全国高等院校规划教材

C语言程序设计

● 赵喜清　李思广　主编

中国农业科学技术出版社

全国高等院校规划教材

C语言程序设计

● 赵喜清　李思广　主编

中国农业科学技术出版社

图书在版编目（CIP）数据

C 语言程序设计/赵喜清，李思广主编．—北京：中国农业科学技术出版社，2008.8
ISBN 978-7-80233-637-7

Ⅰ.C… Ⅱ.①赵…②李… Ⅲ.C 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2008）第 087395 号

责任编辑 朱 绯
责任校对 贾晓红 康苗苗

出 版 者 中国农业科学技术出版社
北京市中关村南大街 12 号 邮编：100081
电 话 （010）82106632（编辑室）
传 真 （010）82106626
网 址 http://www.castp.cn
经 销 者 新华书店北京发行所
印 刷 者 北京科信印刷有限公司
开 本 787 mm×1 092 mm 1/16
印 张 17.75
字 数 415 千字
版 次 2008 年 8 月第 1 版 2013 年 1 月第 2 次印刷
定 价 35.00 元

《C语言程序设计》编委会

主　编　赵喜清　李思广

副主编　钱冬云　杨晓波
　　丁丽英　李忠哗

编　者　丁丽英（黑龙江生物科技职业学院）
　　王凤利（河北北方学院）
　　王永乐（许昌职业技术学院）
　　卢凤伟（黑龙江畜牧兽医职业学院）
　　张文英（河北北方学院）
　　张艳慧（河北北方学院）
　　李向前（周口职业技术学院）
　　李忠哗（河北北方学院）
　　李思广（周口职业技术学院）
　　杨晓波（西藏民族学院）
　　肖桂云（河北北方学院）
　　赵喜清（河北北方学院）
　　钱冬云（浙江工贸职业技术学院）
　　康振华（山东工商学院）

主　审　郭喜凤（河北北方学院）

前　言

C 语言程序设计是高等学校普遍开设的一门计算机基础课程，在大学生现代思维训练、创新能力培养、计算机素质教育等方面发挥着重要作用。无论是计算机专业还是非计算机专业学习 C 语言已经成为广大青年学生的迫切需求，也是国内各类计算机考试的必考内容。

本书的作者都是多年从事计算机科学与技术教育的优秀教师，对基础课教学规律、计算机语言课教学特点，有深刻的认识和系统的研究，在教材建设方面也进行了许多有益的探索。本书是作者在吸收和借鉴已有教材长处的基础上，融入多年的教学实践经验和教学研究成果而编著完成的。

突破传统计算机语言教材编写方法，以教师教学实践为根基，以普及学生书写程序为目标，第一次把书写程序与作文写作类比起来，讲究“词汇段落积累”学习法，从而彻底解放编程思想，为编程大众化走出关键性一步。

C 语言是用自然语言来书写程序的，用数学语言表达解题意图，用英语来描述计算机能够接受的指令。遵循 C 语言的内涵、规律，本书以“基本符号→数据→表达式→语句→程序”流程为线索，按照熟悉的自然语言语法规则学习 C 语言，从而做到“统观全局，突出主干，脉络清晰”的目的和效果。

在对 C 语言程序设计知识点进行系统分析论证的基础上，合理取舍每个教学单元的知识内容，将主干知识列入教学目标，放在首位；将分支知识作次要介绍；对不利于课程主体内容教学的“末叶”知识放在后面。实际上，许多问题要在潜移默化中解决。

以大量翔实的图表展示知识及其内在联系是本书实施教学的突出特点。收集整理了过去多年来教学和实践所创造的智慧结晶，把抽象的理论、复杂的运算和深层的本质与内涵通过图表展示出来，通俗易懂。书中所列例题大多是经典编程范例，内容不仅涉及了许多计算机典型语句，更容纳了大量编程思想和编程技术，如经典数学问题解决方案，计算机枚举、递归和模拟仿真等技术。为了更好地强化 C 语言知识，编者精心筛选了近年来全国计算机等级考试部分标准试题列入习题，供大家练习。

本书由赵喜清、李思广主编。钱冬云、丁丽英、杨晓波、李忠哗副主编。全书编写分工如下：第一章、第四章由河北北方学院赵喜清、王凤利编写，第二章、第三章由周口职业技术学院李思广、李向前编写，第五章由山东工商学院康振华、河北北方学院张文英编写，第六章由浙江工贸职业技术学院钱冬云编写，第七章由黑龙江畜牧兽医职业学院卢凤伟编写，第八章、第九章由黑龙江生物科技职业学院丁丽英编写，第十章、第十一章由西藏民族学院杨晓波、许昌职业技术学院王永乐编写，第十二章、第十三章由河北北方学院李忠哗、张艳慧、肖桂云编写。

本书由河北北方学院郭喜凤教授主审，全程指导，并付出了不少心血。编著的整个过程中，得到天津大学任长明教授、天津师范大学李凤来教授的支持和帮助，在此特表感谢。

目　录

第一章　C 程序设计概述

1.1　程序设计语言

计算机的诞生，是科学发展史上的 1 个重要里程碑，使人类部分脑力劳动进入自动化，扩展了人类的认识能力，丰富了人类的精神财富。在科学技术飞速发展的今天，计算机的广泛应用是这个时代的重要标志。

计算机硬件提供了对数据计算问题解决的可能性。要使计算机按照人们的意图完成一项任务，就必须向它发出命令。能使计算机动作的命令叫指令，若干条指令的有序排列叫程序，把解决一项任务的思路、方法和步骤最终落实为计算机程序的过程就是程序设计。用于书写计算机程序的语言叫程序设计语言，它是人与计算机之间进行信息交流的工具。

在现代计算机中，信息是以二进制的形式来表示、存储和处理的，即用二进制数码 0 或 1 来表示机器指令。这种由 0 或 1 来描述机器指令的计算机语言叫机器语言，可以直接为计算机所接受，不必经过翻译，执行的速度快，效率高。但是，采用机器语言编制程序，要求程序员熟练地记忆所有机器指令的二进制代码、数据单元地址和指令地址，工作量大，容易出错。此外，由于写出来的程序不直观，可读性很差，也给程序的检查和分析带来很大的困难。

人类日常用来交流思想的语言称为自然语言，如汉语、英语、法语、俄语等，计算机一般不能直接理解这些语言。人们探求用更接近自然语言的语言来书写程序，并能为计算机接受，这种语言被称为高级语言。高级语言用一些符号来描述解题意图，很接近于数学公式的自然描述，不必了解实际计算机的机型、内部结构及其 CPU 的指令系统，只要掌握某种高级语言本身所规定的语法和语义，便可直接用该语言来编程，大幅度降低了编程的劳动强度，提高了编程效率。当然，计算机也不能直接识别和执行用高级语言编写的程序，必须将高级语言翻译成机器语言后，才能被计算机接受并运行。这个翻译过程是由计算机系统软件中的翻译程序完成的。翻译方式有解释和编译两种形式。编译方式是先编译，后执行；解释方式是边解释，边执行。

计算机解题的过程可归结为：

①程序员用高级语言编写程序；

②将程序与数据输入计算机，并由计算机将程序翻译成机器语言程序，保存在计算机的存储器中；

③运行程序，输出结果。

以上解题的过程建立在“存储程序和程序控制原理”之上，此原理是计算机自动连续工作的基础，在 1964 年由冯·诺依曼所领导的研究小组正式提出并论证，其基本思想如下：

①采用二进制形式表示数据和指令。二进制信号在物理上最容易实现，例如用高、低

两个电位表示“1”和“0”，或用脉冲的有无、脉冲的正负极性来表示。二进制数码的编码、计数、加减运算规则简单，可用开关电路实现。

②将程序（包括数据和指令序列）事先存入主存储器中，使计算机能够自动高速地从存储器中取出指令并执行。

1.2 程序设计的基本步骤

程序是计算机能够识别、执行的一组指令。人们正是通过编写程序来让计算机帮助我们解决各种各样的问题。这个过程一般可以分为以下 4 个步骤。

1.2.1 需求分析

设计程序要有的放矢，首先确定程序要解决的实际问题，然后要认真分析研究，弄清楚我们的核心任务是什么，输入是什么，输出是什么等。假设我们要编写 1 个程序，实现从华氏温度到摄氏温度的转换。显然，对于这个问题来说，输入是 1 个华氏温度，输出是相应的摄氏温度，而我们的核心任务就是如何来实现这种转换。

1.2.2 算法（algorithm）设计

对于给定的问题，采用分而治之的策略，把它进一步分解为若干个子问题，然后对每个子问题逐一进行求解，将问题分析的思路进一步明确化、详细化，建立解决问题的数学模型或者物理模型。把解题的步骤和方法一步一步地详细写下来，最后得到的结果通常是流程图或伪代码的形式，也就是提出算法，以便为下一步用计算机语言表达这些算法奠定基础。

例如，对于上述的温度转换问题，可以设计出如下的算法：

（1）输入 1 个华氏温度 F；

（2）利用公式：$C=5\times(F-32)\div9$，计算出相应的摄氏温度 C；

（3）显示计算出来的结果。

1.2.3 编码实现

在计算机上，使用某种程序设计语言，把算法转换成相应的程序，然后交给计算机去执行。如前所述，我们只能使用计算机能够看懂的语言来跟它交流，而不能用人类的自然语言来命令它。

1.2.4 测试与调试

编写完程序后，要将程序输入到计算机中进行调试，调试的目的是发现程序编写中的语法错误和程序设计算法上的逻辑错误，并将这些错误排除。一般在测试时，要选择一些有代表性的典型数据或者模拟某些特定的环境，对程序进行测试。1 个好的程序，还要求具备良好的人机交互界面，并且具备一定的容错功能，能发现输入数据中的错误，并给予提示，所有的这些都要通过程序调试加以解决。

1.2.5　运行与维护

程序经过调试，纠正其中的错误后，就可以正式投入运行，实现程序设计的功能。由于实际的应用问题和客户的需求不是一成不变的，程序正式投入运行后，可能因为应用环境的变化（包括所使用硬件的更新）和客户需求的变化，要不断地进行维护、修改其中存在的漏洞，开发新增功能。当对某 1 个软件进行较大程度的修改，新增了较多功能，就称为软件的版本升级。一般而言，每 1 个软件都有 1 个版本号，版本号的变化反映了该软件产品的升级情况。

1.3　算法及其表示

1.3.1　算法

算法就是对问题求解方法和步骤的精确描述。在进行程序设计时，最关键的问题是算法的提出。因为它直接关系到程序的正确性、可靠性。如果没有认真地研究实际问题，就草率地提出一些不成熟的算法，那么编写出来的程序就可能出现错误。

算法作为对解题步骤的精确描述，应具备以下特点：

（1）有穷性　1 个算法必须在有限步骤之后结束，而不能无限制地进行下去。因此在算法中必须给出 1 个结束的条件。

例如：不能指望计算机算出圆周率的精确值，因为它是 1 个无穷不循环小数，无法求出它的精确值。

（2）明确性　1 个算法中的任何步骤都必须意义明确，不能模棱两可、含混不清，即不允许有二义性，不能在计算机中使用诸如“老张对老李说：他的儿子考上了清华大学”这种歧义的表达方法。到底是老张的儿子上了大学还是老李的儿子上了大学？无法确定。

（3）可执行性　所采用的算法必须能够在计算机上执行，在算法中所有的运算必须是计算机能够执行的基本运算。要计算机执行的步骤，计算机应该能够实现，不能提出像“让计算机去煮饭，煮完饭之后再炒菜”之类的算法，至少它在目前无法实现。

（4）有一定的输入与输出　要计算机解决问题时，总是需要输入一些原始的数据，计算机向用户报告结果时，总是要输出一些信息。

1.3.2　算法的表示

算法可以用任何形式的语言和符号来描述，通常有自然语言、程序语言、流程图、N-S 图、PAD 图、伪代码等。无论采用那种工具来描述算法，都体现了程序控制流程。

早在 1966 年，Bohm 和 Jacopin 就证明了程序设计语言中只要有 3 种形式的控制结构就可以表示任何复杂程序结构，这 3 种基本控制结构是顺序、选择和循环。顺序结构表示程序中的各操作是按照它们出现的先后顺序执行的；选择结构表示程序的处理步骤出现了分支，它需要根据某一特定的条件选择其中 1 个分支执行；循环结构表示程序反复执行某

个或某些操作，直到某条件为假（或为真）时才可终止循环。

（1）文字描述　例如，我国数学家秦九韶在《数书九章》一书中曾记载了求两个正整数 m 和 n（m≥n）最大公约数的方法，即采用辗转相除法（也称欧几里德算法）。

计算机中这种思想如何实现？其计算方法用文字描述如下：

①将两个正整数存放到变量 m 和 n 中；

②求余数：用 m 除以 n，将所得余数存放到变量 r 中；

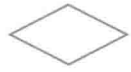
③判断余数 r 是否为 0：若 r 为 0 则执行第⑤步，否则执行第④步；

④更新被除数和除数：将 n 的值存放到 m 中，将 r 的值存放到 n 中，并转向第②步继续循环；

⑤输出 n 的当前值，算法结束。

（2）流程图描述　流程图又称为框图，它用规定的一系列图形、流程线及文字说明来表示算法中的基本操作和控制流程，形象直观，简单易懂，便于修改和交流。表 1.1 中是各种流程图符号。

表 1.1　标准流程图符号

符号名称	符号	功能
起止框		表示算法开始和结束
输入/输出框		表示算法的输入/输出操作，框内填写需输入/输出的各项
处理框		表示算法中的各种处理操作，框内填写各种说明
判断框		表示算法中的条件判断操作，框内填写判断条件
注释框		表示算法中某操作的说明信息
流程线		表示算法的执行方向
连接点		表示流程图的延续

最大公约数算法流程图如图 1.1 所示。

（3）N-S 图描述　N-S 图是另一种算法表示法，是由美国人 I. Nassi 和 B. Shnei derman 共同提出的。在 N-S 流程图中，完全去掉了带有方向的流程线，程序的 3 种基本结构分别用 3 种矩形框表示，将这 3 种矩形框进行组装就可表示全部算法。在 N-S 图中，1 个算法就是 1 个大的矩形框，框内包含若干个基本框，3 种基本结构的 N-S 图描述如下。

①顺序结构 N-S 图如图 1.2 所示，执行顺序为先 A 后 B；

②选择结构 N-S 图如图 1.3 所示。条件成立时执行 A，条件不成立时执行 B；

③ while 型循环的 N-S 图如图 1.4 所示，条件为真时一直循环执行循环体 A，直到条件为假时才跳出循环；

④ do-while 型循环的 N-S 图如图 1.5 所示，一直循环执行循环体 A，直至条件为假时

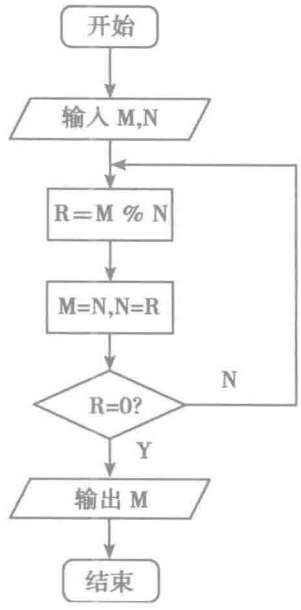


图 1.1　最大公约数算法流程图

才跳出循环。

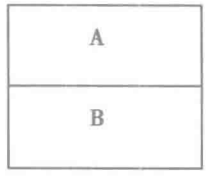


图 1.2　顺序结构 N-S 图

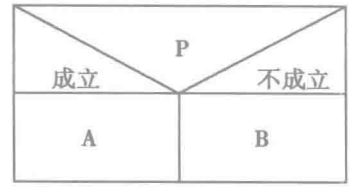


图 1.3　选择结构 N-S 图

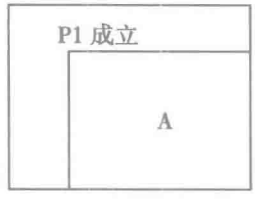


图 1.4　while 型循环 N-S 图

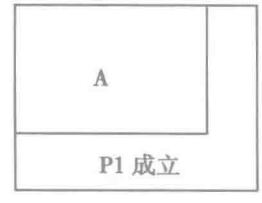


图 1.5　do-while 循环 N-S 图

1.4　C 语言的发展

C 语言是 UNIX 操作系统的主流语言，它与该系统有着互相依存、休戚与共的紧密关系。近年来，随着 UNIX 操作系统在国际上的广泛流行，C 程序设计语言在软件工程领域也越来越引人注目，无论是设计系统软件，还是开发图形处理、数据分析、数值计算等应用软件，都要用到 C 语言。

早先的操作系统（包括 UNIX 在内）主要是用汇编语言编写的。由于汇编语言还没有完全摆脱机器语言的束缚，仍然过分依赖于计算机硬件，因此不仅编程工作量大，难于实现复杂的科学计算和数据处理，而且程序的可读性和可移植性都比较差。为了提高程序的可读性和可移植性，从 1954 年开始陆续出现了接近人们熟悉的自然语言的高级语言。但一般的高级语言难以兼顾汇编语言的一些优点（如直接访问物理地址、操作 I/O 端口、进行位操作、目标代码执行效率高等），人们期待着一种既具有一般高级语言特性，又具有低级语言优点的新型语言出现，于是 C 语言就应运而生了。

C 语言的产生可以追溯到 1960 年出现的 ALGOL60，因为它离计算机硬件比较远，所以不宜用于编写系统程序。1963 年，英国剑桥大学在 ALGOL60 的基础上推出了 CPL，它比较接近硬件，但规模太大难以实现，因此将其简化，于 1967 年推出了 BCPL。1970 年，美国贝尔实验室将 BCPL 进一步简化为更接近硬件的 B 语言（取 BCPL 的第一个字母），并将其用于书写 UNIX 操作系统。后来又感到 B 语言功能有限，便在 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（语言精炼、接近硬件），又克服了二者的缺点（过于简单、数据无类型等）。1973 年，UNIX 操作系统的 90% 以上的代码都用 C 语言改写。1975 年公布了 UNIX6，C 语言的突出优点引起世人普遍关注。随着 UNIX 操作系统的广泛应用，C 语言也得到了迅速推广。1978 年以后，C 语言已经先后移植到大、中、小、微型机上，且在现在的单片机上也可以使用 C 语言开发程序。

C 语言反映了当前计算机的能力，它在各种计算机上的快速推广产生了许多版本，这些版本虽然相似，但通常情况下并不完全兼容，人们希望开发出的代码能够在各种平台上运行，为了解决这个问题，1983 年美国国家标准学会（ANSI）制定了 C 语言的标准草案（83ANSIC）。1987 年又推出了 87ANSIC。此后，又在 1989 年公布了 C89 标准。20 世纪 90 年代期间，尽管有许多人热衷于 C + +，但 C 语言并没有停滞不前，新的标准仍在不断推出，最终形成了 1999 年的 C99 标准。C99 保留了 C89 的全部特性，并增加了一些数据库函数和开发一些创新软件的新功能，这些改进再次把 C 语言推到了计算机语言应用的前沿。

1.5 C 语言的特点

C 语言是一种通用的编程语言，从诞生到现在的 30 多年间，取得了长足的发展。可以毫不夸张地说，C 语言是迄今为止人类发明的最为成功的计算机语言之一，它对计算机科学和软件产业的发展起着广泛而深刻的影响。在 Windows 如此普及、一些面向对象的专业语言方兴未艾的今天，依然有大量专业程序员、计算机爱好者及工程技术人员在使用 C 语言。C 语言作为世界上应用最广泛的计算机语言之一，之所以具有如此魅力，自然有其不可替代的、吸引人的特点。C 语言的主要特点如下：

（1）简单　C 语言的语法简洁、紧凑，是 1 个规模比较小的语言。不管是关键字的命名、变量的定义、运算符的设置，还是程序的结构，处处体现出简洁、紧凑的风格，压缩了一切不必要的成分。例如，在 C 语言中，在描述整型数据类型的时候，使用的是单词的缩写 int，而不是完整的英语单词 integer；在定义 1 个数组的时候，使用的是数组符号

[ ]，而不是英语单词 array。另外，C 语言不支持对组合对象的直接操作。例如，在 C 语言当中，并没有 1 个运算符来比较两个字符串的大小，也没有 1 个运算符来计算 1 个数组的长度。总之，C 语言的基本思路就是能省则省，连专门的输入输出运算符都没有。由于 C 语言的这种简短、精练的特点，使得人们在学习的时候比较容易入门，只要知道很少的东西就可以开始编程。

（2）实用　C 语言的语法虽然简洁、紧凑，但表达能力强，使用灵活、方便。当初 Ritchie 先生在发明 C 语言的时候，也正是为了编写 UNIX 操作系统的实际需要。C 语言共有 32 个关键字（见附录 A）、9 种控制语句，主要用小写字母表示，减少了编译系统的开销；程序书写形式自由，既可以一行一句，也可以一行多句，甚至一句可分写在多行上。C 语言有 34 种运算符，运算丰富，涵盖范围广，可以实现在其他高级语言中难以实现的运算。数据类型丰富。C 语言的数据类型有 13 种，特别是它具有数据类型构造能力，可以在基本类型（如字符型、整型、实型等）的基础上构造各种构造类型（如数组、结构体、共用体等），用于实现各种复杂的数据结构（堆栈、链表、队列、树、图等）的运算。尤其是指针类型数据的功能强大，恰当地使用它不仅可以简化程序结构，源程序更短一些，而且可以节省存储空间，提高运算速度。C 语言还提供了一组标准的库函数，能够帮助程序员完成各种各样的功能，如文件的访问、格式化输入输出、内存分配和字符串操作等。C 语言的语法简明扼要，对程序员的限制很少，从理论上来说，程序员几乎可以做他们想做的任何事情。

（3）高效　如前所述，C 语言是一种高级程序设计语言，但是从某种意义上来说，C 语言又是一种“低级”语言，是一种接近于机器硬件的语言。因为从数据类型来看，C 语言处理的都是基本的数据类型，如字符、整数、实数和地址等，而计算机硬件一般都能直接处理这些数据对象；从运算符来看，C 语言的运算符大都来源于真实的机器指令。例如，对于算术运算符、关系运算符、赋值运算符、指针运算符、位（bit）运算符、自增运算符和自减运算符等，在一般的计算机硬件上，都有与它们相对应的机器指令；从硬件访问来看，C 语言可以直接去访问内存地址单元，也可以直接去访问 I/O（输入/输出）接口。总之，C 语言与硬件的关系非常密切，可以把它看成是汇编语言之上的一层抽象，凡是汇编语言能实现的功能，C 语言基本上都能实现。事实上，甚至可以在 C 语言程序当中直接嵌入汇编语言程序。基于这些原因，人们很容易用 C 语言来编写出在时间上和空间上效率都很高的程序。根据一项统计，C 语言程序生成的目标代码的效率只比汇编语言低 10%。

（4）可移植性好　可移植是指程序从 1 个环境中不加或稍加改动就可搬到另 1 个不同的环境中运行。虽然 C 语言与机器硬件的关系非常密切，但它并不依赖于某一种特定的硬件平台。事实上，小至 PC 机、大至超级计算机，几乎在所有的硬件平台和操作系统上都有 C 语言的编译器。这就使得用 C 语言编写的程序具有很好的可移植性，基本上不做修改就能在另 1 个不同型号的计算机上运行。

1.6　C 语言的应用领域

如前所述，C 语言是一种偏向于机器硬件的高级语言，因此，它比较适合于底层的系

统软件的开发。例如，在操作系统领域，UNIX 和 Linux 的绝大部分代码都是用 C 语言编写的，只有少量的与硬件直接打交道的代码是用汇编语言编写的，如系统引导程序、底层的设备驱动程序等。对于 Windows 操作系统，它的大部分代码也是用 C 语言来编写的，而一些图形用户界面方面的代码则是用 C + + 编写的。在系统软件领域，数据库管理系统、磁盘管理工具乃至一些病毒程序，都是用 C 语言编写的，尤其是在嵌入式系统领域，根据一项统计，81% 的嵌入式系统开发都要用到 C 语言，远远超过其他的任何一种编程语言。在商业应用领域，例子就更多了，如多媒体播放软件、游戏软件、专家系统软件和绘图软件等。C 语言在工业界获得了广泛的应用和好评，根据 2003 年的一项统计，在企业招聘的软件工程师职位当中，有 22.7% 的职位要求使用 C/C + + 编程，在所有的编程语言中排名第二。在教学领域，著名的数学软件工具 MATLAB 就是用 C 语言来编写的。在数字信号处理领域，也经常用 C 语言来编程。另外，像数据结构这样的基础课程，以前都是用 Pascal 语言来教学，现在也都改成了 C 语言。所以说，学好了 C 语言，不管是对今后的其他一些后继课程，还是对将来的职业发展，都会带来很大的帮助。

1.7 C 程序的结构

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由简到难，表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中了解到组成 1 个 C 源程序的基本部分和书写格式。

例 1.1

```
main ( )
{
printf ("2008 北京奥运会! \n");
}
```

main 是主函数的函数名，表示这是 1 个主函数。每 1 个 C 源程序都必须有且只能有 1 个主函数（main 函数）。花括号中的语句是函数调用语句，printf 函数的功能是把要输出的内容送到显示器中去显示。printf 函数是 1 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"
                    /* include 称为文件包含命令，扩展名为 . h 的文件也称为头文件或首部文件 */
#include "math. h"
main ( )
{
double x, s;                    /* 定义两个实数变量，以便被后面程序使用 */
printf ("input number: \n"); /* 显示提示信息 */
scanf ("%lf", &x);              /* 从键盘获得 1 个实数 x */
s = sin (x);                    /* 求 x 的正弦，并把它赋值给变量 s */
```

```
    printf ("sine of %lf is %lf\n", x, s); /* 显示程序运算结果 */
}                                          /* main 函数结束 */
```

程序的功能是从键盘输入 1 个数 x，求 x 的正弦值，然后输出结果。在 main ( ) 之前的两行称为预处理命令（详见第六章）。预处理命令还有其他几种，这里的#include 称为文件包含命令，其意义是把尖括号 < > 或双引号 “” 内指定的文件包含到本程序中来，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为 .h，也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型，凡是在程序中调用 1 个库函数时，都必须包含该函数原型所在的头文件。在本例中，使用了 3 个库函数：输入函数 scanf，正弦函数 sin，输出函数 printf。sin 函数是数学函数，其头文件为 math.h，因此，在程序的主函数前用#include 命令包含了 math.h 文件。scanf 和 printf 是标准输入输出函数，其头文件为 stdio.h，在主函数前也用#include 命令包含了 stdio.h 文件。

需要说明的是，C 语言规定对 scanf 和 printf 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第二行的包含命令#include。同样，在例 1.1 中使用了 printf 函数，也省略了包含命令。

在例 1.1 中的主函数体中又分为两部分，一部分为说明部分；另一部分是执行部分。说明是指变量的类型说明。例题中未使用任何变量，因此无说明部分。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。这一点是编译型高级程序设计语言的 1 个特点，与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分。例 1.2 中使用了两个变量 x 和 s，用来表示输入的自变量和 sin 函数值。由于 sin 函数要求这两个量必须是双精度浮点型，故用类型说明符 double 来说明这两个变量。说明部分后的 4 行为执行部分或称为执行语句部分，用以完成程序的功能。执行部分的第一行是输出语句，调用 printf 函数在显示器上输出提示字符串，请操作人员输入自变量 x 的值。第二行为输入语句，调用 scanf 函数，接受键盘上输入的数并存入变量 x 中。第三行是调用 sin 函数并把函数值送到变量 s 中。第四行是用 printf 函数输出变量 s 的值，即 x 的正弦值。程序结束。

运行例 1.2 程序时，首先在显示器屏幕上给出提示串 “input number:”，这是由执行部分的第一行完成的。用户在提示下从键盘上键入某 1 个数，如 5，按下回车键，接着在屏幕上给出计算结果。

在前两个例子中用到了输入和输出函数 scanf 和 printf，在第三章中我们要详细介绍。这里我们先简单介绍一下它们的格式，以便下面使用。

scanf 和 printf 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入输出值。因此，这两个函数在括号中的参数表都由 “格式控制串” 和 “参数列表” 组成。格式控制串是 1 个字符串，必须用双引号括起来，它表示了输入输出量的数据类型。各种类型的格式表示法可参阅第三章。在 printf 函数中还可以在格式控制串内出现非格式控制字符，它在显示屏幕上将原文打印。参数表中给出了输入或输出的量。当有多个量时，用逗号间隔。例如：printf (“sine of %lf is %lf\n”, x, s)；语句中%lf 为格式字符，表示按双精度浮点数处理。它在格式串中出现 2 次，对应了 x 和 s 两个变量。其余字符为非格式字符则照原样输出在屏幕上。