微电脑学习与应用丛书·翁瑞琪 主编

C 程序设计及应用

张洪儒 滕建辅 郭继昌 编著

WEIDIANNAO XUEXI YU YINGYONG CONGSHU

国防工业出版社
GFGYCBS

C 程序设计及应用

张洪儒　滕建辅　郭继昌　编著

国防工业出版社
·北京·

图书在版编目(CIP)数据

C 程序设计及应用/张洪儒等编著．—北京：国防工业出版社，1997.1
（微电脑学习与应用丛书/翁瑞琪主编）
ISBN 7-118-01665-9

Ⅰ.C… Ⅱ．张… Ⅲ.C 语言-程序设计 Ⅳ.TP312C

中国版本图书馆 CIP 数据核字(96)第 18373 号

国防工业出版社出版发行
（北京市海淀区紫竹院南路 23 号）
（邮政编码 100044）
三河腾飞胶印厂印刷
新华书店经售
*
开本 787×1092 1/16 印张 19¾ 453 千字
1997 年 1 月第 1 版 1997 年 1 月北京第 1 次印刷
印数：1—4000 册 定价：24.70 元

丛书总序

当今人类正步入信息化时代，人们所面临的信息处理量之大、信息处理复杂程度之高，以致单靠人脑去处置已不能胜任。因此，人类迫切要求辅助脑力劳动的工具。电子计算机以其在信息处理上的独特优点而充当了人类脑力劳动的辅助工具，或者说，它起到了人类脑力活动在体外延长之作用。因此，人们美称电子计算机为电脑。

微型计算机（微电脑）的出现使计算机的应用得到极大普及。至今，微电脑已成为人类活动中不可缺少的有力助手。学习和应用微电脑是当今必然的趋势。为适应当今时代普及微电脑应用的需要，决定编写《微电脑学习与应用丛书》。

本丛书以教会如何使用微电脑为目的，帮助读者步入微电脑应用世界。

本丛书可供具有高中以上文化程度的微电脑使用者和广大爱好者自学，可用作微电脑应用短训班的培训教材，也可供大专院校广大学生学习计算机应用时参考。

本丛书由天津大学技术经济与系统工程系翁瑞琪教授主编，参加本丛书编写的是长期从事计算机教育和计算机科研工作的教学经验丰富、实践能力强的教师和专家。

本丛书的编辑出版得到国防工业出版社的大力支持，在此表示衷心的感谢。

期望本丛书的出版能为我国计算机应用范围的扩大和应用水平的提高起到促进作用。

热诚欢迎有关专家和广大读者对本丛书的编辑出版提出建设性的建议和改进意见。

翁瑞琪

前　言

本书是翁瑞琪教授主编的《微电脑学习与应用丛书》之一。

C 语言是 70 年代初出现的一种新型通用程序设计语言，它是当今世界上最有影响的程序设计语言之一。C 语言具有语言简洁、灵活、表达能力强、代码质量高、可移植性好等优点，既具有高级语言的特点，又具备低级语言的功能。它能代替汇编语言直接与机器的硬件打交道，这是其它许多高级语言所不能做到的。C 语言的这种双重性，使它既可用来编写系统软件，又可用来开发应用软件，从而使它成为一种成功的结构化程序设计语言。

随着计算机技术的发展，现在 C 语言已经风靡全球，成为世界上应用最广泛的新型的现代主流程序设计语言之一，成为微型机、工作站、小型机以及大型机等各类计算机共同使用的语言。C 语言广泛应用于系统软件、应用软件、数据处理和数值计算等各个领域，深受广大计算机工作者的喜爱。

本书全面、系统地介绍了 C 语言程序设计的基本概念和方法，包括 C 语言的运算符、数据结构、程序结构、函数、文件以及编译预处理等。为了使读者在学习 C 语言的基础上进一步学习 C＋＋语言，本书第十二章简单介绍了 C＋＋语言。由于 C 语言使用灵活、规则繁多、涉及到的概念比较复杂，不少读者感到学习有些困难。为了减少学习的难度，本书在体系结构和内容安排上，力求做到由浅入深、由简到繁、循序渐进，把难点分散在不同的章节中。每介绍一个新的概念和规则都给出例题，以便于读者对概念和规则的理解，同时也方便读者上机操作练习。在例题的选配上均不涉及复杂的数学和物理背景，主要用于说明语法规则和掌握程序设计的基本要求与技巧。对于初学者容易混淆的概念和常见编程错误也做了分析和提示。

参加本书编写的有张洪儒、滕建辅和郭继昌同志。第一、二、三、四、五、六章由张洪儒同志编写，第七、八、九、十章由滕建辅同志编写，第十一、十二、十三章和附录由郭继昌同志编写，最后全书由张洪儒负责修改定稿，翁瑞琪教授审定。

由于编者水平有限，书中错误和不当之处在所难免，敬请读者批评指正。

内容简介

《微电脑学习与应用丛书》由天津大学翁瑞琪教授主编。本丛书以教会如何使用微电脑为目的，帮助读者步入微电脑应用世界。

本书是该丛书之一，侧重介绍C语言及其程序设计。C语言是一种结构化的通用程序设计语言，是目前世界上应用非常广泛的新型的现代主流程序设计语言。本书全面、系统地介绍了C语言的基本概念、语法规则和利用C语言进行程序设计的方法。

本书内容丰富，取材广泛，并配有不同类型的例题和习题。根据初学者的特点，在内容的安排上由浅入深、循序渐进，叙述上条理清楚、通俗易懂，便于初学者自学使用。

本书可供具有高中以上文化程度的微电脑使用者和广大爱好者自学，又可作为大专院校非计算机专业和各类计算机应用培训班的培训教材，也可供大专院校广大学生参考。

目　录

第一章 引 论

计算机的出现是近代重大科学成就之一，它的出现有力地推动了其它科学技术的发展，在科学研究、工农业生产、国防建设以及社会生活等方面，都获得了越来越广泛的应用。计算机系统由硬件和软件组成，只有硬件和软件同时具备，计算机才能自动、快速、连续地工作，完成各种各样的工作任务。任何计算机都只能执行程序安排它做的事情，离开了程序，计算机就无法发挥作用。

1.1 程序设计语言概述

人们设计和制造计算机的目的是使计算机按照人们的意图去工作，要做到这一点就必须事先按照人们的某种意图编制好一组指令，送入计算机中，用这组指令控制计算机的动作。通常我们把这样一组指令称为程序，编写程序的过程称为程序设计，而编写程序使用的语言称为程序设计语言。人与人对话需要语言，人与计算机对话也需要一种特定的语言，这种语言叫计算机语言，换句话说，计算机语言是人与计算机进行信息交换的工具。

1. 机器语言

计算机只能识别由 0 和 1 组成的二进制代码，由一组 0 和 1 按照一定的规则组合起来就构成一条指令，它控制计算机执行相应的操作。由于这种二进制的指令是面向计算机的，因此称为机器指令。若干条机器指令的集合称为机器语言。

由于计算机指令全部是由二进制代码组成，并且它是严格按照所用计算机的指令系统规定的格式写成的，所以送入计算机后，可以直接被计算机接受和执行。但是由于每一种计算机都有自己特定的指令系统，因此程序设计人员必须非常熟悉所用计算机的指令系统以及内存单元的分配，这就使程序设计工作既繁重又困难，且所编程序直观性差，不利于阅读和修改。同时由于所编程序依赖于某一机器，无通用性，这就给用户造成很大困难。

2. 汇编语言

为了克服机器语言的上述缺点，随着计算机科学的发展，人们设计出了一种改进了的语言——汇编语言，用来进行程序设计。汇编语言是一种符号语言，它是用便于人们理解和记忆的符号来代替机器语言中的二进制代码。用汇编语言编写的程序，其指令的操作码和操作数地址全部用符号表示，这大大方便了记忆，比用机器语言编写的程序直观、易读。但是由于计算机只能识别由 0 和 1 组成的二进制代码，而汇编语言采用助记符来表示操作码和操作数的地址，因而把用汇编语言编写的程序送入计算机后，计算机不能识别和执行，必须把由汇编语言编写的程序翻译成计算机系统的特定机器指令，才能识别和执行，担任此翻译工作的程序叫汇编程序。因为每个汇编语言语句和特定的机器指令之间存在一一对应的关系，所以汇编语言被看作是低级语言。

采用汇编语言虽然可以改善程序的直观性，但是工作量仍然很大，同时，由于不同计算机的汇编语言指令形式也不尽相同，所以程序设计人员也必须在编写程序之前充分了解所用计算机的指令系统，才能用汇编语言编写程序，而且所编写的程序只能在该型号的计算机上使用，不能用于其它型号的计算机，即不能移植。

3. 高级语言

为了克服汇编语言的缺点，随着计算机的发展，科学工作者在50年代中期又研制成功了高级语言。高级语言又称为程序设计语言或算法语言。它摆脱了机器语言和汇编语言面向机器、依赖于具体机器的缺点，人们不必懂得计算机的具体结构和工作原理，不需要考虑具体机器的指令系统，只需考虑所要求解决的问题本身。这给广大计算机使用者提供了很大方便，为计算机的推广使用扫清了障碍。

自从50年代中期高级语言问世以来，科学工作者已先后研制出几十种乃至上百种高级语言。目前国内外通用的高级语言有十几种，它们分别适用于各个不同的领域。例如：

FORTRAN　（适用于科学计算）
BASIC　（小型会话式语言）
COBOL　（适用于数据处理）
PASCAL　（结构化程序设计语言）
C　（结构化程序设计语言，特别适合于编写系统软件）
LISP　（用于人工智能方面的语言）

其中FORTRAN、BASIC和COBOL是面向过程的最早使用的高级语言，C和PASCAL是典型的结构化语言，它继承了面向过程语言的模块化等优点，对程序结构作了改进。

由于高级语言非常接近人们常用的自然语言和数学表达式，所以程序设计非常方便。例如，要计算D=A×B+C，如果要用C语言或FORTRAN语言编写程序，只要写成D=A*B+C即可。由此可见，其形式和数学表达式一模一样。使用高级语言的另一个优点是，由于大多数语言都有国际统一标准，所以编写的程序可以不做修改或者稍加修改就能在任何一台配有此种语言的计算机上运行，从而大大提高了程序的通用性。

用高级语言编写的程序称为源程序。由于它完全脱离了具体的计算机系统，按照统一的规定和格式进行编写，因而比用汇编语言编写的程序更便于阅读和修改。但它与用汇编语言编写的程序一样，不能直接为计算机所识别和执行，必须事先把它翻译成计算机所能理解的形式，即翻译成计算机的特定指令，然后才能在计算机上执行。一般我们把在计算机内担任此翻译任务的程序称为编译程序，而把翻译的过程称为编译。

1.2　C语言的发展和ANSI标准

1960年出现的ALGOL 60(AlGOrithmic Language 60)是一种面向问题的高级语言，与计算机硬件相距甚远，所以不适宜编写系统软件。1963年英国的伦敦大学和剑桥大学在ALGOL 60的基础上推出了CPL(Combined Programming Language)语言，CPL语言是仿照ALGOL 60而设计出来的语言，它接近硬件一些，但仍过于庞大和复杂，所以影响了它的实现和广泛应用。1967年英国剑桥大学的Martin Richards推出了BCPL(Basic

Combined Programming Language)语言，BCPL 语言对 CPL 语言进行了精减，并保持了它的基本特点，它是一种没有数据类型，或者说只有一种数据类型——机器字的单一数据型语言。1970 年美国贝尔实验室的 K Thompson 以 BCPL 语言为基础，又作了进一步简化，设计出了很简单且很接近硬件的 B 语言(取 BCPL 的第一个字母)，并在 PDP-7 机上实现了用 B 语言写的第一个 UNIX 操作系统。1971 年又在 PDP-11/20 机上实现了 B 语言，并写了 UNIX 操作系统。但 B 语言过于简单，语言能力有限，只能用于特殊用途而不能解决一般常见的许多问题。为此，1972 年美国贝尔实验室的 D. M. Ritchie 在 PDP-11 机上实现了 C 语言(取 BCPL 的第二个字母)。它以 B 语言为基础，吸收了 B 语言中合理而有效的部分，系统地引进了多种基本数据类型即字符、整数和符点数，并导出数组、指针、结构、共用体和函数。既保持了 BCPL 和 B 语言的精练、接近硬件的优点，又克服了它们过于简单、数据无类型等缺点。1973 年，K. Thompson 和 D. M. Ritchie 合作用 C 语言重写了 UNIX 操作系统，实现了 UNIX 第五版。它比原先的版本更易于理解、修改和扩充，并增加了多道程序设计功能，开创了 UNIX 系统发展的新局面。C 语言的发展历史可用图 1—1 来表示。

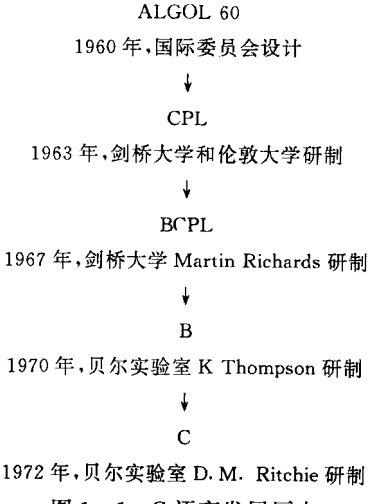

图 1—1　C 语言发展历史

从 C 语言发展历史来看，它属于 ALGOL 语言族系。C 语言在发展过程中与 UNIX 相辅相成，C 语言开创了 UNIX 的新局面，使其获得了巨大成功；UNIX 促成了 C 语言的大发展，使其得到了迅速推广，从而形成一对繁荣与共的孪生兄弟。1978 年以后，C 语言逐渐独立于 UNIX 系统和 PDP-11 而蓬勃发展。以 1978 年发表的 UNIX 第七版本中的 C 编译程序为基础，B. W. Kernighan 和 D. M. Ritchie(合称 K&R)合作出版了《The C Programming Language》，由它产生了 C 语言版本的基础，称为标准 C。1983 年，美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展，1988 年，K&R 按照 ANSI C 重写了他们的著作《The C Programming Language》。目前人们常将 1978 年的标准 C 称为旧标准，将 ANSI C 称为新标准。

C 语言之所以能在多种型号计算机上移植使用，并引起广泛注意是由于它在计算机语言领域中做出了如下贡献：

①C 语言十分有效地描述并实现了 UNIX 操作系统。整个 UNIX 大约是 13000 行程序，其中仅有 1000 行是面向硬件的汇编程序，即 UNIX 的 90%以上是用 C 语言编写的。另外，C 语言多数运行在 UNIX 系统上。因此，C 语言和 UNIX 操作系统联系紧密，相辅相成。

②C 语言实现了描述自身的语言编译程序。C 语言编译程序近 10000 行，程序的 90%用 C 语言编写。另外，C 语言已成功地成为 PASCAL 等语言编译程序的系统程序设计语言。这一贡献说明 C 语言、C 语言编译程序、UNIX 操作系统以及由 C 语言编制的其它系统实用程序具有高度的可移植性。

③C 语言将规模小、灵活性大、描述力强、可读性好等特点集于一身，成为计算机专业很理想的教学语言。众所周知，PASCAL 语言是世界上第一个结构化程序设计语言，曾被认为是计算机专业的比较理想的教学语言，在数据结构等课程中一般用 PASCAL 语言举例，但 PASCAL 语言难以推广到各实际应用领域，到目前为止，基本上只是教学语言。C 语言也是理想的结构化语言，且描述能力强，同样适于教学，而且操作系统课程多结合 UNIX 讲解，而 UNIX 与 C 语言不可分，因此，C 语言有可能取代 PASCAL 而成为被广泛使用的教学语言。

现在 C 语言已风靡全球，成为世界上应用非常广泛的新型的现代主流程序设计语言，广泛应用于系统软件(如操作系统、编译系统等)、应用软件(如图形处理等)、数据处理(如企业管理等)以及数值计算等各个领域。

1.3 C 语言的特点

C 语言之所以具有强大的生命力，成为国际上公认的最重要的少数几种通用程序设计语言之一，固然与它产生的环境及历史背景有关，但起决定作用的是它自身的特点。概括起来 C 语言的主要特点如下：

①语言简洁、紧凑。C 语言是一种小型语言，经精心选用，它总共只有 32 个关键字，9 种控制语句，压缩了一切不必要的成分，基本组成部分紧凑，表示方法简洁。C 语言虽小但却被公认为是强有力的语言，它使用一些简单、规整的方法就可以构造出相当复杂、功能很强的数据类型、语句和程序结构。例如在 PASCAL 语言中，用 BEGIN...END 作复合语句或函数体的“括号”，在 C 语言中用花括号{}来代替 BEGIN...END。又例如用 i++表示 i=i+1；用 i+=3 表示 i=i+3 等。另外，从语言内部实现角度来看，C 语言本身不提供输入/输出机构，即没有在一般高级语言中常用的 read，write 语句和固定的文件存取方式，其输入输出通过调用 scanf，printf 等函数来实现。

②C 语言提供了多种运算符。C 语言有 34 种运算符和 15 个等级的运算优先顺序，以及由此产生的实用、简洁的表达式。C 语言把括号、赋值、变量的自增自减操作、变量的地址获取运算等都作为运算符处理。众多的操作符和由此构成的表达式都很简短，编译处理也统一简单，从而使 C 语言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它高级语言中难以实现的运算。

③有利于实现结构化程序设计。C 语言是一种结构化语言，它提供了编写结构化程序所需要的基本控制结构语句，例如先测试后执行的循环语句“while，for”；先执行后测试

的循环语句“do...while”;条件语句“if...else...”;多路分支选择控制语句“switch”。C语言用函数作为程序设计的基本单位,实现程序的模块化。包含程序的源文件,可以分割为多个源文件,可以分别对各个源文件进行编译,再通过连接得到可执行的目标程序文件,提供多种存储属性,使得数据可以按需要在相应的模块中共享或隐藏。

④C语言具有丰富的数据结构。它包含了现代化语言所要求的各种数据结构: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,能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。尤其是引入指针类型数据,使C语言描述力大大增强,使用起来更加灵活、多样。

⑤C语言具有灵活、实用、方便的特点。C语言语法限制不太严格,程序设计自由度大,这使用户在程序设计中有更大的主动性。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,如整型量与字符型数据以及逻辑型数据可通用。另外语言格式自由,限制少,编写程序较自由。C语言允许程序编写者有较大的自由度,因此放松了语法检查,这就给程序编写者提出了较高的要求。

⑥用C语言编写的程序可移植性好。汇编语言由于依赖于硬件,所以根本不可移植。一些高级语言,如FORTRAN等的编译程序也不可移植,它们从一种机器搬到另一种机器,基本上都要根据国际标准进行重新编写。而C语言目前在不同机器上出现,大部分都是C语言编译移植得到的,统计资料表明,不同机器上的C编译程序80%的代码是公共的。C语言的编译程序便于移植,就使一个环境上用C语言编写的许多程序,从该环境下不加修改或稍加修改就可以搬到另一个完全不同的环境上运行。

⑦C语言生成的目标代码质量高。相对汇编语言而言,许多高级语言生成的目标代码质量很低,所以,迄今汇编语言仍是编写系统软件的主要工具。但是,许多试验表明,针对同一问题,用C语言编写的程序,生成代码的效率比用汇编语言写的代码低10%~20%。由于用C语言描述问题比用汇编语言迅速,工作量小,可读性好,易于测试、修改和移植,而在代码质量上可与汇编语言相媲美,因此,C语言迅速成为人们进行程序设计和软件开发得心应手的工具。

⑧C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。C语言的这种双重性,使它既可用来编写系统软件,又可用来开发应用软件,成为一种成功的通用程序设计语言。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”。

⑨C语言提供编译预处理的功能,这是它与其它高级语言的一个重要区别。C语言提供的预处理功能主要有以下三种:宏定义、文件包含和条件编译,分别用宏定义命令、文件包含命令和条件编译命令来实现。编译预处理是C编译系统的一个组成部分。预处理程序对这些特殊命令的分析处理是在编译系统对C程序其它部分编译之前进行的,然后将预处理的结果和源程序一起再进行通常的编译处理,以得到目标代码。C语言预处理程序和有关语句能够帮助程序员编写易读、易改、易于移植、便于调试的程序,对于结构化程序设计也提供了很大的帮助。

C语言的优点很多,但和其它一些程序设计语言一样,它也有其弱点,如运算符的优先级较多,有些还与常规约定不同,不便记忆;各种C语言版本之间略有差异,缺乏统一

的标准；C 语言不是强类型的语言，它在强调灵活、高效的同时，一定程度上牺牲了某些安全性，如类型检验太弱，转换比较随便等。因此，C 语言对程序设计人员提出了较高的要求。但是，C 语言的优点远远超过了它的弱点，这些优点使 C 语言具有强大的生命力，使学习和使用 C 语言的人越来越多。

1.4 C 程序结构

C 语言程序简称 C 程序一般由若干个函数组成。函数是完成某个算法的一个程序段，它是 C 语言的基本构成单位。组成一个程序的若干个函数可以保存在一个或几个源程序文件中，这些文件都以 .c 为扩展名。一个程序必须有且只能有一个名为 main 的函数，它是该程序的主函数，运行 C 程序时，总是从 main 函数开始执行。

下面举一个简单的 C 程序的例子，通过这个例子使初学者对 C 程序的结构有一个初步的了解。

例 1 写一个程序，输入 a 和 b 两个数，求出其中较大者并与第三个数求和。

```
main(  )                                  /* 主函数 */
{
      int a,b,c,d,sum;                    /* 定义变量 */
      scanf("%d,%d",&a,&b);               /* 输入 a 和 b 的值 */
      c=max(a,b);                         /* 调用 max 函数 */
      d=6;
      sum=c+d;
      printf("max=%d\n",c);               /* 输出 c 的值 */
      printf("sum is %d",sum);
}

int max(x,y)                              /* 定义 max 函数，函数返回值为整型 */
int x,y;                                  /* 对形参 x,y 作类型定义 */
{
      int z;
      if(x>y)
            z=x;
      else
            z=y;
      return(z);                          /* 将 z 的值返回调用处 */
}
```

上面是一个完整的 C 程序，经编译、连接、执行后，可以得到运行结果。

说明

①本程序包括两个函数，第1行开始的是主函数，其中main是函数名，它表明本函数是该程序的主函数。紧跟其后的括号"( )"为函数标志，告诉编译程序这是一个函数。一个函数在其名字之后一定要有一对圆括号，圆括号中的参数可有可无，这里主函数没有参数。

第11行开始的是被调用的函数max，max是函数名，max左边的int是对该函数函数返值的类型说明，这里说明函数返回值为整型。一对括号中的x、y为形式参数，第12行定义形参x、y为整型。第15行的if语句是一个条件语句。max(x,y)函数的作用是将x和y中较大者的值赋给变量z。第19行return语句将z的值返回给主函数main，返回值是通过函数名max带回到main函数的调用处。

②程序第3行是变量定义部分，说明a、b、c、d、sum是整型(int)变量。

③程序第4行中的scanf是输入函数的名字(scanf和printf都是C语言提供的标准输入输出函数)。"%d"是输入输出的格式字符串，用来指定输入输出时的数据类型和格式(详见第二章)，"%d"表示十进制整数类型，此处指定输入的两个数据按十进制整数形式输入。&a和&b中的&的含义是取地址。此scanf函数的作用是：将两个数值分别输入到变量a和b的地址所标志的单元中，也就是输入给变量a和b。这种形式是与其它语言不同的。它相当于PASCAL语言中的read(a,b)。关于scanf函数，详见第二章。

④程序第5行是一个函数调用语句，是主函数main调用名为max的函数。一对圆括号中的a和b是实际参数，调用时先将实际参数a和b的值分别传递给max函数中的形式参数x和y，执行max函数后得到一个返回值，通过函数名max把返回值带回到main函数的调用处并赋给变量c。

⑤程序第6行和第7行是两个赋值语句，使d的值为6，并把c加d的值赋给sum。

⑥程序第8行和第9行也是函数调用，printf是C语言提供的标准输出函数，printf是输出函数的名字，函数中双引号内的"max=%d\n"，表示输出时max=原样输出，"%d"将由圆括号中最右端的c的值取代，"\n"表示输出换行。

⑦若输入7和4给a和b，则程序运行情况如下：

```
7,4↙                (↙表示回车)
max=7               (输出c的值)
sum is 13           (输出sum的值)
```

通过例1说明C语言的结构如下：

①C语言程序的基本单位是函数。一个C程序由一个或多个函数构成，其中必须有且只能有一个名为main的主函数。主函数可以调用其它函数，被调用的函数可以是系统提供的库函数，也可以是用户自己编写的函数。C语言的库函数十分丰富，标准C提供一百多个库函数，Turbo C和MS C 4.0提供三百多个库函数。主函数与被调用函数在程序中的先后位置是无关紧要的，可以任意放置。

②一个函数由函数头和函数体两部分组成。

函数头也叫函数首部，它是函数的说明部分。包括函数名、函数值的类型、函数参数名和形式参数类型。如例1中max函数的函数头为：

int	max	(x,y)
↓	↓	↓
函数值类型	函数名	函数参数

int	x,y;
↓	↓
形参类型	形参

一个函数名后面必须跟一对圆括号,它是函数的标志。函数参数可有可无。

由一对花括号{…}括起来的部分组成函数体。花括号{和}分别表明函数体的开始和结束,它的作用类似于 PASCAL 语言中的 BEGIN 和 END。如果一个函数内有多个花括号,则最外层的一对花括号{ }为函数体的范围。

函数体一般包括:变量定义(如 max 函数中的 int z)和由若干语句组成的可执行部分。在特殊情况下,可以既无变量定义也没有可执行部分,只有一对花括号构成一个空函数体,它与函数头一起组成一个空函数,什么也不干,这也是合法的。

③每个语句和数据定义的最后必须有一个分号,分号是 C 语句的必要组成部分,分号在 C 语言中是语句的终止符,而不是一般的分隔符,所以不能省略。即使是程序中最后一个语句也应包含分号,这是和 PASCAL 语言不同的。

④可以用 /*…*/ 对 C 程序中的任何部分作注释。注释可在一行内写完或分多行写完。注释的作用是帮助阅读和理解程序,编译时对注释行不编译。注释在 C 语言中是很有用的,一个好的程序设计者应该在程序中多使用注释来说明整个程序的功能和部分语句的功能以及算法等。注释可以用英语、汉语、汉语拼音或数学表达式等任意形式表示。

⑤C 语言是用小写字母按自由格式书写的程序。通常一个语句写成一行,也可以一个语句写成几行,一行内也可写多个语句。语句没有行号,也没有像 FORTRAN 或 COBOL 那样严格规定书写格式的限制。为了便于阅读和修改,在一个 C 程序中,某些行可以缩格书写,错落有序,但要注意上下对齐。

⑥在 C 语言中,变量要先定义,然后才能引用。如例 1 中的 int a,b,c,d,sum;说明 a,b,c,d,sum 是整型变量。也可以在对变量定义的同时对其初始化,如 int d=6。

⑦C 语言本身没有输入输出语句,输入和输出的操作是由库函数 scanf 和 printf 等来完成的。

1.5 C 程序的开发过程

用户根据具体问题的需要,选用 C 语言来编写程序,大体要经过构造数学模型、编写源程序、编辑、编译、连接和运行等步骤。现以 UNIX 操作系统为例,说明 C 程序的开发过程。