

□ 21世纪高等学校规划教材

案例式C语言 程序设计

蔡庆华 主编

吴长勤 王一宾 马丽生 副主编



高等教育出版社
HIGHER EDUCATION PRESS

21 世纪高等学校规划教材

案例式 C 语言程序设计

Anlishi C Yuyan Chengxu Sheji

蔡庆华 主编

吴长勤 王一宾 马丽生 副主编

 高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书是一本趣味性和实用性兼有的 C 语言程序设计教材。全书由 13 章组成, 内容包含 C 语言概述、基本数据类型、输入输出函数、选择结构、循环结构、函数、数组、指针、字符串、复杂数据类型、位运算、文件操作和综合程序设计。

本书以程序设计为主线, 以编程应用为驱动, 通过案例和问题引入内容, 重点讲解程序设计的思想、方法和相关 C 语言知识。通过案例循序渐进地引出知识点, 形成逻辑清晰的脉络和主线, 案例内容紧密结合实践, 举一反三, 融会贯通。本教材配有多媒体课件、例题和习题源代码等教学资源, 可免费向任课教师提供。

本书可作为高等学校各专业的 C 语言程序设计课程教材、计算机等级考试的教学用书, 也可作为对 C 语言程序设计感兴趣的读者的自学用书。

图书在版编目 (C I P) 数据

案例式 C 语言程序设计/蔡庆华主编. --北京: 高等教育出版社, 2012. 2

ISBN 978 - 7 - 04 - 034135 - 5

I. ①案… II. ①蔡… III. ①C 语言 - 程序设计 - 高等学校 - 教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 000073 号

策划编辑	刘茜	责任编辑	时阳	封面设计	杨立新	版式设计	余杨
插图绘制	尹莉	责任校对	俞声佳	责任印制	张福涛		

出版发行	高等教育出版社	咨询电话	400 - 810 - 0598
社 址	北京市西城区德外大街 4 号	网 址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印 刷	北京奥鑫印刷厂	网上订购	http://www.landaco.com
开 本	787mm × 1092mm 1/16		http://www.landaco.com.cn
印 张	18.25	版 次	2012 年 2 月第 1 版
字 数	440 千字	印 次	2012 年 2 月第 1 次印刷
购书热线	010 - 58581118	定 价	25.10 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 34135 - 00

前 言

程序设计是高等学校重要的计算机基础课程，它以编程语言为平台，介绍程序设计的思想和方法。学习程序设计本身是一件充满挑战也充满乐趣的事情，然而它常常给人以枯燥乏味的感觉，本书力图用最简洁的语言、最典型的实例，以通俗的解释将这种趣味挖掘出来，带给读者以全新的体验，一起去体会程序设计的无穷乐趣。

C 语言是目前国内外最广泛流行的通用程序设计语言之一，其功能强大、使用方便、程序代码效率高，应用面极广，已经成为高校各类型本专科学生和程序设计人员必修的课程，并逐步被许多高校列为程序设计课程的首选程序语言。

C 语言程序设计是一门实践性很强的课程，该课程的学习有其自身的特点，听不会，看不会，只能练会。学习者必须通过大量的编程训练，在实践中掌握 C 语言的相关知识，逐步理解和掌握程序设计的思想和方法。本书以具体应用为背景，面向工程实践和编程能力训练，从实际问题出发，在一个个案例中引出相关知识点，通过类比、模拟使读者很容易解决类似问题。

全书以程序设计为主线，以编程应用为驱动，通过案例和问题引入内容，重点讲解程序设计的思想和方法，并穿插相关的语言知识。全书分为 13 章，内容主要包括 C 语言概述、基本数据类型、输入输出函数、选择结构、循环结构、函数、数组、指针、字符串、复杂数据类型、位运算、文件操作和综合程序设计等。

本书对于 C 语言语法的介绍力求简练，用层次渐进的实例说明问题。对于学生感到较为困难的常用算法也进行了细致的介绍。内容上吸取多年的教学经验，尝试将字符串的相关内容专门形成一章进行系统介绍。

在内容编排上，全书内容注重教材的易用性。每章开头都有内容导读，指导读者阅读；每章结尾均给出本章知识点，帮助读者整理思路。为了提高读者的学习兴趣，对语言知识的介绍一般通过实例引入，每节的第一个内容均是程序解析，从而导出本节要学习的知识点。为了鼓励读者多思考，多练习，提高综合能力，书中设计了很多练习题、思考题，均放在相关知识点和例题之后，做到每节有练习，每章有习题，力求做到内容有宽度、有深度、有高度。当然，为避免内容的重复，每章的课后习题将放在配套教材中。

与本书配套的教材为《C 语言程序设计案例教程实验与习题指导》。书中搜集了许多有代表性的习题并提供了详细的解答，对 C 语言上机实验的一般方法和注意事项进行了介绍，并按照章节的顺序和进度每章安排一至两个实验，每个实验包括精心设计的程序阅读实例、程序填空、程序改错和编程题。读者可以先模拟程序阅读实例，然后再做改错题和填空题，通过模仿来解决编程问题。

与本书配套的还有完整的电子教案、课件、源程序代码和参考资料，可在“计算机基础”精品课程网站（<http://www.aqtc.edu.cn>）下载或通过电子邮箱 caiqh74@aqtc.edu.cn、caiqh74@163.com 索取。

本书由蔡庆华主编并统稿。参编人员都是长期从事“C 语言程序设计”课程教学的教师。

II 前言

他们在长期的教学工作中积累了丰富的经验，并且主编、参编过包括 C 语言教材在内的多本教材。全书各章节的编写分工如下：第 1 章由吴长勤编写，第 2 章由胡芬玲和江健生编写，第 6、7 章由蔡庆华编写，第 3、11 章由葛华编写，第 4、5 章由王一宾和陈义仁编写，第 8、13 章由马丽生编写，第 9、10 章由程一飞编写，第 12 章由吴海峰编写。

本书在策划和出版过程中得到了许多高校从事 C 语言教学工作同仁的关心和帮助，特别是陈桂林教授和苏本跃教授，他们对本书的编写提出了很多宝贵的建议；西南大学的邹显春教授对本书进行了审稿并提出了许多修改意见，在此一并表示感谢。本书在编写过程中参考了国内外许多正式或非正式出版的相关著作，同样向这些著作的作者们致以衷心的感谢。

由于编者水平有限，书中难免存在许多不足之处，敬请广大读者批评指正。

编 者

2011 年 11 月

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010) 58581897 58582371 58581879

反盗版举报传真 (010) 82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

目 录

第 1 章 C 语言概述.....	1	2.4 四舍五入运算.....	24
1.1 为什么要学 C 语言.....	1	2.4.1 程序解析.....	25
1.1.1 C 语言与黑客.....	1	2.4.2 数据类型转换.....	25
1.1.2 C 语言的发展.....	2	2.5 综合应用实例.....	28
1.1.3 C 语言的特点.....	3	第 3 章 输入输出函数.....	30
1.2 两个 C 语言程序.....	4	3.1 求整数的各位数字和.....	30
1.2.1 C 语言程序框架.....	4	3.1.1 程序解析.....	30
1.2.2 C 语言的语句.....	6	3.1.2 格式化输出 printf() 函数.....	31
1.2.3 C 语言的编程风格.....	8	3.2 求华氏温度对应的摄氏温度.....	35
1.3 程序与算法.....	8	3.2.1 程序解析.....	35
1.3.1 程序.....	8	3.2.2 格式化输入 scanf() 函数.....	36
1.3.2 算法特点.....	8	3.3 字母大小写转换.....	41
1.3.3 算法表示.....	9	3.3.1 程序解析.....	41
1.3.4 程序设计过程.....	10	3.3.2 字符输入函数 getchar().....	41
第 2 章 基本数据类型.....	12	3.3.3 字符输出函数 putchar().....	42
2.1 求平均分.....	12	3.4 综合应用实例.....	43
2.1.1 程序解析.....	13	第 4 章 选择结构.....	47
2.1.2 整型.....	13	4.1 求整数绝对值.....	47
2.1.3 实型.....	14	4.1.1 程序解析.....	47
2.1.4 字符型.....	15	4.1.2 关系运算符.....	49
2.2 求三角形面积.....	15	4.1.3 if 语句.....	50
2.2.1 程序解析.....	16	4.1.4 条件运算符.....	52
2.2.2 常量.....	16	4.2 判断字符类别.....	53
2.2.3 变量.....	18	4.2.1 程序解析.....	53
2.2.4 函数.....	19	4.2.2 逻辑运算符.....	55
2.2.5 表达式.....	20	4.2.3 多分支 if 语句.....	57
2.3 整数逆序输出.....	21	4.3 转换成绩等级.....	59
2.3.1 程序解析.....	21	4.3.1 程序解析.....	59
2.3.2 算术运算.....	21	4.3.2 switch 语句.....	60
2.3.3 赋值运算.....	23	4.3.3 if 语句的嵌套.....	64
2.3.4 逗号运算.....	24		

II 目录

4.4 综合应用实例	66	6.4.1 程序解析	121
第5章 循环结构	71	6.4.2 变量的作用域	122
5.1 求1到100之和	71	6.4.3 变量的存储属性	123
5.1.1 程序解析	71	*6.4.4 内部函数与外部函数	127
5.1.2 for 循环语句	74	6.5 编译预处理	127
5.2 求满足条件的 N	76	6.5.1 宏定义命令	128
5.2.1 程序解析	76	6.5.2 文件包含命令	130
5.2.2 while 循环语句	77	*6.5.3 条件编译命令	130
5.3 判断整数的位数	81	6.6 综合应用实例	131
5.3.1 程序解析	81	第7章 数组	135
5.3.2 do-while 循环语句	82	7.1 求高于平均分人数	135
5.3.3 循环语句的选择	85	7.1.1 程序解析	135
5.4 素数判断	86	7.1.2 一维数组的定义和引用	136
5.4.1 程序解析	86	7.1.3 一维数组的存储和初始化	137
5.4.2 循环控制语句	89	7.1.4 一维数组的应用	138
5.5 求阶乘的和	92	7.2 求矩阵最大值	139
5.5.1 程序解析	92	7.2.1 程序解析	139
5.5.2 循环嵌套	94	7.2.2 二维数组的定义和引用	140
5.6 综合应用实例	97	7.2.3 二维数组的存储和初始化	141
第6章 函数	106	7.2.4 二维数组的应用	142
6.1 求周长与面积	106	7.3 数据排序	144
6.1.1 程序解析	106	7.3.1 程序解析	144
6.1.2 程序的结构	107	7.3.2 数组元素作为函数参数	149
6.1.3 函数的分类	108	*7.3.3 二维数组作为函数参数	150
6.1.4 函数的定义	108	7.4 综合应用实例	153
*6.1.5 函数设计的基本原则	111	第8章 指针	156
6.2 验证哥德巴赫猜想	111	8.1 保险柜密码	156
6.2.1 程序解析	111	8.1.1 程序解析	156
6.2.2 函数调用及参数传递	112	8.1.2 指针与地址	157
6.2.3 函数的返回值	114	8.1.3 指针变量的定义	159
6.2.4 函数的声明	115		
6.3 求最大公约数	116		
6.3.1 程序解析	116		
6.3.2 函数的嵌套调用	118		
6.3.3 函数的递归调用	119		
6.4 Fibonacci 数列	121		

8.1.4 指针运算.....	159	9.5.1 指针数组.....	201
8.1.5 多级指针的概念.....	163	9.5.2 指针数组用于命令行 参数.....	203
8.2 企业员工考勤.....	164	第 10 章 复杂数据类型.....	205
8.2.1 程序解析.....	164	10.1 二维坐标.....	205
8.2.2 一维数组的地址表示.....	166	10.1.1 程序解析.....	205
8.2.3 指向数据元素的指针.....	167	10.1.2 结构体的概念与定义.....	207
8.2.4 指针作为函数参数.....	168	10.1.3 结构体变量的定义.....	208
8.3 发扑克牌.....	170	10.1.4 结构体变量的初始化.....	208
8.3.1 程序解析.....	170	10.1.5 结构体类型的嵌套 定义.....	209
8.3.2 二维数组的地址表示.....	172	10.1.6 结构体变量的使用.....	209
8.3.3 一级指针访问二维数组.....	174	10.1.7 结构体所占内存的 字节数.....	211
8.3.4 行指针访问二维数组.....	174	10.1.8 结构体数组.....	212
8.3.5 行指针作为函数参数.....	175	10.1.9 结构体指针.....	214
8.4 抽奖游戏.....	177	10.1.10 结构体与函数.....	215
8.4.1 程序解析.....	177	10.2 职工信息.....	217
8.4.2 指向函数的指针.....	178	10.2.1 共用体类型的定义.....	218
8.4.3 返回指针的函数.....	180	10.2.2 共用体变量的说明.....	219
8.5 综合应用实例.....	181	10.2.3 共用体变量的引用.....	219
第 9 章 字符串.....	186	* 10.3 颜色枚举.....	220
9.1 加密问题.....	186	10.3.1 枚举类型的定义.....	221
9.1.1 程序解析.....	186	10.3.2 枚举变量的说明及 引用.....	221
9.1.2 字符串的表示.....	187	10.3.3 用户定义类型.....	222
9.1.3 字符串的输入输出.....	188	* 10.4 链表.....	223
9.1.4 字符串和字符指针.....	189	10.4.1 程序解析.....	224
9.2 奥运会入场次序.....	191	10.4.2 内存分配.....	225
9.2.1 程序解析.....	191	10.4.3 单向链表.....	226
9.2.2 字符串数组.....	192	10.5 综合应用实例.....	231
9.2.3 字符串处理函数.....	192	第 11 章 位运算.....	237
9.3 统计数字字符个数.....	195	11.1 几个基本概念.....	237
9.3.1 程序解析.....	195	11.2 位运算符.....	239
9.3.2 字符串作为函数参数.....	196	* 11.3 位域.....	241
9.3.3 返回字符串的函数.....	198		
9.4 综合应用实例.....	198		
* 9.5 本章扩充内容.....	201		

IV 目录

11.4 综合应用实例.....	244	第 13 章 综合程序设计	266
第 12 章 文件操作	247	13.1 通讯录管理	266
12.1 写数据文件.....	247	13.1.1 项目要求及分析	266
12.1.1 程序解析.....	247	13.1.2 总体设计	266
12.1.2 文件的概念.....	248	13.1.3 代码实现	267
12.1.3 文件的分类.....	248	13.2 计算 100 的阶乘实例	271
12.1.4 文件的指针.....	249	13.2.1 项目要求及分析	271
12.2 读学生成绩.....	249	13.2.2 总体设计	271
12.2.1 程序解析.....	250	13.2.3 代码实现	272
12.2.2 文件的打开.....	251	附录 A C 语言运算符的优先级与 结合性	274
12.2.3 文件的关闭.....	252	附录 B 常用字符与 ASCII 码值 对照表	275
12.3 文件复制.....	252	附录 C C 语言常用库函数表	276
12.3.1 程序解析.....	252	参考文献	281
12.3.2 字符读写函数.....	253		
12.3.3 字符串读写函数.....	255		
12.3.4 格式化读写函数.....	256		
12.3.5 数据块读写函数.....	258		
12.3.6 文件的其他函数.....	260		
12.4 综合应用实例.....	261		

第 1 章 C 语言概述

【本章要点】

- 什么是程序？程序设计语言一般包含哪些功能？
- 为什么要学 C 语言？
- C 语言有哪些特点？
- C 语言程序的基本框架是怎样的？
- 什么是算法？算法如何描述？
- 程序设计过程是怎样的？

对于将 C 语言作为学习的第一门程序设计语言的读者来说，最关心的问题是如何尽快学会用 C 语言进行程序设计。要做到这一点，首先要对程序设计语言有所了解，更重要的是通过不断地编程实践，逐步领会和掌握程序设计的基本思想和方法。

熟练的编程技巧是在知识与经验不断积累的基础上发展起来的。初学者一开始由于缺乏足够的程序语言知识和编程经验，对于较简单的问题也会感到无所适从，不知如何编写程序。本书建议读者从一开始就学习编写 C 语言程序，通过模仿教材中的程序，试着改写并体会它们，循序渐进，进而独立地解决比较复杂的问题。

为了使读者能从简单的模仿中体会程序设计的基本思想和方法，而不是拘泥于具体的语法细节，本章作为全书的引导，将简要介绍 C 语言的发展历史、基本特点、简单的 C 语言程序结构和算法，以及程序设计求解问题的一般步骤。

1.1 为什么要学 C 语言

“为什么要学 C 语言？”是一个总是被许多初学程序设计的人问到的问题。不同的人可能会给出不同的答案。但是，很多初学者，尤其是有独立思考精神的初学者，往往会对得到的答案表示怀疑。这种怀疑直接导致的后果是学习目标不明确，进而学习兴趣不大，最后就不爱学习 C 语言了。究其根源在于学习 C 语言的真正理由并没有被揭示出来。

1.1.1 C 语言与黑客

黑客一般是指世界上顶尖的计算机高手，其中比较著名的有李纳斯、丹尼斯·利奇、沃兹尼克、肯·汤普生等。黑客一词最早源自英文单词 **hacker**，原指热衷于计算机技术、水平高超的计算机专家，尤其是程序设计人员。早期在美国的计算机界，黑客是带有褒义的。但在媒体报道中，黑客往往用于指代那些“软件骇客”（software cracker），因而具有了贬义性。

Dennis M. Ritchie（丹尼斯·利奇，又译为丹尼斯·里奇），C 语言之父，UNIX 之父，如图 1-1 所示。他生于 1941 年 9 月 9 日，获得哈佛大学数学博士学位，曾担任朗讯科技公司贝尔实验室（原 AT&T 实验室）下属的计算机科学研究系统软件研究部的主任一职。

Ken Thompson (肯·汤普生), C语言前身B语言的作者, UNIX的发明人之一, Belle (一个著名的国际象棋程序)的作者之一, 操作系统Plan 9的主要作者, 如图1-2所示。

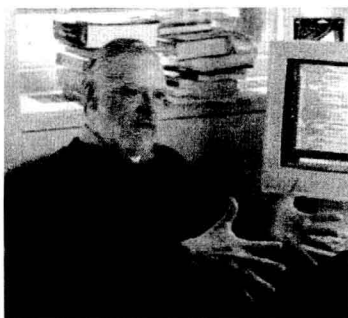


图 1-1 Dennis M. Ritchie



图 1-2 Ken Thompson

作为 Dennis M. Ritchie 亲密的合作者, Ken Thompson 在 UNIX 的开发中起了主导作用, 而 Dennis M. Ritchie 则在 C 语言的设计中发挥更大的作用。他们两人共同获得了 1983 年度的图灵奖, 这也是图灵奖唯一一次被颁发给计算机软件工程师。

Ken Thompson 和 Dennis M. Ritchie 被奉为黑客圈中的英雄, 大家都尊称他们为 Ken 和 DMR。几十年过去了, 黑客圈中再没有第二个人自称 Ken, 虽然这是一个在英语中司空见惯的名字。

C 语言、UNIX 操作系统和脚本语言至今仍是年青黑客被圈子接受前必须苦练的三大技艺。它们由黑客设计, 被黑客推崇。当然, 对于程序设计的初学者来说, 学好 C 语言, 对理解计算机软件的工作过程, 以及今后从事软件开发、学习其他软件都有非常大的帮助。

1.1.2 C语言的发展

语言是人类交流思想和情感的工具, 当今世界很多民族都有自己的语言。计算机程序设计语言是人类根据计算机硬件的特征所设计的、计算机可识别的语言, 用于描述人们希望解决问题的方法和步骤, 供计算机阅读和执行。

C 语言是国际上广泛流行的、很有发展前途的计算机高级程序设计语言。它适合于作为系统描述语言, 既可用于编写系统软件, 也可用于编写应用软件。

以前的操作系统 (包括 UNIX 操作系统在内) 等系统软件主要是用汇编语言编写的。由于汇编语言依赖于计算机硬件, 程序的可读性和可移植性都比较差。为了在这些方面有所改善, 最好使用高级语言, 但一般高级语言难以实现汇编语言的某些功能 (汇编语言可以直接对硬件进行操作, 例如, 对内存地址的操作、位操作等)。人们希望能够找到一种既具有一般高级语言特性, 又具有低级语言特性的语言, 集两者的优点于一身。于是, C 语言就在这种情况下应运而生。

C 语言的前身是 ALGOL 语言 (ALGOL 60 是一种面向问题的高级语言)。1963 年, 英国剑桥大学推出了 CPL (Combined Programming Language) 语言。CPL 语言在 ALGOL 语言的基础上增加了硬件处理能力, 但其语言规模比较大, 难以实现。

1967 年,剑桥大学的马丁·理查德对 CPL 语言进行了简化,提出 BCPL (Basic Combined Programming Language) 语言;1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,进一步对其简化,提出了 B 语言(取 BCPL 的第一个字母),并用 B 语言写出了 UNIX 操作系统,在 PDP-7 计算机上实现。1972 年,美国贝尔实验室的 Brian W. Kernighan 和 Dennis M. Ritchie 对 B 语言进行了完善和扩充,提出了 C 语言(取 BCPL 的第二个字母)。最初的 C 语言是为了描述和实现 UNIX 操作系统而提出的一种工作语言。1973 年, Ken Thompson 和 Dennis M. Ritchie 两人合作,将 UNIX 操作系统 90% 以上的代码用 C 语言改写 (UNIX 第 5 版),实现了 C 语言的现实应用。

后来, C 语言多次改进,但主要还是在美国贝尔实验室内部使用。1977 年,出现了不依赖于具体机器的 C 语言编译版本,使 C 语言程序移植到其他机器的过程得到简化,推动了 UNIX 操作系统在各种机器上实现。随着 UNIX 操作系统的广泛使用, C 语言也迅速得到推广。

1978 年, Ken Thompson 和 Dennis M. Ritchie 合著了《The C Programming Language》,被称为标准 C。1983 年,美国国家标准化协会根据不同 C 语言版本对 C 语言的发展和扩充,制定了新的 C 语言标准——ANSI C, 1987 年,又公布了新标准——87 ANSI C,即现在流行的 C 语言。

1980 年,美国贝尔实验室的 Bjarne Stroustrup 博士在 C 语言的基础上,开发出一种过程性与对象性相结合的程序设计语言,即 C++ 语言,这种语言增加了面向对象的特征,并弥补了 C 语言在开发大型应用程序方面的不足。

1.1.3 C 语言的特点

1. C 语言的优点

C 语言之所以能存在和发展,并具有生命力,是因为其具有不同于其他语言的特点。

(1) C 语言简洁、紧凑,使用方便、灵活

C 语言一共只有 32 个关键词, 9 种控制语句,程序书写形式自由,主要用小写字母表示,压缩了一切不必要的成分,相对于其他高级语言源程序更加短小,因此输入程序时的工作量更少。

(2) 运算符丰富,表达式能力强

C 语言共有 34 种运算符,运算范围广泛,除一般高级语言所使用的算术、关系和逻辑运算符外,还可以实现以二进制位为单位的运算,并且具有如++、--等单目运算符和+=、-=、*=、/=等复合运算符。

(3) 数据结构丰富,便于数据的描述与存储

C 语言具有丰富的数据结构,其数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型和共用体类型等,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。

(4) C 语言是结构化、模块化的编程语言

程序的逻辑结构可以使用顺序、选择和循环 3 种基本结构组成。C 语言程序采用函数结构,十分便于把整体程序分割成若干相对独立的功能模块,并且为程序模块间的相互调用以及数据传递提供了便利。

(5) 语法限制不太严格,程序设计自由度大

例如,对数组下标越界不作检查,由程序编写者自己保证程序的正确。对变量的类型使用

比较灵活,例如,整型数据与字符型数据以及逻辑型数据可以通用。一般的高级语言语法检查比较严格,能检查出几乎所有的语法错误。而C语言允许程序编写者有较大的自由度,因此放宽了语法检查。程序编写者应当仔细检查程序,保证其正确,而不要过分依赖C编译程序去查错。“限制”与“灵活”是一对矛盾。限制严格,就失去灵活性;而强调灵活,就必然放松限制。一个不熟练的程序编写者,编一个正确的C语言程序可能会比编一个其他高级语言程序难一些。也就是说,对要用C语言编程的人,要求对程序设计更熟练一些。

(6) C语言是具有低级语言功能的高级语言

C语言允许直接访问物理地址,能进行位(bit)操作,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,可以用来编写系统软件。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语言称为“高级语言中的低级语言”,也有人称它为“中级语言”,即兼有高级语言和低级语言的特点。

(7) 生成目标代码质量高,程序执行效率高

用C语言编写的程序,经编译后生成的可执行代码比用汇编语言直接编写的代码执行效率仅低15%~20%,这是其他高级语言无法比拟的。

(8) 可移植性好

与汇编语言相比,C程序不做修改就可以运行于各种型号的计算机和各种操作系统。

2. C语言的不足

当然,C语言也有一些不足,主要表现在以下几个方面。

(1) C语言的数据类型检查不严格,表达式容易出现二义性,不能自动检查数组下标越界,初学者较难掌握运算符的优先级及结合性等,这使得程序中的一些错误不能在编译时被发现。

(2) C语言缺乏支持代码重用的语言结构。

(3) C语言不适合开发大型应用程序,因为当程序规模达到一定程度时,程序员很难控制程序的复杂性。

C++语言正是为解决上述问题而设计的。C++继承了C语言的精髓,增加了面向对象机制,弥补了C语言的不足,并保持与C语言的兼容,所以,许多C程序代码不用修改就可以直接在C++编译环境中使用。本书所有案例均在C++环境中调试通过。

1.2 两个C语言程序

为了让读者对C语言有一个直观的认识,首先介绍两个实例,通过这两个实例,读者可轻松步入C语言程序的世界。在例1-1和例1-2中,程序实现的功能非常简单,但通过它们可以了解C语言程序的基本结构和框架,并了解程序是如何运行的。

1.2.1 C语言程序框架

为了使读者尽快地了解C语言程序的构成,增加对C语言程序的感性认识,首先分析两个实例。

【例1-1】 根据给定的程序代码初步认识C语言程序的构成。该程序的功能是在屏幕上显

示 “How are you!”。

【程序代码】

```
#include <stdio.h>                /* 编译预处理命令 */
void main()                       /* 主函数 */
{
    printf("How are you!\n");      /* 输出函数 */
}
```

【分析】

(1) 一个 C 语言程序由带有“#”的编译预处理语句开始。`#include <stdio.h>`也可写成`#include "stdio.h"`，表示在编译系统预定义的目录中查找需要的库函数，本例中的库函数是 `printf()` 函数。

(2) `main()` 函数是每个程序必须具有的主函数，一个程序有且仅有一个 `main()`，`void` 表示函数没有返回值。函数若有返回值，可在函数名前指定返回值类型。

(3) “{” 表示函数的开始，“}” 表示函数的结束。

(4) `printf("How are you!\n");` 是函数体，是一条输出函数调用语句，功能是在显示器屏幕上显示信息 “How are you!”，其中 “\n” 表示换行。

(5) 由于本程序功能简单，不需要定义其他函数，只有一个 `main()` 函数。本教材前 5 章的实例中大多只有一个 `main()` 函数。

上述代码在 Visual C++ 6.0 环境下，通过编辑、编译、连接和运行，可得到如图 1-3 所示的运行结果。

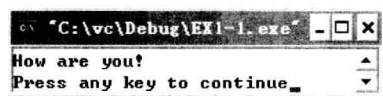


图 1-3 例 1-1 的运行结果

图 1-3 中第 2 行的 “Press any key to continue” 并非程序所指定的输出，而是 Visual C++ 输出运行结果后由系统自动加上的一行信息，通知用户 “按任意键继续”。当用户按下任意键后，输出窗口消失，回到 Visual C++ 的主窗口。

【例 1-2】 下列程序代码的功能是输入圆的半径，计算并输出圆的面积。

【程序代码】

```
#include <stdio.h>                // 编译预处理命令
#define PI 3.14                  // 定义宏常量 PI
float area(int r)                /* 定义圆面积 area() 函数 */
{
    return PI*r*r;              /* area() 函数的返回值 */
}
void main()                     // 定义 main() 函数
{
    int r;                      /* 说明圆半径 r 为整型变量 */
    float s;                    /* 说明面积 s 为单精度浮点型变量 */
    printf("请输入圆的半径: ");
```



```

scanf("%d",&r);           /*输入半径 r*/
s=area(r);                /*调用 area()函数计算 s 的值*/
printf("r=%d, s=%f\n", r, s); /*输出圆的半径和面积*/
}

```

【分析】

- (1) 上述程序由3部分组成，第一部分是以#开头的两条编译预处理命令。
- (2) 第3到6行是求圆面积子函数 area()的定义。
- (3) 程序的其余部分是 main()函数。其功能是输入圆半径，调用 area()函数求圆面积并输出。
- (4) scanf()函数的功能是通过键盘输入数据，输入整型数据时使用的格式符是%d。
- (5) 在 printf()函数中，输出单精度浮点型数据时使用的格式符是%f，默认为6位小数；输出整型数据时使用的格式符是%d。

运行程序，通过键盘输入半径10，得到如图1-4所示的结果。

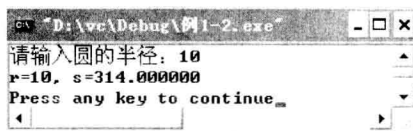


图1-4 例1-2的运行结果

通过以上两个C语言程序的例子，可以看出：

- (1) C语言程序由函数组成。每个C语言程序有且仅有一个主函数，该主函数的函数名规定为 main，也可以包含若干个子函数。
- (2) C语言程序的书写格式自由，一行内可以写多条语句，一条语句也可以写在多行上。每条语句必须以“;”作为结束标志。
- (3) C语言程序的执行总是从主函数“{”开始，并在主函数“}”中结束。
- (4) C语言本身没有输入输出语句。C语言对输入输出实行“函数化”。输入和输出操作是由 scanf()和 printf()等库函数来完成的。
- (5) 可以用/*.....*/或 // 对C语言程序中的任何部分作注释。

【练习1-1】编写程序，在屏幕上显示两个短句“Programming is fun.”和“And Programming in C is even more fun!”，每行显示一句。

【练习1-2】编写程序，在屏幕上显示如下图案。

```

* * * *
* * *
* *
*

```

1.2.2 C语言的语句

在进行程序设计时有两部分工作，一部分是数据的设计，另一部分是操作的设计。数据设计的结果是一系列的数据描述语句，主要用来定义数据的类型，完成数据的初始化等；操作设计的结果是一系列的操作控制语句，其作用是向计算机系统发出操作指令，以完成对数据的加

工和流程控制。

在 C 语言中，无论是运算操作还是流程控制，都是由相应的语句完成的。C 语言的语句用来向计算机系统发出操作指令。一个语句经过编译后产生若干条机器指令。C 语言的语句可分为以下 5 种类型：表达式语句、函数调用语句、控制语句、复合语句和空语句。

1. 表达式语句

由表达式组成的语句称为表达式语句，其作用是计算表达式的值或改变变量的值，其一般形式是：

表达式：

即在表达式的末尾加上分号，就变成了表达式语句。最典型的例子是由赋值表达式构成赋值语句，如 `x=5;`。

温馨提示：分号是 C 语言中语句的标志，一个语句必须有分号，没有分号，则一定不是 C 语言语句。

2. 函数调用语句

函数调用语句是由一个函数调用加上分号构成的，其作用是完成特定的功能。它的一般形式是：

函数名（参数列表）；

例如

```
printf("This is a C statement"); /*调用库函数，输出字符串*/
```

3. 控制语句

控制语句用于完成一定的控制功能，以实现程序的各种结构方式。C 语言有 9 种控制语句，可分为 3 类。

(1) 条件判断语句：if 语句、switch 语句。

(2) 转向语句：break 语句、continue 语句、goto 语句、return 语句。

(3) 循环语句：for 语句、while 语句、do-while 语句。

4. 复合语句

复合语句是用大括号将若干语句组合在一起形成的语句，又称为分程序，在语法上相当于一条语句。例如，下面是一个复合语句：

```
{
    i++;
    printf("%d\n",i);
}
```

温馨提示：复合语句中最后一条语句的分号不能省略；大括号后面没有分号。

5. 空语句

只有一个分号的语句称为空语句。空语句是什么也不执行的语句，常用于循环语句中，表示循环体什么都不做。例如：

```
while (getchar()!='\n')
    ; /*空语句*/
```

该循环语句的功能是从键盘输入字符，一直到输入回车键才退出循环。