

高等院校信息技术规划教材

C语言程序设计

马秀丽 刘志妩 李筠 编著

INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY

清华大学出版社



TP312/2779

2008

高等院校信息技术规划教材

C语言程序设计

马秀丽 刘志妩 李筠 编著

清华大学出版社
北京

内 容 简 介

本书较全面和详细地介绍了 C 语言的所有语法规则,并以语法规则为基本知识点,通过大量的举例应用和程序分析,重点讲解语法规则的运用和编程的解题思路,目的是培养读者 C 语言的程序分析和编程设计能力。本书开篇以简单 C 程序入手,详细介绍了 C 程序的基本结构,以及在常用的 Turbo C 环境下和在 Visual C++ 6.0 环境下的控制台应用程序的开发过程,使读者很快就能上机编程。

本书内容全面,概念清楚,结构合理,实例丰富,逻辑性强,文字通俗易懂。教师可以从清华大学出版社网站 www.tup.com.cn 上下载本书的电子课件、所有例题代码、习题解答及编程题程序代码及其注释。本书可作为高等院校计算机及其相关专业的本科生教材,也可作为爱好 C 语言的初学者和程序设计人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

C 语言程序设计/马秀丽,刘志妩,李筠编著. —北京:清华大学出版社,2008.3
(高等院校信息技术规划教材)

ISBN 978-7-302-16938-3

I. C… II. ①马… ②刘… ③李 III. C 语言—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 012404 号

责任编辑:袁勤勇 赵晓宁

责任校对:梁 毅

出版发行:清华大学出版社

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

c-service@tup.tsinghua.edu.cn

社总机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 装 者:北京季蜂印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:21.5

版 次:2008 年 3 月第 1 版

印 数:1~4000

定 价:29.00 元

字 数:497 千字

印 次:2008 年 3 月第 1 次印刷

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:028155-01

前言

foreword

C语言是一种面向过程的程序设计语言,对于软件工程师来说,它是今后从事面向对象的C++或VC++程序设计的基础;对于硬件工程师来说,它是从事以单片机或嵌入式内核处理器等为核心的硬件产品开发过程中,程序设计的必备工具。所以,将要从事计算机及其外围设备的软硬件开发的技术人员一定要学好C语言。

很多C语言的初学者认为,C语言是一门难学的功课,其实不然。一门功能难学主要有两个方面的原因:一是这门功能技术高深,难于理解,因此难于掌握和运用;二是不了解这门功能的特点,没有掌握它的学习方法。很显然,认为C语言难学主要是因为没有掌握到它的学习方法。

通过总结多年的工程实践和教学工作经验,借此机会与读者谈谈学习体会。C语言是一种人与机器对话的语言工具,既然它是一门语言工具,就应遵循学习一门语言的学习方法。所以,学好C语言一方面是要记忆,即要掌握C语言的语法规则(类似一门语言中的句型部分)、关键字和运算符的使用规划(类似一门语言中的单词部分)。语法规则主要包括3个方面:数据类型(包括整型、实型、字符型、数组、指针、结构体、共用体、枚举类型)的定义规则,语句(主要包括9种控制语句)的语法规则,函数的定义和调用规则;关键字有32个;运算符有34个。学好C语言另一方面是要实践,即编程和上机实践。编程包括设计算法和编写程序代码两方面:设计算法要按照面向过程的程序设计方法进行(即自顶向下,逐步细化的设计方法,以及任何复杂的问题都可以描述成由三种基本结构所组成的设计思想);编写程序代码是将设计好的算法翻译成C语言代码的过程,也是将前面记忆的语法知识依照算法的逻辑结构加以运用的过程。上机实践是最终实现人机对话的过程,也是检验程序设计方法和程序代码(即所记忆知识的运用)正确与否的过程。最后要牢记一点的是记忆和实践不是孤立的两个方面,而是在学习过程中不断交替和循环进行的两个方面。对于初学者,一定要边学习、边记忆、边实践。相信读者对学好C语言一定有信心了。

本书作为 C 语言的学习教材和参考书,内容系统全面,深入浅出,配套资源丰富。在本书的编写过程中,笔者力求通过对 C 语法规则的程序举例与分析,帮助读者掌握语法规则的运用,理解程序设计思路,掌握程序设计方法,从而培养程序设计能力。

本书的特色体现在以下 4 个方面。

(1) 强化基础知识及运用。

本书较全面和详细地介绍了 C 语言的所有语法规则,并以语法规则为基本知识点,通过举例强化语法规则的运用。

(2) 强调重要概念及编程方法。

强调算法的重要性,一个好的程序设计离不开好的算法,算法是程序设计的方法,是核心和灵魂,语法是程序设计的工具,读者既要了解语法工具和学会使用工具,更要掌握程序设计的方法。

基本数据类型、控制语句、数组和函数是面向过程程序设计的重要基础,也是掌握 C 程序结构和程序设计方法的关键内容,这方面内容的讲解较为详细,举例由浅入深,循序渐进。

指针是 C 语言的重要概念,是掌握 C 语言精华的关键,正确而灵活地运用指针,在程序设计中可以起到事半功倍的效果,本书对这方面的内容重点讲解,并通过程序分析和编程举例加深理解和运用。

结构体和链表是数据结构的基础,也是灵活运用 C 语言解决实际问题的重要方法和工具。本书通过图解形式,详细讲解了链表的操作方法,结构清晰,易于理解。

(3) 强调动手实践。

本书详细介绍了常用的 Turbo C 环境下和 Visual C++ 6.0 环境下的控制台应用程序的开发过程;书中所有相关例程均在 Visual C++ 6.0 环境下调试通过,并给出了运行结果,不是纸上谈兵,而是实战演习。

(4) 配套资源丰富。

本书配有多媒体光盘,其中包含电子课件、所有相关例程源代码、习题解答及编程题的程序源代码,并且在程序的关键部分加以注释,既适合作为教材供教师和学生使用,也适合自学成才。

本书第 1、4、6、10、11 章由马秀丽编写,第 2、5、7、8 章由刘志妩编写,第 9、12 章由李筠编写,第 3 章由刘志妩和李筠共同编写。王红霞、周焕春、王威、龙治海参加了部分编写工作。另外,特别感谢张秉权教授审阅了全稿,并提出了许多宝贵建议。同时感谢本书所列参考文献的作者。感谢为本书出版付出辛勤劳动的清华大学出版社的工作人员。

感谢读者选择本书,欢迎对本书内容提出建议,对此我们将非常感激。作者的联系地址如下。

电子邮件地址:xiulima@sina.com

通信地址:沈阳理工大学信息科学与工程学院 马秀丽

邮政编码:110168

作者

2007 年 11 月于沈阳

目录

Contents

第 1 章 C 语言概述	1
1.1 C 语言简介	1
1.2 C 程序的设计过程	2
1.2.1 程序设计	2
1.2.2 程序编写	2
1.2.3 上机实现	3
1.3 简单 C 程序介绍	4
1.4 C 程序的上机步骤	7
1.4.1 在 Turbo C 环境下建立和运行 C 程序步骤	8
1.4.2 在 Visual C++ 环境下建立和运行 C 程序的步骤	11
习题 1	16
第 2 章 算法	17
2.1 算法的概念	17
2.2 计算机算法的表示方法	19
2.2.1 自然语言表示算法	19
2.2.2 传统流程图表示算法	21
2.2.3 用 N-S 结构化框图表示算法	22
2.3 算法的特点及算法设计的要求	25
2.4 计算机程序设计的基本方法	26
习题 2	28
第 3 章 基本数据类型及数据的输入输出	29
3.1 C 语言的基本数据类型简介	29
3.2 标识符、常量和变量	30
3.2.1 标识符	30



3.2.2	常量	31
3.2.3	变量	33
3.3	基本数据类型	33
3.3.1	整型	33
3.3.2	实型	35
3.3.3	字符型	37
3.4	数据的输入与输出	38
3.4.1	字符输出函数	39
3.4.2	字符输入函数	40
3.4.3	格式输出函数	42
3.4.4	格式输入函数	47
习题 3		51
第 4 章	运算符与表达式	54
4.1	C 的运算符与表达式	54
4.1.1	运算符	54
4.1.2	表达式	54
4.2	算术运算符与算术表达式	55
4.3	强制类型转换运算符	55
4.4	增量运算符与增量表达式	56
4.5	赋值运算符与赋值表达式	58
4.6	逗号运算符与逗号表达式	61
4.7	关系运算符与关系表达式	62
4.8	逻辑运算符与逻辑表达式	63
4.9	条件运算符与条件表达式	65
4.10	位运算符和位运算	66
4.11	其他运算符	68
习题 4		69
第 5 章	C 语句及其程序设计	72
5.1	C 语句概述	72
5.2	顺序结构程序设计	74
5.3	选择结构程序设计	75
5.3.1	if 语句及程序设计	76
5.3.2	条件运算符及程序设计	86
5.3.3	switch 语句及程序设计	87
5.3.4	选择结构程序设计综合举例	91

5.4 循环结构程序设计	97
5.4.1 while 语句及程序设计	97
5.4.2 do...while 语句及程序设计	99
5.4.3 for 语句及程序设计	100
5.4.4 循环的嵌套	104
5.4.5 转移控制语句的应用	107
5.4.6 循环结构程序设计综合举例	109
习题 5	114
第 6 章 数组	119
6.1 一维数组	119
6.1.1 一维数组的定义	119
6.1.2 一维数组元素的引用	121
6.1.3 一维数组的初始化	121
6.1.4 一维数组编程举例	122
6.2 二维数组	135
6.2.1 二维数组的定义	135
6.2.2 二维数组的引用	136
6.2.3 二维数组的初始化	136
6.2.4 二维数组编程举例	142
6.3 字符数组	142
6.3.1 字符数组的定义	142
6.3.2 字符数组的初始化	143
6.3.3 字符数组的引用	145
6.3.4 字符数组的输入输出	145
6.3.5 字符串处理函数	147
6.3.6 字符数组编程举例	151
习题 6	157
第 7 章 函数	161
7.1 函数的概念	161
7.1.1 标准库函数	161
7.1.2 用户自定义函数	162
7.2 函数的定义	164
7.3 函数的调用	166
7.3.1 调用函数	166
7.3.2 声明函数	168



7.4	函数的参数传递	169
7.4.1	传值方式	170
7.4.2	传地址方式	171
7.5	函数的嵌套调用与递归调用	174
7.5.1	函数的嵌套调用	174
7.5.2	函数的递归调用	176
7.6	变量的作用域	180
7.6.1	局部变量	180
7.6.2	全局变量	180
7.7	变量的存储属性	183
7.7.1	变量的存储类型	183
7.7.2	动态变量	184
7.7.3	局部静态变量	184
7.7.4	外部变量	186
7.7.5	静态外部变量	187
7.8	函数应用程序设计综合举例	187
习题 7		192
第 8 章	编译预处理命令	197
8.1	宏定义	197
8.1.1	不带参数的宏定义	197
8.1.2	带参数的宏定义	200
8.1.3	宏与函数的区别	203
8.1.4	宏定义的解除	204
8.2	文件包含	205
8.3	条件编译	207
习题 8		210
第 9 章	指针	214
9.1	指针与指针变量	214
9.1.1	指针的概念	214
9.1.2	指针变量的定义	215
9.1.3	指针变量的两种运算符	216
9.1.4	指针变量的初始化	217
9.1.5	引用指针变量	218
9.2	指针与函数	219

9.2.1 指针作函数的参数	219
9.2.2 函数返回指针	221
9.2.3 指向函数的指针	222
9.3 指针与数组	223
9.3.1 通过指针引用一维数组元素	223
9.3.2 指针基本运算	227
9.3.3 通过指针引用二维数组元素	229
9.4 字符串与指针	232
9.4.1 字符数组与字符指针	232
9.4.2 常见的字符串操作	233
9.5 指针数组、数组指针及应用	237
9.5.1 指针数组与数组指针	237
9.5.2 main 函数的参数	241
9.6 指向指针的指针	242
习题 9	248
第 10 章 结构体类型与链表操作	251
10.1 结构体的概念和定义	251
10.2 结构体变量的定义和初始化	254
10.2.1 结构体变量的定义	254
10.2.2 结构体变量的初始化	255
10.3 结构体变量的引用	256
10.4 结构体数组	257
10.4.1 结构体数组的定义	257
10.4.2 结构体数组的初始化	258
10.4.3 结构体数组的应用	259
10.5 结构体指针	262
10.5.1 指向结构体变量的指针	262
10.5.2 指向结构体数组的指针	263
10.5.3 结构体变量和结构体指针作函数参数	265
10.6 链表的操作	267
10.6.1 链表的概述	267
10.6.2 链表操作所需要的函数	268
10.6.3 链表的操作	269
习题 10	278

第 11 章 共用体与枚举类型	281
11.1 共用体	281
11.1.1 共用体的概念及定义	281
11.1.2 共用体变量的引用	282
11.1.3 共用体类型编程举例	285
11.2 枚举类型	287
11.2.1 枚举类型的概念及其变量的定义	287
11.2.2 枚举类型数据的使用	287
11.3 用 typedef 定义类型	290
11.3.1 用 typedef 定义类型	290
11.3.2 有关 typedef 的使用	292
习题 11	293
第 12 章 文件	295
12.1 文件的概念	295
12.1.1 什么是文件	295
12.1.2 C 文件的分类	295
12.1.3 文件的使用	297
12.2 文件的处理	297
12.2.1 文件类型的定义	297
12.2.2 打开文件	298
12.2.3 关闭文件	299
12.3 文件的读写	300
12.3.1 fputc 函数和 fgetc 函数	300
12.3.2 fgets 函数和 fputs 函数	303
12.3.3 fprintf 函数和 fscanf 函数	305
12.3.4 fread() 函数和 fwrite() 函数	308
12.4 文件的定位操作	310
12.4.1 fseek() 函数	310
12.4.2 rewind() 函数	311
12.4.3 ftell() 函数	312
12.5 文件操作的状态和出错检测	313
12.5.1 feof() 函数	313
12.5.2 ferror() 函数	313
12.5.3 clearerr() 函数	314

习题 12	314
附录 A C 语言中的关键字	316
附录 B C 语言的运算符及其优先级和结合性	317
附录 C 常用字符与 ASCII 代码对照表	319
附录 D C 库函数	321
参考文献	327

第1章

chapter 1

C 语言概述

本章要点:

- C 语言的特点。
- 简单 C 程序的结构。
- C 程序的设计过程和上机步骤。

C 语言是国际上广泛流行的一种程序设计语言,它的应用十分广泛,适合作为系统描述语言,既可用于编写系统软件,也可用于编写应用软件。

1.1 C 语言简介

C 语言是于 1972 年由贝尔实验室的 Dennis Ritchie 在 B 语言的基础上开发出来的。最初的 C 语言是作为 UNIX 操作系统的开发语言而被人们所认识。此后,贝尔实验室对 C 语言进行了多次改进和版本的公布,C 语言的优点才引起人们的普遍注意。随着 UNIX 操作系统在各种机器上的广泛使用,使 C 语言得到了迅速推广。1978 年由 Brian W. Kernighan 和 Dennis M. Ritchie 合著了《The C Programming Language》一书,该书对 C 语言作了详细的描述,这本书对 C 语言的发展影响深远,并成为了后来 C 语言版本的基础,称之为标准 C。随后 C 语言在各种计算机上快速得以推广,并导致了許多 C 语言版本的出现。

1983 年,美国国家标准化学会(ANSI)对各种 C 语言版本进行统一和改进,制定了新的 C 语言标准,称为 ANSI C,ANSI C 比标准 C 有了很大发展,它进一步明确地定义了与机器无关的 C 语言。1987 年 ANSI 又公布了新的标准,称之为 87 ANSI C。1990 年国际标准化组织 ISO 接受了 87 ANSI C,称之为 ISO C 标准。目前流行的 C 编译系统都是以它为基础的。

目前常用的 C 语言编译系统有 Microsoft C、Turbo C、Quick C 和 Borland C 等,这些编译系统之间略有差异,因此,用户在使用时要注意了解自己的计算机配置了哪种 C 编译系统。

C 语言是一种结构化的程序设计语言。1983 年贝尔实验室在 C 的基础上,又推出了 C++,在经历了三次改进和修订后,1994 年,由美国国家标准化学会制定了 ANSI C++

标准的草案。此后,C++ 又经过不断完善,目前 C++ 仍在不断发展中。

C 语言是建立 C++ 的基础上的,C++ 包含了整个 C,并在 C 的基础上添加了对面向对象编程的支持,使 C++ 成为一种面向对象的程序设计语言。

C 语言之所以应用广泛,主要是其具有如下特点。

(1) C 语言只有 32 个关键字(见附录 I),9 种控制语句,语言简洁、紧凑,使用方便、灵活,程序书写形式自由。

(2) C 语言运算符丰富。C 共有 34 种运算符(见附录 II)。

(3) 数据结构丰富,能够实现各种复杂的数据结构的运算。

(4) 具有结构化的控制语句,用函数作为程序的模块单位,因此,它是理想的结构化程序设计语言。

(5) C 语言能直接访问内存地址,能直接对硬件进行操作,能实现汇编语言的大部分功能。因此,可用于编写系统软件。

(6) 生成的代码质量高,程序运行效率高。

(7) C 语言程序的可移植性好。

总之,使用 C 语言编写程序功能强、限制少、灵活性大,适合编写任何类型的程序。但 C 语言对程序员的要求也较高,需要使用者认真学习和多加练习。

1.2 C 程序的设计过程

C 程序的设计过程通常要经过程序设计、程序编写和上机实现这三大过程。

1.2.1 程序设计

C 语言是一种结构化的程序设计语言,它特别适合于面向过程的程序设计。面向过程的程序设计思路是自顶向下,逐步细化。其方法是将一个复杂问题的解题过程分阶段进行,再将每个阶段的处理过程分解成若干个相对容易处理的子过程,每个子过程的处理方法还可以逐步细化,直到将整个问题的各个环节的处理方法都具体规划设计好为止。

规划好程序的设计方法之后,需要将它描述出来,在描述一个面向过程的程序设计方法(程序的设计方法又称为算法)时,无论一个实际问题的解题过程多么复杂,都可以由顺序结构、选择结构和循环结构这三种基本结构,按自上而下的顺序用算法描述语言(如流程图、N-S 流程图等)描述出来。

当描述好一个 C 程序的算法之后,接下来的工作就是用 C 语言表示这个算法。

1.2.2 程序编写

用 C 语言代码表示算法的过程就是 C 程序的编写过程。在 C 程序的编写过程中必须严格按照 C 语言的语法规则进行,编写好的 C 程序称之为源程序,将源程序输入到计算机中,将生成源代码程序文件。

在编写C程序时,如果程序的算法简单,可以编写成一个由主函数构成的简单C程序;如果程序的算法复杂,通常将一个复杂的算法按功能划分为若干个模块,每个模块的功能可以用C语言编写成一个子函数来表示,而整个算法被编写成一个调用若干个子函数的主函数。因此,一个复杂的C程序就是由一个主函数和若干个子函数构成的。可见,函数是构成C程序的基本单位。

C程序中的函数通常由两部分组成:函数的首部 and 函数体。函数的首部是反映函数的名称、类型、属性和参数。函数体是反映函数的作用和功能。

函数体一般包括两个部分:数据的声明部分和功能的执行部分。

数据的声明部分是定义所使用的变量的数据类型和大小,以用来存放数据。对于采用C的基本数据类型(如整型、实型和字符型)和指针类型的变量,定义时只指定变量的数据类型就可以了;而对于采用数组类型的变量,定义时要同时指定变量的数据类型和大小;对于采用C的构造数据类型(如结构体、共用体)和枚举类型等的变量,在定义变量之前还要定义该变量的构造模型。

功能的执行部分是由若干个语句组成的。C语言的语句通常分为5大类:过程控制语句、表达式语句、函数调用语句、空语句和复合语句。过程控制语句(有9种)是用于完成一定的过程控制功能的,用C语言表示算法中的三种基本结构时,主要是靠这9种过程控制语句来完成。表达式语句是只有表达式构成的语句,主要用于数值计算和变量赋值。函数调用语句主要用于在主函数中实现对子函数的调用。空语句是不执行任何操作的语句,它的作用是延迟程序的继续执行。复合语句是上述几种语句的包装组合。

在C语句中通常含有各种类型的表达式。表达式是由运算符、运算对象(又叫做操作数)和标点符号组成的式子,其中的运算符是用于各种运算的符号,运算对象可以是常量,也可以是变量。表达式的作用是用于算术或逻辑计算,因此,表达式的值有数值型和逻辑型两种。C语言中没有专门用于表示逻辑型数据的变量,它借用数值型数据表示逻辑值,因此,逻辑运算的结果也是一个数值(1或0),1表示逻辑运算结果为逻辑真,0表示逻辑假。

当一个C程序大到函数、小到C语句甚至表达式都编写好之后,就可以上机实现了。而有些简单的C程序可以事先想好思路直接上机实现。

1.2.3 上机实现

C程序的上机实现过程就是编辑、调试和运行程序,从而获得问题结果的过程。在上机实现过程中,调试是一个非常重要的手段和环节,它反映了程序员发现问题和解决问题的能力。初学者常常会遇到下面一些情况:在调试程序时,一旦程序出错,或者看不懂编译不通时的错误提示,不知道程序错在哪里;或者编译通过,程序可以跑通,但是结果不对,不知道怎样调试来发现程序问题在哪里;或者知道问题在哪里,但是不知道出现问题的原因,找不到解决方法。这些情况会随着C语言知识的加深、不断地上机实践及不断地总结和学习而逐步减少。但愿这些问题成为激发我们进一步探寻C程序设计奥秘的动力。



1.3 简单 C 程序介绍

下面从最简单的 C 语言程序来分析 C 程序的结构。

【例 1.1】 在屏幕上显示一句话。

```
/* c01_01.c */
#include<stdio.h>          /* 文件包含部分 */
main()                    /* 主函数 */
{                          /* 函数体 */
    printf("Hello,everyone!\n");
}
```

程序运行结果为显示下面一行句子：

Hello,everyone!

这个 C 程序的第 1 行是一个注释语句，注释语句是用“/*”和“*/”包含起来的一段文字，用于对程序的功能进行说明，它在 C 程序中起到一个说明的作用，以便于程序的阅读和理解。

第 2 行是一个文件包含句子，#include 是 C 语言用来实现文件包含的命令（在以后章节详细介绍），stdio.h 是一个被包含的文件的名字，它是一个“标准输入输出”文件。把文件 stdio.h 包含到程序中的作用是可以在程序中实现数据的输入和输出，有了这个文件包含句子，第 5 行的数据输出才能被执行。

第 3 行是主函数名 main()，它和数学中的函数 $\phi(\xi)$ 类似，只是在括弧中省略了变量（在 C 程序中称之为参数）。

main() 叫做“main 函数”，又叫做“主函数”，每个 C 和 C++ 程序都必须有一个 main 函数，它是 C 和 C++ 程序的入口，因为每个 C 和 C++ 程序都是从 main 函数开始执行和结束的。

第 4 行的“{”和第 6 行的“}”及它们所包含起来的部分是 main 函数的函数体（即 main 函数的实体部分）。一个 main 函数的完整书写形式为

```
main()
{
    程序语句;
}
```

第 5 行是一个输出语句，printf() 是 C 语言中的输出函数，它的功能定义在 stdio.h 这个头文件中，所以程序开头用文件包含命令 #include 把它包含进来。printf() 函数的含义是按一定格式输出数据，这里的输出是指在显示器上输出。程序的运行结果即在显示器上输出：

```
Hello, everyone!
```

上面这个C程序在C语言中就叫做一个“源程序”，它是最简单的一个C程序。

下面再看一个C程序。

【例 1.2】 求三个整数 a、b、c 的和。

```
/* c01_02.c */
#include<stdio.h>           /* 文件包含部分 */
main ()                     /* 主函数 */
{                             /* 函数体 */
    int a,b,c;              /* 数据声明部分 */
    scanf ("%d,%d,%d",&a,&b,&c); /* 执行部分 */
    c=a+b+c;
    printf("sum=%d\n",c);
}
```

这个程序的结构与前面的程序一样，只是函数体部分的内容增加了而已。

程序中第 3~9 行是 main 函数，第 4~9 行是 main 函数的函数体部分内容，其中第 5 行是数据声明部分，用来定义程序中所用到的变量。程序中用到了三个变量 a、b、c，分别用于存放三个整数，三个整数之和的结果存放到变量 c 中（即将原来的一个整数值覆盖）。

第 6 行是一个数据输入语句，scanf() 是 C 语言中的输入函数，它的功能定义在头文件 stdio.h 中，所以程序开头要用文件包含命令 #include 把它包含进来。scanf() 函数的含义是按一定格式输入数据，这里的输入是指从显示屏上输入 a、b 和 c 的值。

第 7 行是一个表达式语句，用于计算三个整数的和。

第 8 行是一个数据输出语句，用于输出结果 c 的值。

运行此程序时，先输入三个变量 a、b、c 的值，若分别输入 1,2,3，程序的执行结果是在显示器上输出：

```
sum=6
```

在这个程序的第 1~6 行的后面，都有“/*”和“*/”包含起来的一段文字，这是程序的注释部分。

上面的程序还可以编写成如下的形式。

【例 1.3】 求三个整数 a、b、c 的和，采用主函数调用子函数的方法。

```
/* c01_03.c */
#include<stdio.h>           /* 文件包含部分 */
int sum(int x,int y);       /* 子函数的声明 */
main ()                     /* 主函数 */
{                             /* 函数体 */
    int a,b,c;              /* 数据声明部分 */
    scanf ("%d,%d,%d",&a,&b,&c); /* 以下是执行部分 */
}
```