·普通高等教育“十二五”规划教材

C程序设计教程

孟宪伟 刘前 主编

C CHENGXU
SHEJI
JIAOCHENG

国防工业出版社
National Defense Industry Press

普通高等教育“十二五”规划教材

C 程序设计教程

孟宪伟　刘前　主编

国防工业出版社

·北京·

内容简介

本书是学习C语言的基础教材，内容循序渐进，通俗易懂，并辅以大量的实例和习题，使得初学者能够很快掌握C语言的基本内容，并应用它编写程序来解决实际问题，为以后学习其他的高级语言打下良好的基础。

本书的主要内容包括：C语言的基本概念、各种数据类型的使用技巧、C语言模块化程序设计的方法、文件的基本操作和使用规则，考虑到学习程序设计必须重视实践环节，本书除了大量的例题和习题之外，还编写了配套的《C程序设计教程习题解析与上机指导》，供教师和学生参考。

本书内容丰富，教师可以根据不同专业的教学需要，灵活分配学时，选取合适的教学内容。

本书既可以作为高等院校计算机专业本科低年级学生学习计算机语言的入门教材，也可以作为高等院校非计算机专业学生的计算机语言教材，还可以作为科技人员自学C语言的自学参考书。

图书在版编目(CIP)数据

C程序设计教程/孟宪伟，刘前主编. —北京：国防工业出版社，2013.1

普通高等教育“十二五”规划教材

ISBN 978-7-118-08340-8

Ⅰ.①C... Ⅱ.①孟...②刘... 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(2012)第244242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)

涿中印刷厂印刷

新华书店经售

*

开本787×1092 1/16 印张17 字数416千字

2013年1月第1版第1次印刷 印数1—4000册 定价35.00元

(本书如有印装错误，我社负责调换)

国防书店：(010)88540777 发行邮购：(010)88540776

发行传真：(010)88540755 发行业务：(010)88540717

前　言

随着计算机技术的发展与普及，计算机已经成为各行各业最基本的工具之一，而且正迅速地进入人类生活的各个领域。C 语言作为国际上曾经最广泛流行的通用程序设计语言，在计算机的研究和应用中已展现出强大的生命力。C 语言兼顾了诸多高级语言的特点，是一种典型的结构化程序设计语言，它处理能力强，使用灵活方便，应用面广，具有良好的可移植性，既适合于计算机专业人员编写系统软件，又适合于应用开发人员编写应用软件，所以长久以来，广泛流行，经久不衰。虽然现在有了很多应用比较广泛的高级语言，但对于 C 语言，在现在的很多领域仍然有其用武之地。例如，使用 C 语言学习面向过程的编程思想，把实际问题建立模型、拆分。使用 C 语言要面对很多底层的东西，如说指针、字符串，对使用更“高级”的语言很有帮助。

C 语言的教材有很多，但是大部分都是循规蹈矩的“教材”。本教材在结构上突出了以程序设计为中心，以语言知识为工具的思想，对 C 语言的语法规则进行了整理和提炼，深入浅出地介绍了它们在程序设计中的应用；在内容上注重知识的完整性，以适合初学者的需求；在写法上追求循序渐进，通俗易懂，旨在引导初学者入门。本书以实际应用为目的，侧重于 C 语言基本知识，着重讲解概念，深入浅出地讲解用计算机解决问题的方法。对于初学者来说，掌握程序设计的技术和方法，在最初往往是枯燥乏味的。作为一种尝试，我们在教学中曾经将各种程序设计的技术和方法融于趣味问题之中，通过对一些饶有趣味的问题的讨论和求解，使读者在轻松、愉快的气氛中理解和探索程序设计的奥妙，从而达到事半功倍的学习效果。基于这样指导思想，本书在加强基础训练、介绍基本算法的同时，也选用了具有趣味性的例题。其目的就是要加强本书的可读性，使读者在轻松自然的学习过程中，掌握程序设计的方法。

在编写过程中参考了大量同类教材并吸收了这些教材的优点，同时又保持了自己的特色。本书的主要特点是：文章的叙述通俗易懂，内容的编排由浅入深、循序渐进；通过精心设计的例题，着重介绍 C 语言程序设计的基本方法与基本技巧；通过精心选择的习题，训练程序设计的技能。全书体现“程序设计 = 算法 + 数据结构”的程序设计课程教学内涵，贯彻传授知识、培养能力、提高素质的教学理念。全书各章配有习题，并有与之配套的习题及上机实训指导内容。全部的例题和习题均在 Turbo C 环境下调试、运行，以方便读者上机学习。

本书前六章由辽宁科技学院的刘前完成，后六章由辽宁科技学院的孟宪伟完成。沈阳工程学院的姜柳参与了本书的部分编写工作并审阅了书稿。

本书的出版得到了国防工业出版社的大力支持，在此表示衷心感谢。由于时间仓促和水平有限，书中难免有错误和疏漏之处，敬请专家、读者不吝赐教。

编　者

目　录

第1章　C语言概述

【学习目标】

(1) 了解几种常见问题的算法。
(2) 熟悉C语言的开发环境。
(3) 掌握用流程图描述算法。
(4) 掌握C语言程序的组成。

本章通过介绍算法概念、程序概念、程序设计语言、C语言的发展与特点、简单的C语言程序，引出了C语言程序的组成，即C语言程序由函数组成，函数由函数首部和函数体两部分组成。最后介绍了C语言的开发环境。通过本章的学习，读者应重点掌握用流程图描述算法，通过上机实验掌握调试C程序的方法，同时了解C语言基本结构。

1.1　算法与程序设计

1.1.1　算法的概念

做任何事情都有一定的步骤。为解决一个问题而采取的方法和步骤，称为算法。计算机解决问题所依据的步骤称为计算机算法，简称算法。

计算机算法可分为两大类：

(1) 数值运算算法：求解数值。如求方程的根、求一个数的绝对值函数等。
(2) 非数值运算算法：事务管理领域。如人事管理、图书检索等。

本课程的目的是使读者学会编写C语言程序，进行编写程序的初步训练，因此，只介绍算法的初步知识。请看下面两个例子。

【例1-1】计算1+2+3+…+100，可采取以下两种算法中的一种。

算法一：可以设两个变量(变量是指其值可以改变的量)，一个变量代表和(s)，一个变量代表加数(i)，用循环算法表示如下：

第一步：0⇒s,1⇒i。
第二步：s+i⇒s。
第三步：i+1⇒i。
第四步：如果i≤100，转第二步；否则，转第五步。
第五步：输出结果s，结束。

算法二：只有两步：

第一步：100×101/2⇒s。
第二步：输出s，结束。

【例1-2】判断一个大于等于3的正整数是不是素数。

所谓素数是指除了 1 和该数本身之外，不能被其他任何整数整除的数，例如 13 是素数，因为它不能被 2，3，4，…，11，12 整除。

判断素数的方法很简单，例如判断 n(n≥3)是不是素数，只需将 n 作为被除数，将 2 到(n-1)各个整数轮流作除数，作除法运算，如果都不能被整除(余数不为 0)，则 n 是素数。算法表示如下：

第一步：输入 n 的值。

第二步：i 作除数，2⇒i。

第三步：n 除以 i，得余数 r。

第四步：如果 r=0，表示 n 能被 i 整除，则打印 n 不是素数，转第七步；否则执行第五步。

第五步：i+1⇒i。

第六步：如果 i≤n-1，返回第三步；否则打印 n 是素数，转第七步。

第七步：结束。

通过上面例题，可以看出算法有如下特性。

1．有穷性

有穷性是指一个算法的操作步骤必须是有限的和合理的，即在合理的范围之内结束算法。例如求整数累加和的算法，由于整数本身是个无限集合，如果不限定其范围，会导致求解步骤是无限的。又例如，计算机执行某个算法需要几千年，虽然是有限的，但却是不合理的。当然，究竟什么算“合理”，并没有严格标准，由人们的常识和需要而定。

2．确定性

算法中每个操作步骤都应当是明确的，而不应是含糊的、模棱两可的。在计算机算法中最忌讳的是歧义性，所谓“歧义性”是指可以被理解为两种或多种可能的含义。因为计算机至今还没有主动思维的能力，如果给定的条件不确定，计算机就无法执行。例如，“计算 3 月 1 日是一年中的第几天”，这个问题是不确定的，因为没有指明哪一年，不知道是不是闰年，闰年和平年 2 月份的天数不一样，所以无法执行。

3．有零个或多个输入

执行算法时需要从外界获得必要信息的操作称为输入。输入的数据个数根据算法确定。例如计算 1～100 累加和的算法不需要输入；计算 n！的算法需要输入 n 的值；计算 m 和 n 的最大公约数和最小公倍数则需要输入 m 和 n 两个数的值。

4．有一个或多个输出

执行算法得到的结果就是算法的输出，没有输出的算法是没有意义的。

最常见的输出形式是屏幕显示或打印机输出，但并非唯一的形式。执行算法的目的就是为了求解，“解”就是输出。

5．有效性

算法中的每一个步骤都应当有效地执行，并得到确定的结果。例如当 b=0 时，a/b 是不能有效执行的。又例如，在 C 语言中，“a%b”中的 a 和 b 都必须是整型数据，否则也不能有效执行。

算法有优劣之分，一般希望用简单的和运算步骤少的算法。因此，为了有效地进行解题，不仅要保证算法正确，还要考虑算法的质量，选择合适的算法。

1.1.2 算法的表示

知道了算法的概念，那么如何表示一个算法呢？

表示一个算法，可以有不同的方法，常用的方法有自然语言、传统流程图、结构化流程图、伪代码等，下面分别作以介绍。

1．用自然语言表示算法

自然语言就是人们日常使用的语言，可以是汉语、英语或其他语言。用自然语言表示通俗易懂，但文字冗长，容易出现歧义性。除了很简单的问题，一般不用自然语言表示算法。

2．用流程图表示算法

流程图表示算法，直观形象，易于理解。美国国家标准化协会(ANSI)规定了一些常用的流程图符号(图 1-1)，已为世界各国程序工作者普遍采用。

菱形框(即判断框)的作用是对一个给定的条件进行判断，根据给定的条件是否成立决定如何执行其后的操作。它有一个入口，两个出口。

连接点是用于将不同地方的流程线连接起来。用连接点可以避免流程线的交叉或过长，使流程图清晰。

【例 1-3】将例 1-1 的算法用流程图表示，如图 1-2 所示。

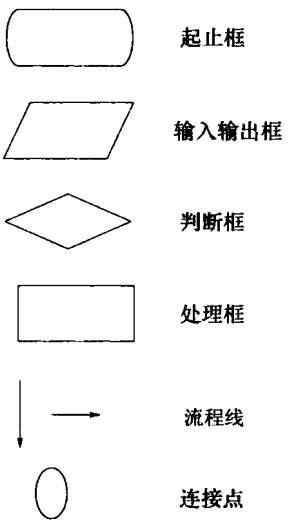

图 1-1

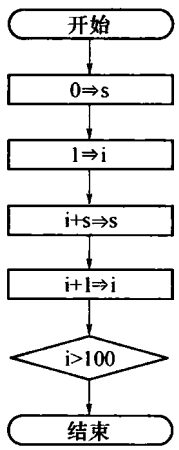

图 1-2

C 语言有三种基本程序结构：顺序结构、选择结构和循环结构。1966 年，Bdhra 和 Jacopini 提出了这三种基本结构，用这三种基本结构作为表示一个良好算法的基本单元。

下面介绍这三种基本结构的流程图。

1) 顺序结构

如图 1-3 所示，虚线框内是一个顺序结构。其中 A 和 B 两个框是顺序执行的。即在执行完 A 框所指定的操作后，必然接着执行 B 框所指定的操作。顺序结构是最简单的一种基本结构。

2) 选择结构。

选择结构又称选取结构或分支结构，如图 1-4 所示。虚线框内是一个选择结构。此结构中必包含一个判断框。根据给定的条件 P 是否成立而选择执行 A 框或 B 框。例如 P 条件可以是“x>=0”或“x>y”，“a+b<c+d”等，详见第 4 章。

注意：无论 P 条件是否成立，只能执行 A 框或 B 框之一，不可能既执行 A 框又执行 B

框。无论走哪一条路径，在执行完 A 或 B 之后，都经过 b 点，然后脱离本选择结构。A 或 B 两个框中可以有一个是空的，即不执行任何操作，如图 1-5 所示。

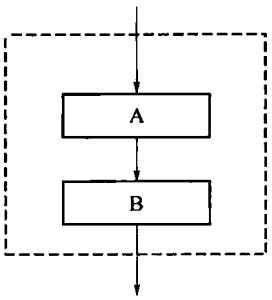

图 1-3

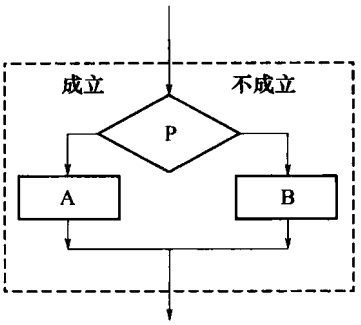

图 1-4

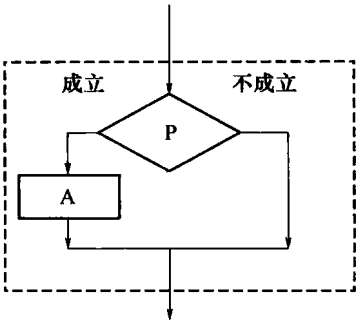

图 1-5

3) 循环结构

循环结构又称重复结构，即反复执行某一部分的操作。有两类循环结构：

(1) 当(while)型循环结构。当型循环结构如图 1-6(a)所示。它的功能是：当给定的条件 P1 成立时，执行 A 框操作，执行完 A 后，再判断条件 P1，是否成立，如果仍然成立，再执行 A 框，如此反复执行 A 框，直到某一次 P1 条件不成立为止，此时不执行 A 框，脱离循环结构。

(2) 直到(until)型循环结构。直到型循环结构如图 1-6(b)所示。它的功能是：先执行 A 框，然后判断给定的 P2，条件是否成立，如果 P2 条件不成立，则再执行 A，然后再对 P2，条件作判断，如果 P2 条件仍然不成立，又执行 A，……如此反复执行 A，直到给定的 P2 条件成立为止，此时不再执行 A，脱离本循环结构。

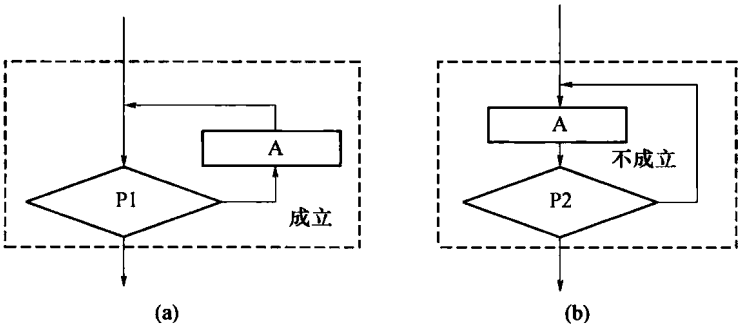

图 1-6

以上三种基本结构的共同特点：

① 只有一个入口。

② 只有一个出口。注意：一个判断框有两个出口，而一个选择结构只有一个出口。不要将判断框的出口和选择结构的出口混淆。

③ 结构内的每一部分都有机会被执行到。也就是说，对每一个框来说，都应当有一条从入口到出口的路径通过它。

④ 结构内不存在“死循环”(无终止的循环)。

3. 用 N-S 流程图表示算法

1973 年美国学者 I.Nassi 和 B.Shneiderman 提出了一种新的流程图形式。在这种流程图中，

完全去掉了带箭头的流程线，全部算法写在一个矩形框内，在该框内还可以包含其他的从属于它的框，或者说，由一些基本的框组成一个大的框。这种流程图又称 N-S 结构化流程图(N 和 S 是两位美国学者的英文姓氏的首字母)。这种流程图适于结构化程序设计，因而很受欢迎。

N-S 流程图用以下的流程图符号。

(1) 顺序结构。顺序结构用图 1-7 表示。A 和 B 两个框组成一个顺序结构。

(2) 选择结构。选择结构用图 1-8 表示。它与图 1-4 相应。当 P 条件成立时执行 A 操作，P 不成立则执行 B 操作。注意：图 1-8 是一个整体，代表一个基本结构。

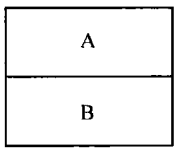

图 1-7

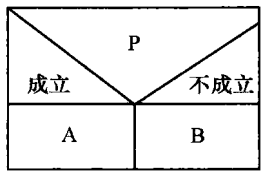

图 1-8

(3) 循环结构。

当型循环结构用图 1-9 表示。图 1-9 表示当 P1 条件成立时反复执行 A 操作，直到 P1 条件不成立为止。

直到型循环结构用图 1-10 表示。

在初学时，为清楚起见，可如图 1-9 和图 1-10 那样，写明“当 P1”或“直到 P1”，待

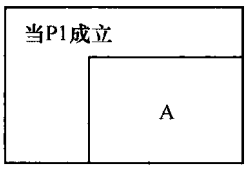

图 1-9

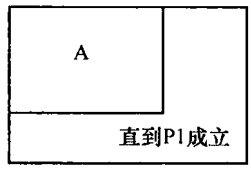

图 1-10

熟练之后，可以不写“当”和“直到”字样，只写“P1”，从图的形状即可知道是当型还是直到型。

【例 1-4】将例 1-1 的算法用 N-S 图表示，如图 1-11 所示。

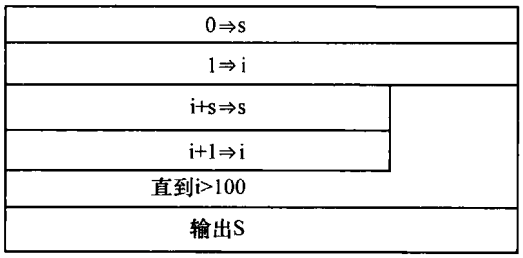

图 1-11

4．用伪代码表示算法

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。它如同一篇文章

一样，自上而下地写下来。每一行(或几行)表示一个基本操作。它不用图形符号，因此书写方便，格式紧凑，也比较好懂，也便于向计算机语言算法(即程序)过渡。

例如，“打印 x 的绝对值”的算法可以用伪代码表示如下：

```
if  x  为正 then
  print  x
else
 print -x
```

它好像一个英语句子一样易懂，在西方国家用得比较普遍。可以用英文、汉字伪代码，也可以中英文混用，即将计算机语言中的关键字用英文表示，其他的可用汉字。总之，以便于书写和阅读为原则。用伪代码写算法并无固定、严格的语法规则，只要把意思表达清楚，并且书写的格式要写成清晰易读的形式。

1.1.3 程序

用计算机语言描述的算法称为计算机程序，简称程序。只有用计算机语言描述的算法才能在计算机上执行。换言之，只有计算机程序才能在计算机上执行。人们编写程序之前，为了直观或符合人类思维方式，常常先用其他方式描述算法，然后再翻译成计算机程序。

1.1.4 程序设计语言

人类社会中有多种语言交流工具，每种语言又都有它的语法规则。人和计算机通信需要通过计算机语言。计算机语言是面向计算机的人造语言，是进行程序设计的工具，因此也称程序设计语言。程序设计语言可以分为机器语言、汇编语言、高级语言。高级语言种类繁多(据统计有上千种)，曾经引起广泛关注和使用的高级语言有 FORTRAN、BASIC、Pascal 和 C 等命令式语言(或称过程式语言)；有 LISP、PROLOG 等陈述式语言；还有当前流行的面向对象的程序设计语言，例如 C++、Java、Visual C++、Visual Basic、Delphi、PowerBuilder 等。

计算机硬件能直接执行的是机器语言程序。汇编语言也称符号语言，用汇编语言编写的程序称汇编语言程序。计算机硬件不能识别和直接运行汇编语言程序，必须由“汇编程序”将其翻译成机器语言程序后才能识别和运行。同样，高级语言程序也不能被计算机硬件直接识别和执行，必须把高级语言程序翻译成机器语言程序才能执行。语言处理程序就是完成这个翻译过程的，按照处理方式的不同，可以分为解释型程序和编译型程序两大类。C 语言采用编译程序，即把用 C 语言写的“源程序”编译成“目标程序”，再通过连接程序的连接，生成“可执行程序”才能运行。

1.2 C 语言的发展与特点

1.2.1 C 语言的发展

在 C 语言诞生以前，系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差；但一般的高级语言又难以实现对计算机硬件的直接操作(这正是汇编语言的优势)，于是人们盼望出现一种兼有汇编语言和高级语言特性的新语言。C 语言是贝尔实验室于 20 世纪 70 年代初研制出来的，后来又被多次改进，并出现了多种版

本。80 年代初，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 语言的发展和扩充，制定了 ANSI C 标准(1989 年再次做了修订)。

C 语言的发展过程如下：

ALGOL60→CPL→BCPL→B→C→标准 C→ANSI C→ISO C。

ALGOL60：一种面向问题的高级语言。ALGOL60 离硬件较远，不适合编写系统程序。

CPL(Combined Programming Language，组合编程语言)：CPL 是一种在 ALGOL60 基础上更接近硬件的语言。CPL 规模大，实现困难。

BCPL(Basic Combined Programming Language，基本的组合编程语言)：BCPL 是对 CPL 进行简化后的一种语言。

B 语言：是对 BCPL 进一步简化所得到的一种很简单接近硬件的语言。B 语言取 BCPL 语言的第一个字母。B 语言精练、接近硬件，但过于简单，数据无类型。B 语言诞生后，UNIX 开始用 B 语言改写。

C 语言：是在 B 语言基础上增加数据类型而设计出的一种语言。C 语言取 BCPL 的第二个字母。C 语言诞生后，UNIX 很快用 C 语言改写，并被移植到其他计算机系统。C 语言是在 20 世纪 70 年代初问世的。1978 年，美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时，B.W.Kernighan 和 D.M.Ritchit 合著了著名的 THE C PROGRAMMING LANGUAGE 一书。后来由美国国家标准学会在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称为 ANSI C。

早期的 C 语言主要是用于 UNIX 系统。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、小和微型计算机上得到广泛使用，成为当代最优秀的程序设计语言之一。

目前最流行的 C 语言有以下几种：

(1) Microsoft C 或称 MS C。

(2) Borland Turbo C 或称 Turbo C。

(3) AT&T C。

这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

C 语言是 C++语言的基础，C++语言和 C 语言在很多方面是兼容的。因此，掌握了 C 语言，再进一步学习 C++语言，就能用熟悉的语法来学习面向对象的语言，从而达到事半功倍的目的。

本教材选用的是 Turbo C 2.0 版本。

1.2.2 C 语言的特点

C 语言发展如此迅速，而且成为最受欢迎的语言之一，主要因为它具有强大的功能。归纳起来 C 语言具有下列特点：

(1) C 语言简洁、紧凑，使用方便、灵活。C 语言一共只有 32 个关键字和 9 种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。

注意：在 C 语言中，关键字都是小写的。

(2) 运算符丰富。C 语言共有运算符 34 种。C 语言把括号、赋值、逗号等都作为运算符处理。从而使 C 语言的运算类型极为丰富，可以实现其他高级语言难以实现的运算。

(3) 数据结构类型丰富。C 语言提供的数据类型有整型、浮点型、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的运算。尤其引入了指针概念，使用十分灵活和多样化，可使程序效率更高。

(4) 具有结构化的控制语句。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；按模块化方式组织程序，层次清晰，易于调试和维护。C 语言的表现能力和处理能力极强。

(5) 语法限制不太严格，程序设计自由度大。例如，对数组下标越界不做检查，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，例如，整型数据与字符型数据可以通用。

(6) C 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。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。因此有人把它称为中级语言。C 语言可以像汇编语言一样对位、字节和地址进行操作，它也可以直接访问内存的物理地址，进行位(bit)一级的操作，还实现了对硬件的编程操作，因此 C 语言既可用于系统软件的开发，也适合于应用软件的开发。

(7) 生成目标代码质量高，程序执行效率高。C 语言一般只比汇编语言生成的目标代码效率低 10%~20%。

(8) 与汇编语言相比，用 C 语言写的程序可移植性好。基本上不做修改就能用于各种型号的计算机和各种操作系统。

但是，C 语言对程序员要求也高，程序员用 C 语言写程序会感到限制少、灵活性大，功能强，但较其他高级语言在学习上要困难一些。

总之，C 语言简洁、紧凑、实用、方便、移植性好、执行效率高、处理能力强、结构化程度高，但对编程人员要求较高，较难掌握，不够安全。

1.3 简单的 C 语言程序介绍

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一个 C 源程序的基本部分和书写格式。

【例 1-5】 输出一行信息。

```
#include <stdio.h>
void main()
{
  printf("This is the first C program .\n");
}
```

程序运行结果：This is the first C program.

程序说明：

(1) include 称为文件包含命令，扩展名为.h 的文件称为头文件，使用标准库函数时应在程序开头一行写：#include <stdio.h>。

(2) main 是主函数的函数名，表示这是一个主函数。每一个 C 源程序都必须有且只能有一个主函数(main 函数)。函数体用花括号{}括起来。

(3) main 前面的 void 表示此函数是“空类型”，void 是“空”的意思，是指执行此函数后不产生一个函数值(有的函数在执行后会产生一个函数值，如正弦函数 sin(x))。

(4) 本例中主函数内只有一个输出语句，printf 函数的功能是把要输出的内容送到显示器显示。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。本例 printf 语句中双撇号内的字符串按原样输出。

(5) “\n”是换行符，即在输出“This is the first C program.”后回车换行。

【例 1-6】从键盘输入一个数 x，求 x 的正弦值。

```
#include<stdio.h>
#include<math.h>
void main()
{
  double x,s;                              /*声明，定义变量为浮点型*/
  printf("input number:");                 /*原样输出提示信息*/
  scanf("%lf",&x);                         /*由键盘获得变量 x 的值*/
  s=sin(x);                                /*计算 x 的正弦值，并赋值给变量 s */
  printf("sine of %lf is %lf\n",x,s);      /*输出结果*/
}
```

程序运行结果：

```
"input number:5↵
        sine of 5.000000 is -0.958924
```

程序说明：

(1) /*……*/表示注释。注释只是给人看的,对编译和运行不起作用。所以可以用汉字或英文字符表示，可以出现在一行中的最右侧，也可以单独成为一行。

(2) 在 main()之前的两行称为预处理命令(详见 8.2 节)。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<>或引号""内指定的文件包含到本程序来，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。因此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。因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件。在本例中，使用了三个库函数：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 文件，因此在程序的主函数前用 include 命令包含了 math.h。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一行的包含命令#include<stdio.h>。

同样，在例 1-5 中使用了 printf 函数，也可省略包含命令。

(3) 在例题中的主函数体中又分为两部分，一部分为说明部分，另一部为分执行部分。说明是指变量的类型说明。例 1-5 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一个特点，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。本例中使用了两个变量 x、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。说明部分后的四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，