

21

世纪高等职业教育
计算机技术规划教材

C语言 程序设计

解晨光 葛竹春 主 编
王旭 张瑞雪 郝春梅 副主编

齐景嘉 主 审



人民邮电出版社
POSTS & TELECOM PRESS

21 世纪高等职业教育计算机技术规划教材

C 语言程序设计

解晨光 葛竹春 主 编

王 旭 张瑞雪 郝春梅 副主编

齐景嘉 主 审

人民邮电出版社

北 京

图书在版编目 (CIP) 数据

C 语言程序设计 / 解晨光, 葛竹春主编. —北京: 人民邮电出版社, 2008.3
21 世纪高等职业教育计算机技术规划教材
ISBN 978-7-115-17251-8

I. C… II. ①解…②葛… III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 002959 号

内 容 提 要

本书是根据计算机软件水平测试内容, 结合全国计算机等级考试 (二级) 考试大纲, 按照任务驱动的思想来编写的。书中系统、全面地介绍了 C 语言程序设计的方法, 内容包括 C 语言的各种数据类型、运算符和表达式、各种语句、函数、编译预处理以及文件操作等, 并通过案例分析对 C 语言的语法要点进行详尽的阐述。每章后均附有大量的习题供读者练习。

本书内容选择合理、层次安排清晰和文字叙述简明便于教学与自学, 可作为高职高专、各类高等院校计算机程序设计的教材, 也可作为各种成人教育和计算机培训的教材或供自学者参考。

21 世纪高等职业教育计算机技术规划教材

C 语言程序设计

-
- ◆ 主 编 解晨光 葛竹春
副 主 编 王 旭 张瑞雪 郝春梅
主 审 齐景嘉
责任编辑 刘雁斌
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 15.75
字数: 381 千字 2008 年 3 月第 1 版
印数: 1—3 000 册 2008 年 3 月北京第 1 次印刷

ISBN 978-7-115-17251-8/TP

定价: 25.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

C 语言是应用广泛、最具影响力的程序设计语言，它概念简洁，数据类型丰富，运算符较多，表达能力强，使用灵活，既有高级语言的优点又具有低级语言的功能（能对硬件直接进行操作），因此既适合编写应用程序又适合编写系统程序，是一种理想的结构化程序设计语言，赢得了广大用户的喜爱。

C 语言是计算机专业的一门必修课，也是全国计算机等级考试及 NIT 考试的重要科目。本书是根据教育部《职业院校计算机和软件专业领域技能型紧缺人才培养培训指导方案》的精神，按照全国计算机等级考试（二级）考试大纲来编写的。本书是计算机专业技能型人才培养的入门课程，也是非计算机专业学生参加全国计算机等级考试（二级）的学习教材，侧重于应用，不强调学科知识理论的系统性，编写时按照任务驱动的教学思想、案例教学的格式组织材料。全书给出大量的案例，通过案例教学，努力将 C 语言作为解决实际问题的工具，强化学生的编程能力。

全书共分 12 章，具体内容如下。

第 1 章介绍程序设计的基本概念，程序设计的步骤以及 C 语言的特点、基本结构、上机步骤等。

第 2 章介绍 C 语言程序设计的初步知识，包括标识符、常量、变量的概念及运算表达式等。

第 3 章讲解算法的概念，语句的种类，输入输出函数的使用和简单的程序设计方法。

第 4 章讲解选择结构的程序设计，介绍关系运算符、逻辑运算符及其表达式，if 和 switch 语句的使用方法。

第 5 章介绍循环结构的程序设计，while 语句、do...while 语句和 for 语句等。

第 6 章介绍数组，讲述数组的定义、引用、初始化等。

第 7 章介绍函数的定义、调用和程序设计方法。

第 8 章介绍指针的概念，指针变量的定义、赋值和引用等。

第 9 章讲述编译预处理的方法，包括宏定义、文件包含和条件编译的使用。

第 10 章介绍结构体和共用体的特点、定义和使用。

第 11 章讲述位运算符及其运算功能。

第 12 章介绍文件的概念、文件的操作等。

每章之后都给出了大量各种类型的习题，供学习使用。

本书是作者在多年从事 C 语言程序设计教学实践的基础上编写的，全书由浅入深、结构清晰、案例丰富，主要有以下特点。

（1）从应用与实用的角度阐述 C 语言的基本概念、语法规则。

(2) 在内容的组织形式上, 采用案例教学和启发式教学的方法。

(3) 以案例为主线, 使读者既掌握基本知识和技能, 又掌握应用能力和编程能力。

本书由解晨光和葛竹春担任主编, 王旭、张瑞雪和郝春梅担任副主编, 齐景嘉担任主审。各章编写分工如下: 解晨光编写第 1, 9 章, 葛竹春编写第 3, 6 章; 王旭编写第 7, 8 章; 张瑞雪编写第 2, 5 章; 郝春梅编写第 4, 10 章; 王晓编写第 11 章; 刘明刚编写第 12 章。

由于编者水平有限, 书中难免有错误和不妥之处, 敬请读者不吝指正, 以便修订时改进。

编 者

2007 年 11 月

目 录

第1章 概述	1
1.1 C语言的发展及主要特点	1
1.1.1 C语言的发展史	1
1.1.2 C语言的主要特点	2
1.2 C程序设计相关知识	3
1.2.1 程序设计的基本概念	3
1.2.2 C程序结构	4
1.2.3 标准库简介	7
1.3 C语言上机过程	8
1.3.1 Turbo C集成环境的使用	8
1.3.2 程序调试方法	11
1.4 案例分析	12
本章小结	13
习题	13
第2章 C语言程序设计初步知识	15
2.1 标识符、常量和变量	15
2.1.1 标识符	15
2.1.2 常量和变量	16
2.2 简单数据类型	17
2.2.1 整型数据	17
2.2.2 实型数据	19
2.2.3 字符型数据	20
2.3 运算符和表达式	22
2.3.1 算术运算符及其表达式	22
2.3.2 自加、自减运算符	23
2.3.3 赋值运算符与赋值表达式	24
2.3.4 关系运算符和逻辑运算符	26
2.3.5 问号运算符及表达式	26
2.3.6 不同类型数据间的转换	27
2.3.7 逗号运算符及其表达式	27
2.3.8 运算符的优先顺序	28

2.4 案例分析	29
本章小结	31
习题	32
第3章 简单程序设计	35
3.1 算法	35
3.1.1 算法的概念	35
3.1.2 算法的描述	36
3.2 结构化程序设计	37
3.2.1 基本结构	37
3.2.2 程序设计举例	39
3.3 C 语言基本语句	40
3.4 C 语言中数据的输入输出	41
3.4.1 格式输入与输出	42
3.4.2 字符数据的输入输出	50
3.5 案例分析	51
本章小结	53
习题	54
第4章 选择结构程序设计	57
4.1 if 语句	57
4.1.1 单分支 if 语句	57
4.1.2 双分支 if 语句	57
4.1.3 多分支 if 语句	58
4.1.4 if 语句的嵌套	60
4.2 switch 语句	60
4.3 案例分析	62
本章小结	66
习题	66
第5章 循环结构程序设计	73
5.1 循环结构程序设计的概念	73
5.2 while 语句	73
5.2.1 while 语句的一般格式	73
5.2.2 while 语句的执行过程	74
5.3 do···while 语句	75
5.3.1 do···while 语句的一般格式	75
5.3.2 do···while 语句的执行过程	75
5.4 for 语句	76
5.4.1 for 语句的一般格式	76
5.4.2 for 语句的执行过程	77
5.5 循环结构嵌套	79

5.6 break 语句和 continue 语句	81
5.6.1 break 语句	81
5.6.2 continue 语句	82
5.7 案例分析	83
本章小结	86
习题	87
第 6 章 数组	93
6.1 一维数组	93
6.1.1 一维数组的定义	93
6.1.2 一维数组的初始化	94
6.1.3 数组元素的引用	95
6.1.4 一维数组案例分析	96
6.2 二维数组	99
6.2.1 二维数组的定义	99
6.2.2 二维数组的初始化	100
6.2.3 二维数组元素的引用	101
6.2.4 二维数组的应用	102
6.2.5 多维数组	103
6.3 字符数组与字符串	103
6.3.1 字符数组的定义和初始化	103
6.3.2 字符串	104
6.3.3 字符数组的输入/输出	105
6.3.4 字符串处理函数	106
6.3.5 案例分析	107
本章小结	111
习题	111
第 7 章 函数	116
7.1 模块化程序设计的概念	116
7.2 库函数	117
7.2.1 C 语言常用库函数	117
7.2.2 标准库函数的调用	118
7.3 函数的定义和调用	118
7.3.1 函数的定义	118
7.3.2 函数的调用	119
7.4 函数的返回值及其类型	123
7.5 函数调用时参数间的传递	126
7.5.1 变量、常量、数组元素作为函数参数	126
7.5.2 数组名作为函数参数	127
7.6 函数的嵌套调用	128

7.7 函数的递归调用	130
7.8 局部变量和全局变量	131
7.8.1 局部变量	131
7.8.2 全局变量	132
7.9 静态存储变量和动态存储变量	132
7.9.1 静态存储变量	133
7.9.2 动态存储变量	133
7.10 内部函数和外部函数	134
7.10.1 内部函数	134
7.10.2 外部函数	135
7.11 案例分析	135
本章小结	140
习题	140
第 8 章 指针	147
8.1 指针的基本概念	147
8.2 指针和指针变量	148
8.2.1 指针变量的定义	148
8.2.2 指针变量的赋值	149
8.2.3 指针变量的引用	149
8.2.4 指针变量作为函数参数	150
8.3 指针和数组	151
8.3.1 指针与一维数组	151
8.3.2 指针与二维数组	153
8.3.3 指针与字符串	155
8.3.4 指向数组的指针作为函数参数	156
8.4 指针数组	158
8.5 函数的指针和返回指针值的函数	160
8.5.1 函数的指针	160
8.5.2 返回指针值的函数	160
8.6 案例分析	161
本章小结	164
习题	164
第 9 章 编译预处理	170
9.1 宏定义	170
9.1.1 不带参数的宏定义	170
9.1.2 带参数的宏定义	172
9.2 文件包含	174
9.3 条件编译	175
9.4 案例分析	177

本章小结	179
习题	180
第 10 章 结构体与共用体	184
10.1 结构体类型	184
10.1.1 结构体类型的定义	184
10.1.2 结构体类型变量的使用	186
10.1.3 结构体数组	188
10.1.4 指向结构体变量的指针	191
10.1.5 链表	192
10.2 共用体	194
10.3 案例分析	197
本章小结	200
习题	200
第 11 章 位运算	208
11.1 位运算符	208
11.1.1 取反运算符	208
11.1.2 移位	209
11.1.3 “按位与”运算符	210
11.1.4 按位或运算符	211
11.1.5 异或运算符	211
11.2 位运算符的运算功能	212
11.3 案例分析	213
本章小结	215
习题	215
第 12 章 文件	218
12.1 文件概述	218
12.2 文件指针	219
12.3 文件的打开与关闭	219
12.3.1 文件打开函数 fopen	219
12.3.2 文件关闭函数 fclose	221
12.4 文件的读写	221
12.4.1 字符读写函数 fgetc 和 fputc	221
12.4.2 字符串读写函数 fgets 和 fputs	224
12.4.3 数据块读写函数 fread 和 fwrite	225
12.5 文件的随机读写	227
12.5.1 文件定位	227
12.5.2 文件的随机读写	228
12.6 案例分析	229
本章小结	231

习题.....	231
附录 I ASCII 字符表	234
附录 II C 库函数.....	235
附录 III C 语言常用出错信息	238
参考文献.....	242

第 1 章

概 述

教学目标

- 了解 C 语言的发展及主要特点;
- 掌握 C 语言的结构;
- 熟练掌握 C 语言的运行及其运行环境 Turbo C 的使用。

1.1 C 语言的发展及主要特点

计算机内部所能识别的只有“0”、“1”两个二进制数,指令在计算机内部实际上是由 0 和 1 组合起来形成的二进制编码。由二进制指令组成的计算机语言称为机器语言,但机器语言在使用过程中不易理解而且容易出错。为了克服机器语言编程的困难,人们研制出用助记符来代替机器语言编程,这种由助记符表示的计算机语言称为汇编语言。使用汇编语言编程克服了机器语言难记、不易阅读的缺点,但汇编语言同机器语言一样,也是面向特定计算机的,要求编程者精通计算机的内部结构,所以通用性差。

为普及计算机的应用,人们研究开发出更易掌握、更接近人类自然语言的计算机语言,这就是高级语言。用高级语言开发应用程序较机器语言和汇编语言要容易许多,效率也高。使用高级语言编写的程序在计算机内部都必须转换成机器语言。源程序可以通过“编译”或“解释”系统来实现这种转换,编译和解释系统是由专业开发人员设计好的系统程序,不同的高级语言有它自己的编译或解释系统。

早期使用的高级语言种类较多,如 BASIC 语言、COBOL 等,20 世纪 70 年代出现了 Pascal 等结构化程序设计语言。到了 20 世纪 80 年代,计算机语言跨入了面向对象的阶段,出现了 C++ 等语言。20 世纪 90 年代出现了网络化的编程语言——Java 语言。计算机语言的发展丰富了计算机的功能,使计算机在社会生活的各个方面发挥更大的作用。

1.1.1 C 语言的发展史

C 语言的前身是 ALGOL60,1963 年,英国的剑桥大学和伦敦大学首先将 ALGOL60 发展成 PCL,1967 年英国剑桥大学的 Martin Richards 将 CPL 改写成 BCPL;1970 年美国贝尔实验室的 Ken Thompson 将 BCPL 修改成 B 语言,并用 B 语言开发了 UNIX 操作系统。1972 年 Ken Thompson 与在开发 UNIX 操作系统时的合作者 Dennis Ritchie 一起将 B 语言改成了 C 语言。1978 年,Brain W. Kernighan、Ken Thompson 与 Dennis Ritchie 合著了著名的《The C

Programming Language》，该书介绍的 C 语言成为后来广泛使用的 C 语言版本的基础。由于 C 语言自身的优点，在其后的十几年中得到了广泛的使用，适用于不同机种和不同操作系统的 C 编译系统相继问世。1983 年美国国家标准局 (ANSI) 制订了 C 语言标准，这个标准不断完善，并从 1987 年开始实施 ANSI 的标准 C。1988 年，ANSI 公布了标准 ANSI C。

C 语言是一种面向过程的程序设计语言，其特点是：具有简单灵活的编程语言、丰富的数据结构和操作符、先进的结构化程序控制、良好的程序可移植性和高效率的目标代码。C 语言发展迅速，而且成为最受欢迎的语言之一。许多著名的系统软件，如 UNIX 操作系统等都是由 C 语言编写的。由于 C 语言的强大功能和各方面的优点逐渐为人们认识，到了 20 世纪 80 年代，C 语言开始进入其他操作系统，并很快在各类大、中、小型计算机上得到广泛的使用，成为当代最优秀的程序设计语言之一。

1987 年，Borland 公司首次推出了 Turbo C 1.0，使用了全新的集成化开发环境，通过下拉式菜单，将文本编辑、程序编译、连接和运行一体化，大大方便了程序的开发。

1988 年，Borland 公司又推出了 Turbo C 1.5 版本，增加了图形库和文本窗口函数库，Turbo C 2.0 是 1989 年推出的，新增加了差错功能，并可以在 TINY 模式下直接生成 .COM 文件。

进入 20 世纪 90 年代，Borland 公司又推出了面向对象的程序软件包 Turbo C++，继承、发展了 Turbo C 2.0 的集成化开发环境，并包含了面向对象的基本思想和设计方法。

目前，实用的 C 语言系统种类繁多。适用于微机的除 Turbo C 之外主要还有 Microsoft C (MSC)，Quick C (QC) 等，都可在 PC DOS 或 MS DOS 操作系统的支持下运行，此外其他操作系统，如 UNIX 或 XENIX 操作系统都有自己的 C 语言系统，其绝大多数的函数或宏都是标准的、通用的，但也可能有少数是非标准的。在一种操作系统上能正常运行的程序不一定能顺利移到其他操作系统上，对非标准之处需要做相应的修改。

1.1.2 C 语言的主要特点

一种语言之所以能存在和发展，并具有生命力，总是有其不同于其他语言的特点。归纳起来 C 语言主要有以下一些特点。

(1) C 语言基本组成部分紧凑简洁。C 语言只有 32 个标准关键字、44 个标准运算符以及 9 个控制语句，不但语言的组成精练简洁，而且使用方便灵活。

(2) C 语言运算符丰富，表达能力强。C 语言具有高级语言和低级语言的双重特点，其运算符包含的内容广泛，所生成的表达式简练灵活，有利于提高编译效率和目标代码的质量。

(3) C 语言数据结构丰富，结构化好。C 语言提供了编写结构化程序所需要的各种数据结构和控制结构，这些丰富的数据结构和控制结构以及以函数调用为主的程序设计风格，保证了 C 语言程序具有良好的结构化特征。

(4) C 语言具有结构化的控制语句。C 语言用函数作为程序模块以实现程序的模块化，是结构化的理想语言，符合现代编程风格。

(5) C 语言提供了某些接近汇编语言的功能，有利于编写系统软件。C 语言提供的一些运算和操作能够实现汇编语言的一些功能，例如它可以直接访问物理地址并能进行二进制位运算等，这为编写系统软件提供了条件。

(6) C 语言程序所生成的目标代码质量高。C 语言所生成的目标代码效率比用汇编语言描述同一个问题的效率高 20% 左右。

(7) C 语言程序可移植性好。C 语言所提供的语句中没有直接与硬件相关的语句,与硬件有关的操作,如数据的输入、输出等都是通过调用系统的库函数来实现的,而库函数本身不是 C 语言的组成部分,因此,用 C 语言编写的程序可以很容易地从一种计算机环境移植到另一种计算机环境中。

1.2 C 程序设计相关知识

1.2.1 程序设计的基本概念

1. 计算机语言和程序

计算机语言也称程序设计语言,即编写计算机程序所用的语言。计算机语言的种类非常多,总的来说可以分成机器语言、汇编语言和高级语言三大类。

计算机每运行一次,每一个步骤,都是按照已用计算机语言编好的程序来执行的,程序是计算机要执行的指令的集合,程序=算法+数据结构+程序设计方法+计算机语言和环境。而程序全部都是用人们所掌握的计算机语言来编写的,所以人们要控制计算机就一定要通过计算机语言向计算机发出命令。

计算机所能识别的语言只有机器语言,即由 0 和 1 构成的代码。但通常人们编程时,不采用机器语言,因为它非常难于记忆和识别。

目前通用的编程语言有两种形式:汇编语言和高级语言。

汇编语言的实质和机器语言是相同的,都是直接对硬件操作,只不过指令采用了英文缩写的标识符,更容易识别和记忆。它同样需要编程者将每一步具体的操作用命令的形式写出来。汇编程序通常由三部分组成:指令、伪指令和宏指令。汇编程序的每一句指令只能对应实际操作过程中的一个很细微的动作,例如移动、自增,因此汇编源程序一般比较冗长、复杂,容易出错,而且使用汇编语言编程需要有更多的计算机专业知识;但汇编语言的优点也是显而易见的,用汇编语言所能完成的操作不是一般高级语言所能实现的,而且源程序经汇编生成的可执行文件不仅比较小,而且执行速度很快。

高级语言是目前绝大多数编程者的选择。和汇编语言相比,它不但将许多相关的机器指令合成为单条指令,并且去掉了与具体操作有关但与完成工作无关的细节,例如使用堆栈、寄存器等,这样就大大简化了程序中的指令。同时,由于省略了很多细节,编程者也就不需要有太多的专业知识。

高级语言主要是相对于汇编语言而言,它并不是特指某一种具体的语言,而是包括了很多编程语言,例如目前流行的 VB, VC, FoxPro, Delphi 等,这些语言的语法、命令格式都各不相同。

高级语言所编制的程序不能直接被计算机识别,必须经过转换才能被执行,按转换方式可将它们分为如下两类。

解释类:执行方式类似于人们日常生活中的“同声翻译”,应用程序源代码一边由相应语言的解释器“翻译”成目标代码(机器语言),一边执行,因此效率比较低,而且不能生成可独立执行的可执行文件,应用程序不能脱离其解释器,但这种方式比较灵活,可以动态地

调整、修改应用程序。

编译类：编译是指在应用源程序执行之前，就将程序源代码“翻译”成目标代码（机器语言），因此其目标程序可以脱离其语言环境独立执行，使用比较方便，效率较高。但应用程序一旦需要修改，必须先修改源代码，再重新编译生成新的目标文件（*.OBJ）才能执行，只有目标文件而没有源代码，修改很不方便。现在大多数的编程语言都是编译型的，例如 C，Visual Foxpro，Delphi 等。

2. 算法和算法设计

所谓算法，是指为解决某一特定的问题，所给出的一系列确切的、有限的操作步骤。

程序设计的主要工作是算法设计，有了一个好的算法，就会产生质量较好的程序。程序实际上是用计算机语言所描述的算法。也就是说，依据算法所给定的步骤，用计算机语言所规定的表达形式去实现这些步骤，即为源程序。

3. 程序设计

程序设计一般按以下步骤进行。

(1) 确定数据结构：依据任务提出的要求，规划输入数据和输出的结果，确定存放数据的数据结构。

(2) 确定算法：针对所确定的数据结构确定解决问题的步骤。

(3) 编程：根据算法和数据结构，用程序设计语言编写程序，存入计算机中。

(4) 调试：在编译程序环境下，编译、调试源程序，修改语法错误和逻辑错误，直至程序运行成功。

(5) 整理源程序并总结资料。

1.2.2 C 程序结构

1. C 语言程序的结构特点

(1) C 语言程序是由函数构成的，一个源程序可以包含若干个函数，但必须有且只有一个函数为主函数 `main()`。

(2) 由大括号 `{}` 括起来的部分为函数体，函数体可以包含若干条 C 语言。每条语句必须以分号 “;” 结束。

(3) C 语言一般用小写字母书写。

(4) 程序可以加注释以增加可读性，其内容必须用 `/*...*/`。

(5) 除主函数外，C 语言系统还提供一些可直接使用的库函数，例如 `printf()`，`scanf()` 等。用户也可以根据需要，自己定义函数。

函数是 C 程序的基本结构，一个 C 程序由一个或多个函数组成，一个 C 函数由若干条 C 语句构成，一条 C 语句由若干基本单词组成。C 函数是完成某整体功能的最小单位，是相对独立的模块。简单的 C 程序可能只有一个主函数，而复杂的 C 程序则可能包含一个主函数和任意多个其他函数。所有 C 函数的结构都包括三部分：函数名、形式参数和函数体。

一个实现特定功能的 C 程序（一个或多个文件）应包含若干个函数，每个函数又是由若干个语句组成的。同其他高级语言一样，C 语言的语句用来向计算机系统发出操作指令。一个语句经编译后产生若干条机器指令。

C 程序的结构形式如下：

C 程序=预编译命令+外部定义+C 函数 1+C 函数 2+……+C 函数 n

C 函数=类型 函数名 (参数表)

参数说明

{

说明部分

C 语句

}

在 C 语言中, C 语句是指可执行语句, 用它来实现数据的操作。

2. C 语句格式分类

C 语句按形式可分为如下 3 种。

(1) 单语句: 即一条可执行语句, 以分号 “;” 结束。

例如:

```
i--;
```

(2) 复合语句 (也称为分程序): 整体上用一对花括号括起来, 语法等效于一个单语句。

其一般形式为

```
{
```

说明语句

语句 1

⋮

语句 n

```
}
```

例如:

```
{int x=100;
```

```
x--;
```

```
printf("&d\n", x);
```

```
}
```

(3) 空语句: 只有一个分号的语句, 无语句内容。

例如:

```
;
```

3. C 语句构造分类

C 语句按构造可分为如下两种。

(1) 表达式语句: 由表达式加分号 “;” 构成。

例如:

```
x=100;
```

(2) 控制语句: 既可以是单语句, 也可以是复合语句, 主要有以下几种。

```
If    else
```

```
for
```

```
while
```

```
do    while
```

```

continue
break
switch
goto
return

```

这些语句的作用是控制 C 程序的流程，即决定 C 语句的执行顺序。
通过下面的例子可以了解 C 语言的基本程序结构。

【例 1-1】简单的 C 语言程序。

```

main()                /*主函数*/
{
    int x, y, s;        /*定义变量*/
    x=3;b=7;s=(x+y)/2; /*赋值语句*/
    printf("s=%d", s); /*输出语句*/
}

```

运行结果为

```
s=5
```

说明：

- (1) main()是主函数，C 语言总是从主函数开始执行。一个 C 程序有且仅有一个主函数。
- (2) 一对花括号是函数的开始与结束，即为函数体的界限符。
- (3) 第 3 行是定义变量语句。经定义后，编译时为这些变量开辟内存空间，就可以使用这些变量。C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。
- (4) 第 4 行有 3 个赋值语句。C 程序允许多条语句写在同一行，但要用冒号分开。
- (5) 第 5 行是输出语句，即在屏幕上输出 s 的值，引号中的字串原样输出，%d 是输出的格式控制符，表示十进制整数类型，在执行输出时将此位置上代以一个十进制整数值。
- (6) printf()是 C 程序的库函数。格式输出函数 printf()的功能是按指定的格式输出数据，其一般的调用格式为 printf("格式控制字符串", 参数表)。其中，格式控制字符串用双引号括起来，用来规定输出格式，例如%c 用来输出字符型数据；参数表中包含零个或多个输出项，这些输出项可以是实数、变量式表达式，多个输出项之间用逗号隔开。
- (7) /*...*/表示注释部分，注释只是给人看的，对编译和运行不起作用。注释可以加在程序中的任何位置。

【例 1-2】一个复杂的 C 程序。

```

#include<math.h>                /*inclue 为文件包含命令*/
#include<stdio.h>
main()                          /*主函数*/
{
    double a, b;                /*定义变量*/
    printf("input number : "); /*输入字符串 "input number : " */
    scanf("%lf", &a);           /*输入变量 a 的值*/
    b=sqrt(a);                  /*求 a 的平方根，并把它赋给变量 b*/
}

```