m ，即 $x \rightarrow m$ ；
- 2 将 y 中的数据送给变量 x ，即 $y \rightarrow x$ ；
- 3 将 m 中的数据送给变量 y ，即 $m \rightarrow y$ ；

【示例 1.2】演示输入三个不相同的数，求出其中的最小数的过程。

先设置一个变量 $\min$ ，用于存放最小数。当输入 a 、 b 、 c 三个不相同的数后，先将 a 与 b 进行比较，把小者送给变量 $\min$ ，再把 c 与 $\min$ 进行比较，若 $c < \min$ ，则将 c 的数值送给 $\min$ ，最后 $\min$ 中就是三个数中的最小数，具体算法如下：

- 1 若 $a < b$ ，则 $a \rightarrow \min$ ，否则 $b \rightarrow \min$ ；
- 2 再将 c 与 $\min$ 进行比较，若 $c < \min$ ，则 $c \rightarrow \min$ 。这样， $\min$ 中存放的即是三个数中的最小数；

【示例 1.3】演示输入两个正整数 a 和 b ($a > b$)，求它们的最大公约数的过程。

求两个正整数 a 、 b ($a > b$) 的最大公约数，可以归结为求数列：

$a, b, r_1, r_2, \dots, r_{m-1}, m, m+1, 0$

此数列的首项与第二项是 a 和 b ，从第三项开始的各项，分别是前两项相除所得的余数，如果余数为 0，它的前项 $m+1$ 即是 a 和 b 的最大公约数，这种方法叫做欧几里德辗转相除法，其算法如下：

- 1 输入 a, b ($a > b$)；
- 2 求 a/b 的余数 r ；
- 3 如果 $r \neq 0$ ，则将 $b \rightarrow a$ ， $r \rightarrow b$ ，再次求 a/b 的余数 r ，转至③；
- 4 输出最大公约数 b ；

3. 算法特性

一个算法应该具有以下五个重要特征。

(1) 有穷性。一个算法必须保证执行有限步之后结束，而不是无穷的。在示例 1.3 中，如果 r 的值永远都不等于 0，则循环永远都不会停止，这就不是有穷的步骤。

(2) 确切性。算法的每一个步骤必须有确切的定义，不应当是含糊的，模棱两可的。在示例 1.1 中，如果说将 x 中的数据送给一个变量，这就是不确切，计算机无法执行，因为不知道要将数据送给哪个变量，所以要指明变量 m 。

(3) 输入。一个算法有 0 个或多个输入，以描述运算对象的初始情况，所谓 0 个输入是指算法本身就已经确定了初始条件。如例 1.2 和例 1.3 分别需要 3 个和 2 个输入。也有的算法可以不要任何输入，如计算从 1 到 100 的和就是 0 个输入。

(4) 输出。一个算法有一个或多个输出，以反映对输入数据加工后的结果。示例 1.1、示例 1.2 和例 1.3 分别要求输出交换后的数据、三个数中的最小值和两个数的最大公约数。没有输出的算法是毫无意义的。当然，算法的输出不一定都需要计算机打印输出。

(5) 有效性。算法中的每一个步骤都应当有效地执行，并能得到确定的结果。例如，