面向21世纪课程教材
普通高等学校精品课程教材

C语言程序设计

C YUYAN CHENGXU SHEJI

李长云 廖立君 王平 童启 王志兵 编著

国防工业出版社
National Defense Industry Press

面向21世纪课程教材
普通高等学校精品课程教材

C语言程序设计

李长云　廖立君　王平　童启　王志兵　编著

国防工业出版社
·北京·

内容简介

本书的编写既充分考虑C语言重要语法知识点的全面性，又突出学生程序开发的实践能力和工程能力的训练。内容上以一明一暗两条线索来组织材料。明线是C语言语法知识点，从简单数据结构、简单控制结构到复杂数据结构、复杂控制结构，循序渐进地展示C语言特性。暗线则是两个实际应用（科学计算器和学生成绩管理系统）贯穿全书始终，这两个应用涵盖了排序、查找、删除等常见程序算法。针对这两个应用，采用螺旋式的讨论方法：先进行简要介绍，然后在后续章节中再进行一次或多次介绍，每次逐渐增加一些细节内容，由浅入深，相互呼应。

本书适用对象是高等院校计算机专业及非计算机专业的师生，计算机等级考试培训班师生，广大C语言自学者。本书电子教案、扩展练习及其它参考资料请参见网站http://58.20.192.206/ec/C16/Course/Index.htm。

图书在版编目(CIP)数据

C语言程序设计/李长云等编著. —北京：国防工业出版社，2011.8重印

面向21世纪课程教材

ISBN 978-7-118-07266-2

Ⅰ.①C... Ⅱ.①李... 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(2011)第009826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100048）

北京嘉恒彩色印刷有限责任公司印刷

新华书店经售

开本 787×1092 1/16 **印张** 19 **字数** 520千字

2011年8月第2次印刷 **印数** 6001—9000册 **定价** 39.00元

（本书如有印装错误，我社负责调换）

国防书店：(010)68428422 发行邮购：(010)68414474

发行传真：(010)68411535 发行业务：(010)68472764

前　言

程序设计是学习计算机应用与软件开发的基础，如果只会简单的计算机操作，不了解软件开发的实质，就无法从根本上了解计算机的工作原理，也很难应对信息技术日新月异的飞速发展。C 语言作为一种通用的程序设计语言，为大多数高校程序设计课程的选用语言。C 语言数据类型丰富，运算灵活方便，可用于编写高效简洁、风格优美的应用程序以及计算机系统程序。用 C 语言编写的程序，具有运算速度快、效率高、目标代码紧凑、可移植性好等特点。

目前，C 语言程序设计的教材，大多以 C 语言语法知识点为线索和中心来组织材料，程序案例也是围绕 C 语言语法知识点而展开的。这种编写方法具有知识点覆盖全面、逻辑清楚的特点，但是在组织训练学生程序开发的实践能力和工程能力，灵活运用 C 语言解决实际问题方面具有不足。另一些教材的编写方法是以程序案例为线索组织材料，强调学生程序开发的实践能力和工程能力的训练，但往往遗漏一些 C 语言的语法知识点，还易于陷入内容凌乱的境地。

本书的编写融上述两种方法于一体，取长补短，既充分考虑 C 语言重要语法知识点的全面性，又突出学生程序开发的实践能力和工程能力的训练。内容上以一明一暗两条线索来组织材料。明线是 C 语言语法知识点，从简单数据结构、简单控制结构到复杂数据结构、复杂控制结构，循序渐进地展示 C 语言特性。暗线则是两个涵盖了常见的如排序、查找、删除等程序算法的实际应用（科学计算器和学生成绩管理系统）贯穿全书始终。针对这两个应用，采用螺旋式的讨论方法。也就是说，先进行简要介绍，然后在后续章节中再进行一次或多次介绍，每次逐渐增加一些细节内容。本书的进度安排是经过深思熟虑的。每章都按照循序渐进的方式进行组织，并且前后内容由浅入深，相互呼应。对于大多数学生来说，这种循序渐进的方法是最合适的，既能避免产生厌倦，又能防止“信息超载”。

另外，本书每章节的末尾都有一个思考题部分，收集了与本章节内容相关的问题。思考题是对一些难以理解的问题的进一步讨论。虽然具有多种编程语言经验的读者会满足于简明扼要的说明和少量的示例，但是缺乏经验的读者需要更多的内容以帮助理解。

本书由李长云提出编写思路和编写大纲，廖立君、王平、童启和王志兵参加编写，最后由李长云统稿。研究生霍阔、居庆玮、赵正伟参加了文字编辑和图形绘制工作。本书是湖南省普通高等学校省级精品课程“C 语言程序设计”、湖南省普通高等学校特色专业计算机科学与技术的建设与研究成果，本书电子教案、扩展练习及其它参考资料请参见网站 http://58.20.192.206/ec/C16/Course/Index.htm，或联系电子邮箱 lcy469@163.com。

目　录

第1章　C语言程序设计概述

1.1　程序设计语言

1.1.1　自然语言与计算机语言

自然语言是人类在自身发展的过程中形成的语言，是一种自然地随文化演化的语言。例如，通常英语、汉语、日语为自然语言。人类有别于一般动物，一个重要特征就在于人类创造并使用语言及作为其载体的文字。

动物和人类都有生命周期、个体人和个体动物的生命周期相对物种的延续历史都非常短。动物和人类在实践中获得的知识都无法遗传给后代，而言传身教的知识受个体生物生命周期、活动范围、传播方式的局限，所以人类创造了语言及文字，是与动物最重要的区别！因为语言及文字不仅可以让人类高效、精确、广泛地交流人群的思维成果，更能让人类的智慧得以积累，让后人能够在继承前人积累的智慧（知识）基础上，不断发展。

自然语言是人类交流和思维的主要工具，是人与人之间传递信息的媒介。但是，计算机目前不能识别、理解与执行人类的自然语言，要使计算机执行人们的意志，必须有一种既使人能够掌握和书写，又让计算机能够识别、执行的人工语言。人工语言指的是人们为了某种目的而自行设计的语言。用于人与计算机之间通信的人工语言就是计算机语言（Computer Language）。

与自然语言相比，计算机语言具有如下特点：

（1）严格定义，有严格的语法；

（2）语义上无二义性；

（3）比自然语言要精简；

（4）是人能掌握和书写，计算机可识别和理解的。

总之，计算机语言是人与计算机之间传递信息的媒介，是人用来表达需求并控制计算机执行这种需求的专门语言。根据分工的不同，计算机语言大致可以分为以下几种类型：

（1）形式化需求规格语言。用于严格地、无二义地表达计算机用户需求的计算机语言，如 Z 语言、VDM 语言。

（2）软件设计语言。用于表达软件设计策略、设计结构和算法的计算机语言，如统一建模语言 UML、软件体系结构设计语言 ACME。

（3）程序设计语言。通常简称为编程语言，让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下计算机所应当采取的行动。典型地如 C 语言、JAVA 语言等。程序设计语言是最重要的计算机语言。

（4）其它计算机语言。如用于表达 Internet 网页内容的 HTML、用于计算机数据交换的 XML、用于数据库操作的 SQL 语言等。

计算机语言的发展也是一个不断演化的过程，其根本的推动力就是抽象机制更高的要求，以及对程序设计思想的更好的支持。具体而言，就是把机器能够理解的语言提升到也能够很好地模

仿人类思考问题的形式。例如程序设计语言的演化从最开始的机器语言到汇编语言到各种结构化高级语言，最后到支持面向对象技术的面向对象语言。

1.1.2 程序设计语言介绍

程序设计语言（Programming Language）是最为重要的计算机语言，是用于书写计算机程序的语言，通常简称为编程语言。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计语言中，这些记号串就是程序。程序设计语言有 3 个方面的因素，即语法、语义和语用。语法表示程序的结构或形式，亦即表示构成语言的各个记号之间的组合规律，但不涉及这些记号的特定含义，也不涉及使用者。语义表示程序的含义，亦即表示按照各种方法所表示的各个记号的特定含义，但不涉及使用者。语用表示程序与使用者的关系。

一般而言，程序设计语言的基本成分不外乎 4 种。①数据成分，用以描述程序中所涉及的数据；②运算成分，用以描述程序中所包含的运算；③控制成分，用以表达程序中的控制构造；④传输成分，用以表达程序中数据的传输。

经过 50 多年的发展，目前已开发出上千种程序设计语言，著名的有 FORTRAN、COBOL、ALGOL、C、PASCAL、JAVA 等。按语言级别，程序设计语有低级语言和高级语言之分。低级语言包括字位码、机器语言和汇编语言。它的特点是与特定的机器有关，功效高，但使用复杂、繁琐、费时、易出差错。其中，字位码是计算机唯一可直接理解的语言，但由于它是一连串的字位，复杂、繁琐、冗长，几乎无人直接使用。机器语言是表示成数码形式的机器基本指令集，或者是操作码经过符号化的基本指令集。汇编语言是机器语言中地址部分符号化的结果，或进一步包括宏构造。

高级语言的表示方法要比低级语言更接近于待解问题的表示方法，其特点是在一定程度上与具体机器无关，易学、易用、易维护。当高级语言程序翻译成相应的低级语言程序时，一般说来，一个高级语言程序单位要对应多条机器指令，相应的编译程序所产生的目标程序往往功效较低。

按照用户要求，有过程式语言和非过程式语言之分。过程式语言的主要特征是，用户可以指明一列可顺序执行的运算，以表示相应的计算过程。例如，FORTRAN、COBOL、ALGOL 60、C 等都是过程式语言。非过程式语言的含义是相对的，凡是用户无法指明表示计算过程的一列可顺序执行的运算的语言，都是非过程式语言。著名的例子是表格的生成程序（RPG），使用者只须指明输入和预期的输出，无须指明为了得到输出所需的过程。

按照应用范围，有通用语言和专用语言之分。目标非单一的语言称为通用语言，例如 FORTRAN、C 等都是通用语言。目标单一的语言称为专用语言，如 APT 等。

按照使用方式，有交互式语言和非交互式语言之分。具有反映人—机交互作用的语言成分的称为交互式语言，如 BASIC 语言就是交互式语言。语言成分不反映人—机交互作用的称非交互式语言，如 FORTRAN、COBOL、C、PASCAL 等都是非交互式语言。

按照成分性质，有顺序语言、并发语言和分布语言之分。只含顺序成分的语言称为顺序语言,如 FORTRAN、C 等都属顺序语言。含有并发成分的语言称为并发语言，如并发 PASCAL、MODULA 和 ADA 等都属并发语言。考虑到分布计算要求的语言称为分布语言，如 MODULA 便属分布语言。

传统的程序设计语言大都以诺伊曼式的计算机为设计背景，因而又称为诺伊曼式语言，FORTRAN、COBOL、C、PASCAL、JAVA 等都属于诺伊曼式语言。J.巴克斯于 1977 年提出的函数式语言，则以非诺伊曼式的计算机为设计背景，因而又称为非诺伊曼式语言，著名的如 LISP、PROLOG 语言。

1.2 程序和算法

1.2.1 程序及程序设计

计算机程序或者软件程序（通常简称程序）是指使用某种程序设计语言编写的一组指示计算机每一步动作的指令。在《计算机软件保护条例》中的定义为：为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列，或者可被自动转换成代码化指令序列的符号化指令序列或者符号化语句序列。

程序设计是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。由于程序是软件的本体，软件的质量主要通过程序的质量来体现，在软件研究中，程序设计的工作非常重要，内容涉及到有关的基本概念、工具、方法以及方法学等。程序设计往往以某种程序设计语言为工具，给出这种语言下的程序。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。专业的程序设计人员常被称为程序员。

按照结构性质，有结构化程序设计与非结构化程序设计之分。前者是指具有结构性的程序设计方法与过程。它具有由基本结构构成复杂结构的层次性，后者反之。按照用户的要求，有过程式程序设计与非过程式程序设计之分。前者是指使用过程式程序设计语言的程序设计，后者指非过程式程序设计语言的程序设计。按照程序设计的成分性质，有顺序程序设计、并发程序设计、并行程序设计、分布式程序设计之分。按照程序设计风格，有逻辑式程序设计、函数式程序设计、对象式程序设计之分。

程序设计的基本概念有程序、数据、子程序、函数、模块以及顺序性、并发性、并行性和分布性等。程序是程序设计中最为基本的概念，子程序和函数都是为了便于进行程序设计而建立的程序设计基本单位，顺序性、并发性、并行性和分布性反映程序的内在特性。

1.2.2 算法概念及其特性

著名的计算机科学家尼克劳斯·沃思认为：程序=数据结构+算法。算法（Algorithm）是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。一个算法应该具有以下 5 个重要的特征。

（1）有穷性。算法的有穷性是指算法必须能在执行有限个步骤之后终止。

（2）确切性。算法的每一步骤必须有确切的定义。

（3）输入。一个算法有 0 个或多个输入，以刻画运算对象的初始情况，所谓 0 个输入是指算法本身定出了初始条件。

（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。没有输出的算法是毫无意义的。

（5）可行性。算法中执行的任何计算步都是可以被分解为基本的可执行的操作步，即每个计算步都可以在有限时间内完成。

不同的算法可能用不同的时间、空间或效率来完成同样的任务。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。

（1）时间复杂度。算法的时间复杂度是指执行算法所需要的时间。一般来说，计算机算法是问题规模 n 的函数 $f(n)$，算法的时间复杂度也因此记做：

$$T(n)=O(f(n))$$

因此，问题的规模 n 越大，算法执行时间的增长率与 $f(n)$的增长率正相关，称作渐近时间复杂度（Asymptotic Time Complexity）。

（2）空间复杂度。算法的空间复杂度是指算法需要消耗的内存空间。其计算和表示方法与时间复杂度类似，一般都用复杂度的渐近性来表示。同时间复杂度相比，空间复杂度的分析要简单得多。

1.2.3 算法的描述

算法的描述方法可以归纳为以下几种：

（1）自然语言，易写易读，但存在表达冗长和语义多义性的缺陷；

（2）图形，如 N-S 图、程序流程图，具有简洁、直观、准确等特性；

（3）算法语言，即计算机语言、程序设计语言、伪代码；

（4）形式语言，用数学的方法，可以避免自然语言的二义性。

用各种算法描述方法所描述的同一算法，该算法的功用是一样的，允许在算法的描述和实现方法上有所不同。

本书使用程序流程图来描述算法。程序流程图又称程序框图，它定义了一些基本的图框，并用带箭头的直线（称流程线）把各种图框连接起来，箭头表示处理的流向。常用的图框如表 1-1 所列。

表 1-1 常用框图

形 状	名 称	含 义
	起止框	表示一个算法的开始与结束
	数据框	框中指出输入或输出的数据内容
	处理框	框中指出所进行的处理
	判断框	框中指出判断条件，框外可连接两条流程线，分别指明条件为真（True）时或条件为假（False）时的处理流向

程序流程图如图 1-1 所示。

[例] 鸡兔共笼，一共有 30 个头，90 只脚，求鸡兔各有多少？

解：一只鸡有一个头、两只脚，一只兔有一个头、四只脚。设鸡兔的总头数为 H，总脚数为 F，又设鸡的只数为 X，兔的只数为 Y，可列出方程组为：

X+Y=H 整理后得： Y=（F-2H）/2

2X+4Y=F X=（4H-F）/2

据此可得如图 1-1（a）所示算法。

像这种从上至下依次执行一系列操作的算法和程序称为顺序结构。

[例] 输入两个数 A、B，求其中的大者并输出。

解：可写出算法如图 1-1（b）所示。

[例] 用连加法求自然数 1~100 之和。

解：设 N 表示 1~100 间的一个自然数，用 S 保存累加之和，可写出算法如图 1-1（c）所示。

像这种可以多次重复执行某一部分处理的算法和程序称为循环结构。

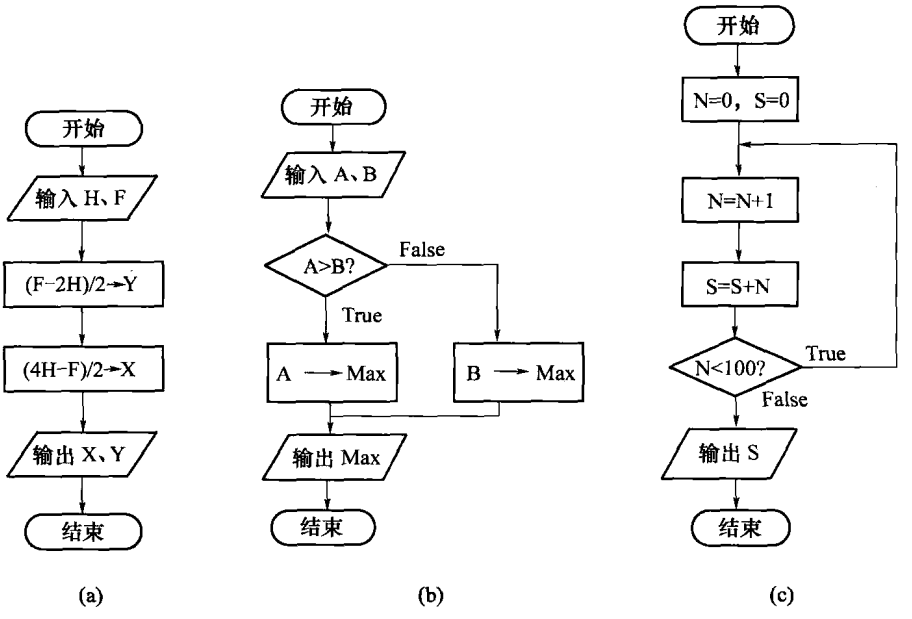

图 1-1　程序流程图示例

顺序结构、分支结构、循环结构是结构化程序设计的 3 种基本结构。

1.3　C 语言的发展及特点

1.3.1　C 语言的发展

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(bit)操作等。早期的操作系统等系统软件主要是用汇编语言编写的，如 UNIX 操作系统。但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。为了提高可读性和可移植性，最好改用高级语言，而一般高级语言又难以实现汇编语言的某些功能，人们设想能否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，集它们的优点于一身。于是，C 语言就在这种情况下应运而生了，之后成为国际上广泛流行的计算机高级语言。它适合作为系统描述语言，既可用来写系统软件，也可用来写应用软件。

C 语言是在 B 语言的基础上发展起来的，它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序，1963 年英国剑桥大学推出了 CPL(Combined Programming Language)语言。CPL 语言在 ALGOL 60 的基础上接近硬件一些，但规模比较大，难以实现。1967 年英国剑桥大学的 Matin Richards 对 CPL 语言作了简化，推出了 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，又作了进一步简化，它使得 BCPL 能挤压在 8KB 内存中运行，这个很简单的而且很接近硬件的语言就是 B 语言(取 BCPL 的第一个字母)，并用它写了第一个 UNIX 操作系统，在 DEC PDP-7 上实现。1971 年在 PDP-11/20 上实现了 B 语言，并写了 UNIX 操作系统。但 B 语言过于简单，功能有限，并且和 BCPL 都是“无类型”的语言。

1972 年至 1973 年间，贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母)。C 语言既保持了 BCPL 和 B 语言的优点(精练，接近硬件)，又克服了它们的缺点(过

于简单，数据无类型等)。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工具语言而设计的。1973 年，K.Thompson 和 D.M.Ritchie 两人合作把 UNIX 的 90%以上用 C 改写，即 UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国贝尔实验室的 K. Thompson 和 D. M. Ritchie 开发成功的，是用汇编语言写的，这样，UNIX 使分散的计算系统之间的大规模联网以及互联网成为可能。

后来，C 语言作了多次改进，但主要还是在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们普遍注意。1977 年出现了不依赖于具体机器的可移植 C 语言编译程序，使 C 移植到其它机器时所需做的工作大大简化了，这也推动了 UNIX 操作系统迅速地在各种机器上实现。随着 UNIX 的日益广泛使用，C 语言也迅速得到推广。1978 年以后，C 语言已先后移植到大、中、小、微型机上，如 IBM System/370、Honeywell 6000 和 Interdata 8/32，已独立于 UNIX 和 PDP 了。现在 C 语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

1983 年，美国国家标准化协会(ANSl)X3J11 委员会根据 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。1987 年，ANSI 又公布了新标准——87 ANSI C。目前流行的 C 编译系统都是以它为基础的。广泛流行的各种版本 C 语言编译系统虽然基本部分是相同的，但也有一些不同。在微型机上使用的有 Visual C、Borland Turbo C、Quick C 和 AT&T C 等，它们的不同版本又略有差异。JAVA、C++、C#都是以 C 语言为基础发展起来的。本书围绕 87 ANSI C 标准展开讲解。

1.3.2 C 语言的特点

C 语言具有以下特点。

（1）简洁紧凑、灵活方便。C 语言一共只有 32 个关键字，9 种控制语句，程序书写自由，主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。 C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

（2）运算符丰富。C 语言的运算符包含的范围很广泛，共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。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）数据结构丰富。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，能用来实现各种复杂的数据类型的运算，并引入了指针概念，使程序效率更高。

（4）C 语言是结构化语言。C 语言是以函数形式提供给用户的，这些函数可方便地调用，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

（5）C 语法限制不太严格、程序设计自由度大。一般的高级语言语法检查比较严，能够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。

（6）C 语言允许直接访问物理地址，可以直接对硬件进行操作。因此 C 既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位、字节和地址进行操作。

（7）C 语言程序生成代码质量高，程序执行效率高。一般只比汇编程序生成的目标代码效率低 10%~20%。

（8）C 语言适用范围广，可移植性好。C 语言的一个突出优点就是适合于多种操作系统，如 DOS、UNIX，也适用于多种机型。

当然，C 语言也有自身的不足，比如：C 语言的语法限制不太严格，对变量的类型约束不严

格，影响程序的安全性，对数组下标越界不作检查等。从应用的角度，C 语言比其它高级语言较难掌握。

总之，C 语言既有高级语言的特点，又具有汇编语言的特点；既能用来编写不依赖计算机硬件的应用程序，又能用来编写各种系统程序；是一种受欢迎、应用广泛的程序设计语言。

1.3.3 C 语言的基本结构

为了说明C语言源程序结构的特点，先看两个 C 语言程序。

例 1-1

```
main()
{
   printf("世界，您好！\n");
}
```

（1）main 是主函数的函数名，表示这是一个主函数。

（2）每一个 C 源程序都必须有，且只能有一个主函数(main 函数)。

（3）函数调用语句，printf 函数的功能是把要输出的内容送到显示器去显示。

（4）printf 函数是一个由系统定义的标准函数，可在程序中直接调用。

例 1-2

```
#include<stdio.h>
int max(int a,int b);              /*函数说明*/
main()                             /*主函数*/
{
  int x,y,z;                       /*变量说明*/
  printf("input two numbers:\n");
  scanf("%d%d",&x,&y);             /*输入x,y值*/
  z=max(x,y);                      /*调用max 函数*/
  printf("maxmum=%d",z);           /*输出*/
}
int max(int a,int b)               /*定义 max 函数*/
{
  if(a>b)return a;
  else return b;                   /*把结果返回主调函数*/
}
```

上例中程序的功能是由用户输入两个整数，程序执行后输出其中较大的数。本程序由两个函数组成：主函数和 max 函数。函数之间是并列关系，可从主函数中调用其它函数。max 函数的功能是比较两个数，然后把较大的数返回给主函数。max 函数是一个用户自定义函数。因此在主函数中要给出说明。在程序的说明部分中，不仅可以有变量说明，还可以有函数说明。关于函数的详细内容将在以后介绍。在程序的每行后用/*和*/括起来的内容为注释部分，程序不执行注释部分。

上例中程序的执行过程是，首先在屏幕上显示提示串，请用户输入两个数，回车后由 scanf 函数语句接收这两个数送入变量 x、y 中，然后调用 max 函数，并把 x、y 的值传送给 max 函数的