高等学校 21 世纪教材

C 语言程序设计

孟庆昌　刘振英　陈海鹏　等　编著

人 民 邮 电 出 版 社

图书在版编目(CIP)数据

C 语言程序设计/孟庆昌等编著.—北京：人民邮电出版社，2002.7
高等学校 21 世纪教材
ISBN 7-115-09873-5

I. C… II. 孟… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 032130 号

内 容 提 要

本书为高等院校计算机专业的教材。本书全面、系统地介绍了最新 C 语言的语法规则和程序设计应用。全书共分 10 章，循序渐进地介绍了 C 语言的基本概念、各种语法成分及其应用，并通过大量实例程序讲述了 C 语言应用中的重点和难点，引导读者掌握一般程序设计的方法。

全书概念清晰、结构合理、内容严谨、讲解透彻、重点突出、示例典型、实用性强，特别适合作为本科院校学生学习 C 语言的教材，也适用于广大软件开发人员和自学人员。

高等学校 21 世纪教材
C 语言程序设计

◆ 编　　著　孟庆昌　刘振英　陈海鹏等
责任编辑　潘春燕

◆ 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@ptph.com.cn
网址　http://www.ptpress.com.cn
读者热线：010-67180876
北京汉魂图文设计有限公司制作
北京印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：24.5
字数：591 千字　　2002 年 7 月第 1 版
印数：1－0 000 册　　2002 年 7 月北京第 1 次印刷

ISBN 7-115-09873-5/TP・2618

定价：31.00 元

丛 书 前 言

当今世界，科学技术突飞猛进，知识经济已见端倪，国际竞争日趋激烈。教育在综合国力的形成中处于基础地位，国力的强弱将越来越取决于劳动者的素质，取决于各类人才的质量和数量，这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初，我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体，是进行教学的基本工具，也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此，高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代，计算机教学内容也随之更新，尤其随着教育部“ 高等教育面向 21 世纪教育内容与课程体系改革 ”计划的实施，对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授，精心策划编写了这套“ 高等学校 21 世纪教材 ”。

为真正实施精品战略，组织编写好这套教材，我们在国内高校做了系统、详细的调查，对教育部制订的教育计划做了认真的研究，还对国内外已出版的教材做了理性的分析，确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中，我们多次组织了由专家及高校一线教师参加的研讨会，对现有比较出色的教材的特点及优点进行了分析，博采众长，力求实现教材权威性与实用性的完美结合。

本套教材有如下特点：

1．考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况，在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。

2．紧密结合当前技术的新发展，在阐述理论知识的同时侧重实用性。

3．力求在概念和原理的讲述上严格、准确、精练，理论适中，实例丰富，写作风格上深入浅出，图文并茂，便于学生学习。

4．为适应当前高校课程种类多、课时数要压缩的教学特点，教材不仅篇幅有很大的压缩，而且均配有电子教案，以满足现代教学新特点的需要，做到易教易学。

5．所选作者均是国内有丰富教学实践经验的知名专家、教授，所编教材具有较高的权威性。

教育的改革将不会停止，教材也将会不断推陈出新。目前本套教材即将推出，将接受广大教学第一线教师的检验。

由于我们的水平和经验有限，这批教材在编审、出版工作中还存在不少缺点和不足，希望使用本套教材的学校师生和广大读者提出批评和建议，以便改进我们的工作，使教材质量不断提高。

目　录

第 1 章　C 语言概述

一个完整的计算机系统是由硬件和软件两大部分组成的。计算机硬件是指计算机物理装置本身，如处理器、内存及各种设备等。而计算机软件是相对硬件而言的，它是计算机程序、过程、规则以及相关的文档资料的总称，如 Windows 98、Windows NT、UNIX、Linux 和 Word 等都属于软件范畴。简单地说，软件是计算机执行的程序。如果把计算机系统比作一个人，那么硬件是人的躯体、器官，软件是人的灵魂、思想。

程序是可以被计算机处理的指令序列。通常，程序是为完成一项任务、由汇编语言或高级语言编写的代码的集合。程序设计是根据所提出的任务，用某种程序设计语言编制一个能正确完成该任务的计算机程序。例如，用 C 语言编写一个求解线性方程组的程序。

本章简要介绍程序设计、高级语言等基本概念，回顾 C 语言的发展历史及其特点，并通过示例，讲解 C 语言的一般知识及上机实习过程。

1.1　程序设计及编程语言的“代”

1.1.1　程序设计

唱歌要有乐谱。乐谱中规定了什么地方是高音，什么地方是低音，什么地方是连唱，什么地方停顿等等。计算机的动作是执行程序的过程，程序就相当于“乐谱”，而程序设计就是编制“乐谱”的过程。

如何进行程序设计呢？一般说来，包括以下步骤：

① 问题定义——把所要解决的问题、所设计的输入数据以及希望得出的结果等，用日常语言尽可能清晰、完整、准确地表达出来，经过抽象，建立相应的数学模型。

② 算法设计——确定解决问题的方法，并把任务分解成计算机能够执行的几个特定操作。如对排序问题，是采用选择排序、分段变换排序还是快速排序等。

③ 流程图设计——用形象的、适于编写程序的方法表达算法。可用自然语言描述，也可用流程图符号表示，或者将二者结合起来。

④ 程序编制——用选定的语言，按流程图提供的步骤写出程序。

⑤ 程序调试、测试及资料编制——对编完的程序进行编译、运行，查找其出错位置，并予以纠正。对有实用价值的程序，还要测试其正确性及效率等，并编写使用和维护该程序的说明书，供别人参考。

要成为一名好的程序员，除了要熟练掌握一些编程语言(高级语言或汇编语言)和程序算法外，还必须多读多练。

1.1.2 编程语言的“代”

计算机程序设计语言经历了从诞生、发展到逐步成熟、完善和广泛应用的发展过程。目前，广泛流行的编程语言就有几十种，如 C、Fortran、Pascal、Ada、Java、C＋＋、VB、VC 等。按照编程语言产生的日期、功能、结构和应用，它们可以划分成不同的“代”。多数专家认为，计算机语言大致可以分为以下五代。

1．第一代语言——机器语言

这是最初使用计算机的编程语言。人们使用计算机的机器指令来进行程序设计。每条机器指令都以机器能识别的“0”、“1”码(即二进制代码)和地址来表达,如：00110101、00001000、11010011 和 00001001 等，但这种代码难写、难读、难理解，不具备移植性。

2．第二代语言——汇编语言

汇编语言是机器指令的助记形式。它用字母、数字等符号表示代码，用符号地址代替物理的机器地址。如：ADD 0，5。汇编语言中的指令称为汇编指令，它们和机器指令是一一对应的。汇编语言与机器语言一样都是面向机器的语言，缺乏易读性，无法进行移植。

3．第三代语言——高级语言

高级语言是与人的思维和表达问题方法的形式比较接近的编程语言。高级语言与上述面向机器的语言相比，较少使用原始符号。用高级语言编写的程序不能直接在机器上运行，必须经过解释程序或者编译程序的转换，才能变成相应的机器语言。第三代语言是面向过程的语言，利用它解题时，不仅要考虑算法的逻辑，还要用语句描述出实现的过程。很多著名的语言都属于第三代语言，如 BASIC、Fortran、Pascal 和 C 语言等。

4．第四代语言

第四代语言是用来快速开发事务处理系统的高产软件工具。这类语言区别于前三代语言的最主要特征之一是非过程化，它面向应用，仅需要说明“做什么”，而不必描述每一步“如何做”。另外，许多第四代语言与数据库的关系十分密切，能够对大型数据库进行高速处理。如 Excel、INFORMIX-4GL、PowerBuilder、FoxPro-4GL、FoxBASE-4GL，以及面向对象的程序设计语言，如 SMALLTALK、C＋＋和 ADA 等。

5．第五代语言

第五代语言主要是为人工智能领域应用而设计的语言，如 PROLOG、LISP 等。这类语言具有很强的推理功能，适合于书写自动定理证明、专家系统、自然语言理解和关系数据库等程序。

随着计算机应用领域的不断扩大和软件开发技术的快速发展，将会涌现出功能更强、更符合人类思维的计算机高级语言。

1.2 C 语言的发展历史和特点

C 程序设计语言(简称 C 语言)是国际上最著名的高级程序设计语言之一，也是使用范围最广的计算机编程语言之一。它不仅可以书写如操作系统、数据库之类的系统软件，而且还可以用来编写各种应用软件。从个人电脑到巨型机，从 DOS 到 UNIX 系统，C 语言几乎可以在

所有的软件工作平台上运行。

1.2.1　C 语言的发展历史

C 语言是 UNIX 系统的主力语言，它与 UNIX 系统有着互相依存、休戚与共的紧密关系。UNIX 系统是美国贝尔实验室的 K.Thompson(汤普森)和 D.M.Ritchie(里奇)从 1969 年开始，用不到两年的工作时间研制成功的。该系统最早运行在 DEC 的 PDP—7 机器上，用汇编语言编写。由于用汇编语言编写的程序不可移植，其描述问题的效率大大低于高级语言，而且可读性又差,所以 K.Thompson 在 1970 年开发出一种高级程序设计语言 B,于 1971 年在 PDP —11/20 上实现了 B 语言，并用 B 语言编写了 UNIX 操作系统和绝大多数的实用程序。B 语言的主要思想来源于 M.Richard(里查德)提出的 BCPL 语言。由于 B 语言是面向字存取，而且具有功能过于简单、数据无类型、运行速度较慢等弱点，因此它并未得到广泛流行。1972 年 D.M.Ritchie 在 B 语言的基础上开发出 C 语言。C 语言克服了 B 语言的缺点，又保持了它的精炼、接近硬件的优点，同时扩充了很多适于系统设计和应用开发的功能。1973 年 K.Thompson 和 D.M.Ritchie 把 UNIX 系统用 C 语言重写了一遍,并加入多道程序设计等新的功能,这就是 UNIX 的第五版。在初期阶段，C 语言还仅限于在贝尔实验室内部使用。到 1975 年，随着 UNIX 第六版的公布并且免费推向世界，C 语言的一系列优点才引起人们的广泛关注。伴随 UNIX 系统从研究室、院校走向商业社会，登上世界舞台，C 语言也逐步为世人所认识、欢迎和推广使用。特别是“可移植 C 语言编译程序”在 1977 年推出之后，大大促进了 C 语言独立于 UNIX 系统而运行的能力。目前，几乎在各种硬件平台上，在形形色色的软件环境下，C 语言都是程序设计人员解决各类问题的最主要的编程工具。

图 1-1 给出几种主要语言的派生关系。

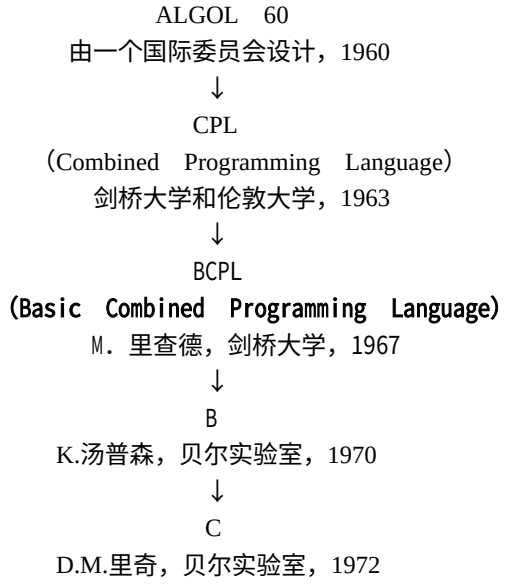

图 1-1　几种主要语言的派生关系

1978 年，B.W.Kernighan 和 D.M.Ritchie 合著了《The C Programming Language》一

书，该书中讲述的 C 语言语法成为后来广泛使用的 C 语言版本的基础，一般把它称为经典 C。随着微型机的迅速普及，出现了一大批 C 语言系统。虽然它们之间具有很好的兼容性，但由于没有统一的标准，必然存在若干差异，从而给程序在不同平台间的移植带来困难。为了改变这种局面，ANSI(American National Standards Institute)于 1983 年设立了一个委员会，专门制定 C 语言标准，这就是常说的 ANSI C。ANSI C 与经典 C 相比，有了很大的发展。B.W.Kernighan 和 D.M.Ritchie 在 1988 年对他们的经典著作《The C Programming Language》进行了修订,使之符合 ANSI C 标准。随着制定以 UNIX 为基础的操作系统标准 POSIX(Portable Operating System Interface for Computer Environments)这一工作的开展，ISO 于 1990 年通过了 C 程序设计语言的国际标准，称之为标准 C。它是以 ANSI C 为基础，吸收了国际上的意见后制定的。从此，C 语言就有了统一的国际标准，这标志着 C 语言进入了一个崭新的时代。

1.2.2 C 语言的特性

C 语言具有很多显著的优点，显示出强大的生命力。下面对 C 语言的主要特点作些介绍。

1．语言表达能力强

C 语言是面向结构化程序设计的高级语言，有良好的通用性，可以在各种硬件平台上运行；它可以直接处理字符、数字和地址，可以完成通常由硬件设备实现的算术、逻辑运算(如位运算、字节运算，对地址的有关操作等)；可以充分地反映出当前计算机的性能，能取代汇编语言编写各种系统软件和应用软件。由于 C 语言既具有高级语言的功能，又具有低级语言的很多特性，所以人们往往把 C 语言称为“中级语言”。这并不意味着 C 语言功能差，恰恰相反，它表明 C 语言把高级语言的基本结构与低级语言的实用性两者有机地结合起来。

2．语言简洁，使用方便、灵活

C 语言在表示方式上力求简单易行，如：用一对花括号“{ }”表示复合语句的 BEGIN、END，利用赋值运算符(如＋=、－=、*=、/= 等)表示进行相应运算并且将结果赋给左值(赋值号左边的变量)，等等。另外，C 语言把一般语言的许多成分都通过显式函数调用来完成，使得编译程序相对小而精。又如，C 语言本身没有提供输入/输出机制，也没有并行操作、同步或协同程序等复杂控制，而是提供了大量而有效的库函数来实现输入/输出、字符串处理及存储分配等功能。C 语言对库函数也可以根据需要方便地予以扩充，这样，就使得其编译程序小巧、紧凑。

3．运算符丰富

C 语言是一种表达式语言，它有功能很强的运算符，如：增 1 运算符(＋＋)、减 1 运算符(－ －)、取地址运算符(&)和间接运算符(*)等，用这些运算符可构成书写简洁而功能很强的表达式，从而提高软件生产效率。C 语言中共有 42 个运算符。由于 C 语言运算类型极其丰富，从而使得表达式的类型灵活、多样，在其他高级语言中难以实现的运算，在 C 语言中都能很容易地办到。

4．生成的代码质量高

高级语言能否用来描述系统软件，特别是操作系统、编译程序等，除了要求该语言表达能力强之外，很重要的一个因素是：语言生成的目标代码质量如何。如果代码质量低，系统开销就大，那就失去了实用价值。许多试验表明，针对同一问题，用 C 语言编写的程序，一

般所生成的目标代码的效率仅比用汇编语言生成的目标代码的效率低 10%～20%。由于用 C 语言描述问题，其算法比用汇编语言简单、快捷、工作量小，可读性好，易于调试、修改和移植，因而 C 语言就成为人们描述系统软件和应用软件比较理想的工具。在代码质量方面，C 语言确实可与汇编语言媲美。

5．具有良好的可移植性

用 C 语言编写的程序很容易进行移植。在一种工作平台上开发的软件只需稍作修改，甚至不做任何修改，就可以在其他工作平台上运行，从而使开发的软件独立于具体的计算机体系结构和软件环境，达到共用的目标，提高了软件的生产效率，节省了用户的投资。

6．具有结构化语言特征

C 语言虽不是严格定义的“ 模块结构语言 ”，但其结构类似于 ALGOL、Pascal 和 Modula-2 等语言，具有结构化语言的一系列特征。所以，通常仍将 C 语言称为结构化语言。结构化语言的一个显著特点是实现代码和数据的隔离，从而使程序之间很容易实现程序段的共享。C 语言的主要结构成分是函数，利用分隔化的函数，调用者仅需知道它实现什么功能，而不必知道其功能是如何实现的(当然，过多使用外部变量会产生某些副作用)。C 语言具备现代化语言的各种数据结构，既有简单数据类型，如整型、字符型、浮点型和双精度型等，又有聚合类型，如数组、结构、指针等。C 语言具有结构化的控制语句，直接支持多种循环结构。在书写格式上允许采用逐层缩进形式，增加了程序的可读性。在 C 语言中使用复合语句也同样支持实现程序的结构化和分隔化。当今人们普遍认为，C 语言层次清晰，结构紧凑，比非结构化语言(如 Fortran、BASIC)更易于使用和维护。

C 语言还有其他一些优点，读者可在学习及实践中体会。当然，C 语言也和其他语言一样，存在不足之处，如：某些运算符优先顺序与习惯不完全一致；类型转换比较随便，不是强类型语言等。尽管如此，相比之下，C 语言仍不愧为优秀的程序设计语言之一。

1.3　C 程序示例

下面用几个具体的 C 程序作示例，简要介绍 C 程序的结构和各个成分的构成及作用，使读者对用 C 语言编写程序有一个感性认识。

例 1-1：计算两个给定的整数之和。

```
#include  <stdio.h>

int  main()
{
      int  a, b, sum ;
      a=8 ;
      b=1000 ;
      sum=a＋b*2 ;
      printf("sum=%d\n", sum) ;
      return (0) ;
```

```
}
```

这个程序执行后，输出的结果是：

```
sum=2008
```

现在对这个程序进行一些解释：

第 1 行——#include <stdio.h>，是文件包含行。它表示本程序中所用到的某些常量或宏定义在头文件 stdio.h 中进行了定义。

第 2 行——int main()，是函数首部，它告诉系统这个函数的名称是 main。在 main 后面紧跟着一对圆括号，从而向 C 编译程序表明这是函数名。main 是 C 语言中标识主函数的专用名，表示该 C 程序从这里开始执行。而 main 前面的 int 是表示类型的关键字，它表示整型量。

第 3 行只有一个开花括号“{”，它等同于 Pascal 语言中的 BEGIN，而第 10 行的闭花括号“}”等同于 END。这一对花括号往往被称为语句括号。它们把一组数据说明和执行语句括在一起，构成这个函数的函数体。在编写程序时要注意：在任何情况下，开花括号与闭花括号都必须成对出现，不允许某一个出现的次数多于另外一个。

第 4 行至第 9 行构成这个函数的函数体。第 4 行

```
int a, b, sum;
```

是数据说明语句，即：定义 *a*、*b* 和 sum 是三个整数类型(由 int 标志)的变量。说明语句的末尾必须以分号结束。

第 5 行至第 7 行是三个赋值语句。在 C 语言中，赋值符是“=”，其作用是先计算赋值符右边表达式的值，然后赋给其左边的变量。于是，左边变量的值就是右边表达式计算的结果。如第 7 行，先把 *b*(值为 1000)乘以 2，然后与 *a*(值为 8)相加，得到 2008，最后赋给 sum。这样，sum 的值就是 2008。

注意：在每一个语句的后面都带一个分号(；)。这是 C 语言的规定，因为在 C 语言中，分号是语句终止符，是语句的一个组成部分，而不是一般意义上的分隔符。所以，一个语句末尾的分号是不可省略的。

第 8 行 printf(…)； 是一个函数调用语句。在 printf 后面紧跟有一对圆括号，表示它是一个函数。这里是对 printf 函数进行调用，由它输出运算的结果(即 sum 的值)。其实，printf 是 C 语言标准程序库中的一个函数，用来进行格式输出。本例中，圆括号内有两个实参，二者以逗号分开。第一个实参是用一对双引号括起来的字符串，表示有关输出格式的控制信息。这里“%d”是输出转换的标志，表示把第二个实参(即 sum)的值按十进制整数形式输出。“%d”前的字符串按原样输出。所以，该程序运行后输出的结果是：sum=2008。

在格式控制字符串的最后是反斜线“\”和字母 n，这两个字符合起来作为一个换行字符。在 printf()函数执行过程中，当遇到换行字符时，屏幕上的光标(或打印机的打印头)就移到下一行的开头。因此，在换行字符后的任何字符，在终端或显示屏幕上都出现在下一行中。

第 9 行是 return 语句，它将其后的值返回。由于它返回的值是整数 0，所在以 main()的前面有关键字 int。如果该函数没有返回值，那么在 main()的前面就带有关键字 void，表示该函数是无值的，在以下的示例中经常见到这种情况。详细说明见 6.6.1 节。

应该指出：实际上 C 程序没有行号，如例 1-1 程序中所显示的那样。书中为了讲解方便，免得重复写出代码，只要一提到第几行，就在程序中很快找到具体行的内容，省去每次都要

从程序开头一行一行向下数的麻烦，所以，在以后的某些示例中，我们采用“ 人为加行号 ”的形式，即在程序的最左一列上标注有行号。这里再次提请读者注意：在你试图上机运行书中所举示例时，不要录入行号；在你编写 C 程序时也不要加行号。

例 1-2：计算半径为 *r* 的圆的面积。

```
1  /* Calculating  the  area  of  a  circle. */
2  #include  <stdio.h>
3  #define  PI  3.14
4
5  int  main()
6   {
7        float  r;    /* radius  of  a  circle */
8        float  area; /* area  of  the  circle */
9        printf("Input：r=？\n");
10       scanf("%f", &r);
11       area=PI*r*r;
12       printf("The area is %f\n", area);
13       return (0);
14  }
```

本程序的第 1 行是注释行。在 C 语言中，注释行是以“ /* ”开头、以“ */ ”结尾的任意字符串，可以是英文、汉字或汉字拼音等。在第 7 行和第 8 行中也都使用了注释。

注释的目的是为了增加程序的可读性。它是编程人员对整个程序或者某些特定的程序段以及某些重要的数据所作的说明，如程序的功能、采用的算法、数据的含义等，这样，既方便其他人员对你所写的程序进行阅读、分析，也方便你以后对它进行修改。所以，一个程序员应养成“ 写程序时加注释 ”的良好习惯，并提倡尽量多用。

正如本例中所示，注释可出现在程序的任何位置上。它仅仅起解释或说明的作用，在编译时并不生成目标码。

在使用注释时，要注意以下几点：

① /*和*/要成对出现，并且在字符“ / ”和字符“ * ”之间不能插入空格；

② 注释不能嵌套，就是说，不能在注释中间又有注释，例如：

/*……/* ……*/……*/

这种形式是不允许的。

如果注释内容较多，在一行中写不下，需占用几行时，可以采用下面形式：

```
/*……
**……
**……
*/
```

③ 注释不要插到一个字符常量(如'A'、'\n')或一个字符串常量(如"abc"、"Hello! ")的中间。

本程序的第 3 行是宏定义行(详见第 10 章)。定义字符串 PI 表示常量 3.14。这样，在程

序中本来应出现 3.14 的地方就都以 PI 代替，从而增加程序的可读性和可移植性。

第 7 行和第 8 行是数据说明语句，定义变量 *r* 和 area 都是 float(浮点)类型。

第 9 行调用 printf()函数，用来输出提示信息：Input：r=？其作用是提醒你：下面要输入圆半径 *r* 的值。

第 10 行调用 scanf()函数。scanf()是标准 I/O 库中的一个函数(详见第 10 章)，它接收用户从键盘上输入的浮点数，并赋给变量 *r*。其中第一个参数“ %f ”表示接收的数据是浮点量，第二个参数“ &r ”表示变量 *r* 的地址。

第 11 行先计算 PI*r*r，即 3.14*r*r，得到圆的面积，然后把结果赋给变量 area。

第 12 行调用 printf()函数，输出给定半径的圆的面积。

下面是一次实际运行的结果：

```
Input：r=？
6.28 < 回车>
The  area is 123.836586
```

例 1-3：给定等差级数的首项、公差和项数，计算该级数的第 *n* 项值以及前 *n* 项和。

设：首项为 a_1，公差为 *d*，项数为 *n*，则第 *n* 项 a_n：$a_n=a_1+(n-1)d$

前 *n* 项和 s_n：

$$s_n=na_1+\frac{n(n-1)}{2}d$$

程序如下：

```
1  /*Calculating nth item of the arithmetical series and the sum of its n items*/
2  #include  <stdio.h>
3  int  n_item(int,int,int);/*Function prototype*/
4  int   sum(int,int,int);/*Function prototype*/
5
6  int  main()
7  {
8        int a1, n, d, an, sn;
9        printf("Input data:a1, n, d\n");
10       scanf("%d%d%d", &a1, &n, &d);
11       an=n_item(a1, n, d);
12       sn=sum(a1, n, d);
13       printf("an=%d\n", an);
14       printf("sn=%d\n", sn);
15       return (0);
16 }
17
18 /* Calculating n_th item of the arithmetical series */
19 int  n_item(int  a, int  x, int  k)
20 {
21     int b;
```