工业和信息化普通高等教育“十二五”规划教材
21世纪高等学校计算机规划教材　　河南省“十二五”普通高等教育规划教材

C语言程序设计（第2版）

C Language Programming (2nd Edition)

贾宗璞 许合利 主编

- 教学改革结晶，一线教师智慧
- 内容系统完整，案例典型丰富
- 训练思维能力，掌握编程艺术

高校系列

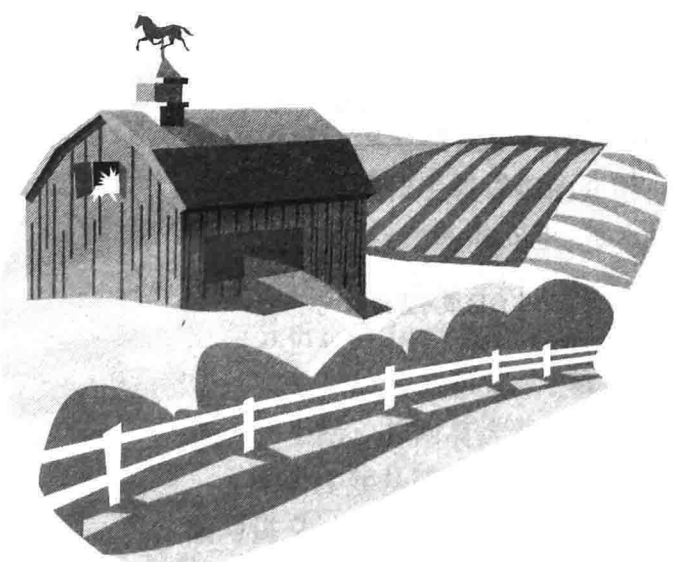
人民邮电出版社
POSTS & TELECOM PRESS

工业和信息化普通高等教育“十二五”规划教材
21世纪高等学校计算机规划教材　　河南省“十二五”普通高等教育规划教材

C语言程序设计（第2版）

C Language Programming (2nd Edition)

贾宗璞 许合利 主编



高校系列

人民邮电出版社

图书在版编目（CIP）数据

C语言程序设计 / 贾宗璞，许合利主编. -- 2版. -- 北京 : 人民邮电出版社，2014.9
21世纪高等学校计算机规划教材
ISBN 978-7-115-36357-2

Ⅰ. ①C… Ⅱ. ①贾… ②许… 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(2014)第167888号

内 容 提 要

本书为工业和信息化普通高等教育“十二五”规划教材和河南省“十二五”普通高等教育规划教材，是高等院校计算机基础教育教材。全书共分 14 章，主要内容包括：C 语言概述，基本数据类型、运算符与表达式，顺序结构程序设计，选择结构程序设计，循环结构程序设计，数组，函数，编译预处理，指针，结构体、共用体及枚举类型，文件，C++基础，VC++ 6.0 开发环境及程序测试与调试，上机实验内容等；各章后均附有大量习题。书后附有完整的 ASCII 代码对照表、C 语言中的关键字、运算符优先级和结合方向、常用库函数。

本书内容丰富、新颖，图文并茂，通俗易懂，实用性强，可作为高等学校非计算机专业的计算机基础课教材，也可作为应用计算机人员的学习参考书。

◆ 主　　编　贾宗璞　许合利
责任编辑　邹文波
责任印制　彭志环　杨林杰

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京圣夫亚美印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：20　　2014 年 9 月第 2 版
字数：524 千字　　2014 年 9 月北京第 1 次印刷

定价：39.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315

第 2 版前言

C 语言是国际上广泛流行的一种面向过程的计算机高级语言，其历史悠久，发展相当迅速。后来发展起来的 C++、Java、C#等语言，无不是在其基础上进行扩充或改造的。C 语言与其他高级语言相比，形式简洁，数据类型丰富，表达能力强，运算丰富，程序设计灵活，可读性和可移植性好，目标程序效率高，既具有低级语言的特点，又具有完善的模块化结构，体现了结构化程序设计的思想，适合于培养良好的编程风格和优秀的程序设计技术的训练。它是继 PASCAL 语言之后的又一门优秀的课程教学语言，并且是教学需要与实际应用相结合的一门语言。C 语言具有很强的处理功能，不仅用于开发系统软件，也可用于开发应用软件。

学习 C 语言，起初会觉得要记的东西太多，这是由于它太灵活了。但是学到一定程度，就会尝到甜头，就会体会出 C 语言的特色。C 语言中的指针是一个核心，是今后开发工作中的得力助手，因为在使用 C/C++的实际工作中指针无处不见，很多参数完全就是指针化的。虽然 Java 从安全性方面考虑摒弃了指针，但从性能上来说，却得不偿失。要学好 C 语言，就要透彻理解概念，辅之以大量编程训练和上机实验。只靠看书学不好 C 语言，要积极实践，善于思考，结合具体的项目（哪怕是很小的项目）学用相长。坚持下去，就会成功。

本书是根据教育部高等学校计算机科学与技术教学指导委员会编制的《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求(试行)》编写的，属于较高要求的 C 语言版本。全书共分 14 章，第 1 章介绍了 C 语言的基本知识、算法及程序设计的一般方法；第 2 章介绍了 C 语言的基本数据类型、常量和变量、运算符以及表达式；第 3 章、第 4 章、第 5 章介绍了 C 语言进行结构化程序设计的基本方法，包括结构化程序的顺序结构、选择结构、循环结构及其设计方法；第 6 章介绍了数组；第 7 章介绍了函数；第 8 章介绍了编译预处理；第 9 章充分阐述了 C 语言的指针；第 10 章对结构体、共用体及枚举类型作了较详细的介绍；第 11 章对 C 语言文件操作作了较详细的阐述；第 12 章对 C++的基本特性、基本概念以及面向对象程序设计的基本方法进行了初步阐述；第 13 章介绍了 VC ++6.0 开发环境、程序测试与调试及常见出错信息与分析；第 14 章提出了上机实验总的目的和要求，并为了配合理论教学介绍了各实验项的目的和要求、实验内容及操作步骤。另外，每章都配有本章重点、难点和小结，并附有大量的习题，在书末还有附录来帮助大家的学习。

编者多年从事 C 语言及计算机相关课程的教学实践，第 1 版成稿前曾多次编写讲义、教辅资料、习题集。本书是在此基础上，经过认真讨论，集思广益，精心整理、编写而成的。本书在内容选取上既注重了先进性、科学性和系统性，又兼顾了实用性和趣味性；在文字叙述上力求做到深入浅出，通俗易懂，便于自学；同时用大量的典型实例化解各章的难点，充分展示了计算机解决问题的

思想和方法，突出了程序设计的基本方法的阐述，注意计算思维的训练。另外，编者多年来一直参与全国计算机等级考试的组织、辅导工作，对全国计算机等级考试的大纲有透彻的理解，所以本书将大纲中二级 C 语言的要求贯穿其中。习题包含了等级考试的选择题、程序填空题、程序修改题和程序设计题等。因此，本书除了可以作为普通高校各专业学生的教材外，还可以作为参加全国计算机等级考试（二级 C 语言）的参考用书，也可以供学习 C 语言的科技人员使用。

本书第 2 版吸收了任课教师的意见和建议，对第 1 版进行了修订和补充，主要包括文字叙述、程序代码、常见编译与连接出错信息和上机实验等。

本书的编写人员（除明确指明单位的编者外）均为河南理工大学多年从事计算机教学的教师。主编为贾宗璞、许合利。副主编为赵珊、焦阳（河南牧业经济学院）。具体分工为贾宗璞编写第 1 章、第 7 章，许合利编写第 2 章、第 4 章，焦阳编写第 3 章、第 12 章，刘本仓编写第 5 章、第 13 章，王国伟编写第 6 章、第 11 章，吴志强编写第 8 章、第 14 章，赵珊编写第 9 章，文运平编写第 10 章及附录。在本书的编写过程中，得到了河南理工大学领导和教务处的大力支持，在此表示衷心感谢。

由于编者水平有限，书中缺点错误在所难免，敬请读者批评指正。

编　者

2014 年 7 月

目 录

第 1 章 C 语言概述

本章重点：

※ C 语言的特点

※ C 语言程序的基本结构

※ 算法及其描述方法

※ 结构化程序设计方法

本章难点：

※ C 语言与其他高级语言的区别

※ 算法的流程图、N-S 图描述方法

计算机本身是无生命的机器，要使之为人类完成各种各样的工作，就必须让它执行预先设计的相应程序。这些程序都是用程序设计语言编制出来的。在众多的程序设计语言中，C 语言有其独特之处，深受软件工作者欢迎。本章主要从程序设计的角度，介绍 C 语言的发展及特点，描述 C 语言程序的基本结构、算法以及程序设计方法等。

1.1 C 语言的发展及特点

1.1.1 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，所以其可读性和可移植性都很差；而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），于是人们盼望能有一种兼有汇编语言和高级语言特性的新语言。

C 语言就是在这种背景下于 20 世纪 70 年代初问世的，当时主要是用于 UNIX 系统的开发。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 用汇编语言开发而成的。后改用 B 语言实现，但 B 语言过于简单，功能有限。为了更好地描述和实现 UNIX 操作系统，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最初的 C 语言。几经修改后，1978 年由美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《The C Programming Language》一书，通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的标准 C 语

言，后来由美国国家标准化协会 ANSI(American National Standards Institute)在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称之为 ANSI C。

随着人们对 C 语言的强大功能和各方面优点的逐步了解，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了广泛的使用，成为当代最优秀的程序设计语言之一。

近年来，由于开发大型软件的需要，C++在我国逐步得到推广。C++是面向对象的程序设计语言，其基础是 C 语言，且二者在很多方面是兼容的。学习 C++要比 C 语言困难得多，并且也不是所有的人都去编写大型软件。因此，在发达国家的大学中，C 语言仍然是一门重要的课程。掌握了 C 语言，再去学习 C++，就会达到事半功倍的效果。

因为 C++是由 C 语言发展而来的，C++将 C 作为一个子集包含了进来。所以，使用 C 语言编写的程序可以用 C++编译系统进行编译。C/C++的编译系统有很多，如 GCC(GNU Compiler Collection)家族的 Dev-C++(Mingw32)、Cygwin，Borland 公司的 Turbo C 2.0（只支持 C 语言）、Borland C++，微软公司的 Visual C++，以及 Intel C/C++ 5.0、VectorC 1.3.3、LCC—Win32 等，它们不仅实现了 ANSI C 标准，而且还各自作了一些扩充，使之更加方便、完美。但是，它们在实现 C 语言时略有差异，请读者参阅相应的手册，注意自己在上机时所使用的 C 编译系统的特点和规定。

本书叙述以 ANSI C 为主，并简单介绍了 C++。上机实验使用 Visual C++ 6.0（以后简称 VC 6.0 或直接简称 VC），与全国计算机等级考试 C 语言环境一样。在介绍时，如果是 VC 的规定，则会专门指出。

1.1.2 C 语言的特点

C 语言是一种优秀的具有很强生命力的高级程序设计语言，与其他程序设计语言相比，主要特点如下。

（1）C 语言简洁、紧凑，使用方便、灵活。ANSI C 一共只有 32 个关键字（见附录Ⅱ），如 int、long、float、if、while、do 等 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。

（2）运算符丰富。共有 34 种运算符（见附录Ⅲ）。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，可以方便地实现其他高级语言难以实现的功能。

（3）数据结构类型丰富，具有现代语言的各种数据结构。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。能实现各种复杂数据结构（如链表、树、栈等）的运算。尤其是指针类型数据，使用起来更为灵活、多样。

（4）具有结构化的控制语句。用函数作为程序的基本单位，便于实现程序的模块化。C 语言是良好的结构化语言，符合现代编程风格的要求。

（5）语法限制不太严格，程序设计自由度大，如对数组下标越界不做检查；对变量的类型使用比较灵活，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。因此有人把它称为中级语言。

（7）生成目标代码质量高，程序执行效率高，可达到汇编语言程序的 80%。

（8）与汇编语言相比，用 C 语言写的程序可移植性好。

C 语言是理想的结构化语言，描述能力强，且现在的操作系统课程多结合 UNIX 讲解，而 UNIX 与 C 不可分，因此，C 语言已经成为被广泛使用的教学语言。C 除了能用于教学外，还有广阔的应用领域，因此更有生命力。PASCAL 和其他高级语言的设计目标是通过严格的语法定义和检查来保证程序的正确性，而 C 则是强调灵活性，使程序设计人员能有较大的自由度，以适应宽广的应用面。“限制”与“灵活”是一对矛盾。限制严格，就失去灵活性；而强调灵活，就必然增加了出错的可能性。一个不熟练的程序设计人员，编写一个正确的 C 程序可能会比编写一个其他高级语言程序更难一些。也就是说，对使用 C 语言的人，要求对程序设计更熟练一些。总之，C 语言对程序员要求较高，但程序员使用 C 语言编写程序会感到限制少，灵活性大，功能强，可以编写出任何类型的程序。现在，C 语言已不仅用来编写系统软件，也用来编写应用软件，因此学习和使用 C 语言的人已越来越多。

1.2　C 语言程序的基本结构

所谓程序，就是一系列遵循一定规则和思想并能正确完成指定工作的代码。使用 C 语言编写的程序称为 C 语言源程序（简称 C 语言程序或 C 程序）。下面介绍两个 C 程序，虽然其中涉及的内容还未介绍，但从中可了解到 C 程序的基本结构和书写格式。

例 1.1　求两个整数之和。

```
#include <stdio.h>                                  /*编译预处理*/
main()                                              /*主函数*/
{
    int a,b,sum;                                    /*定义整型变量 a,b,sum */
    printf("Please input two integers:\n");         /*输出提示信息，增强程序交互性*/
    scanf("%d%d",&a,&b);                            /*输入两个整数，并赋给 a,b */
    sum=a+b;                                        /*将 a+b 的结果赋给 sum */
    printf("%d+%d=%d\n",a,b,sum);                   /*输出结果*/
}
```

程序运行情况如下：

```
Please input two integers:
6 ␣ 8↙                    (␣表示空格，↙表示按 Enter 键)
6+8=14
```

以上程序中，各行右侧用“/*”和“*/”括起来的内容为注释部分，说明该行或程序段的含义。注释可以增加程序的可阅读性，程序不执行注释部分。

例 1.1 中只包含一个主函数。主函数是一个特殊的函数，C 语言规定必须用 main 作为主函数名。每一个 C 语言程序都必须有且只能有一个主函数。函数是具有特定功能的程序模块，由函数首部（起始行）和函数体构成。此例函数首部为 main()。函数体是函数首部后面用一对花括弧“{}”括起来的部分，此例函数体内有五条语句，每条语句末尾用一分号“;”作为语句结束标志。第一句为声明语句，表示定义了 a、b 和 sum 等整型（int）变量；其他四句为执行语句。第二句和第五句中用到了 printf 函数，它是 C 语言的一个标准库函数（系统已编写好，用户可直接使用），其功能是把数据按照一定的格式输出到默认输出终端（显示器）。第二句中的 printf 函数输出一个字符串，“\n”是转义字符，表示换行；第五句中的 printf 函数括

号内的部分是用逗号分隔开的参数列表。第一个参数为格式控制字符串，用一对双撇号（" "）括住，表示输出数据的格式；后边的参数为要输出的数据项（a、b、sum）。格式控制字符串中的三个“%d”表示后面的对应输出项以整型格式输出，其他字符为普通字符，原样输出。第三句中使用的 scanf 函数也是 C 语言的一个标准库函数，本语句的作用是从默认输入终端（键盘）输入两个整数并分别赋给变量 a 和 b。与 printf 函数类似，scanf 函数的第一个参数为输入数据的格式，也用双撇号括住；其后为变量的地址。有关这两个库函数的详细使用方法，将于第 3 章做进一步介绍。

该程序的第一行是一个编译预处理命令，表示把 include 后尖括号“<>”内的文件 stdio.h（头文件）直接包含到本程序中，成为本程序的一部分，因这些工作是在程序编译之前完成的，故得名编译预处理（详见第 8 章）。编译预处理命令不是 C 程序的语句，行尾不能加“;”。C 语言要求在使用函数前必须声明，而文件 stdio.h 中有 scanf 和 printf 等标准库函数的函数声明，因此，在调用这些库函数之前，就应该包含 stdio.h。

例 1.2 求两个整数中的较大者。

```
#include <stdio.h>                                  /*包含头文件*/
main()                                              /*主函数首部*/
{
    int x,y,z;                                      /*定义整型变量 x、y、z */
    int max(int a,int b);                           /*声明函数 max*/
    printf("Please input two integers:\n");         /*输出提示信息*/
    scanf("%d,%d",&x,&y);                           /*输入 x,y 值*/
    z=max(x,y);                                     /*调用 max 函数*/
    printf("The maximum number is %d.\n",z);        /*输出结果*/
}
int max(int a,int b)                                /*max 函数首部*/
{
    if(a>b) return a;                               /*条件语句，实现选择结构*/
    else return b;                                  /*把结果返回主调函数*/
}
```

程序运行情况如下：

```
Please input two integers:
6,8↙
The maximum number is 8.
```

例 1.2 程序共有两个函数：一个是主函数 main，一个是自定义的函数 max。在 main 函数中，定义了三个整型变量 x、y、z，x 和 y 用来存放两个整数，z 用来存放其中的较大者。接下来的第 5 行对 max 函数进行声明。程序的第六行调用 printf 函数在显示器上输出提示信息，请操作人员输入两个整数。第七行调用 scanf 函数，接受键盘上输入的数并赋给变量 x 和 y。第八行调用自定义函数 max，求出 x 和 y 中的大者并赋给变量 z。第九行调用 printf 函数输出变量 z 的值，即 x 和 y 中的较大值。

max 函数前面的 int 表示此函数是整型类型，即执行此函数后产生一个整型的函数值，由 return 语句将这个函数值返回到 main 函数中调用 max 函数的位置。max 函数的函数体中的 if 结构（详见第 4 章）用来完成求两个整数中的较大数，并将结果返回给调用函数。

由以上两个例子，可以得出以下结论。

（1）C 程序可由一个或多个函数构成，函数是 C 程序的基本单位。

（2）一个函数由函数首部和函数体组成。函数首部是函数的第一行，包含了函数类型、函数名、函数参数等信息，如例 1.2 中的 max 函数的首部为：int max(int a,int b)。函数可以没有参数，即函数名后面的一对圆括号中间是空的，如 main()，但这一对圆括号不能省略。函数体一般由声明部分和执行部分组成，声明部分是对函数中所要用到的变量的定义和对所调用函数的声明等，而执行部分则是用来完成函数功能的语句段。当然，函数体内可以没有声明部分，如例 1.2 中的 max 函数。关于函数的详细内容将在第 7 章介绍。

（3）C 程序必须有一个且只能有一个 main 函数，即主函数。

（4）一个 C 程序总是从主函数开始执行，而不论主函数在整个程序中位置如何。主函数执行完了，整个程序也就执行完了。

（5）每一个语句都必须以分号结尾，分号是 C 语句的必要组成部分。但编译预处理命令、函数首部（即函数的起始行）和花括号“}”的后面不能加分号。

（6）C 语言本身没有输入输出语句，输入输出功能是用输入输出库函数来实现的。

（7）“/*”和“*/”为注释符，必须成对出现，二者之间的部分为注释内容。注释内容可以用汉字或英文字符表示，用来向用户提示或解释程序的意义。注释可以出现在一行的最右侧，也可以单独成为一行，甚至可以是多行。编译时，不对注释作任何处理。一个好的程序应该有详细的注释。另外，在调试程序中对暂不使用的语句也可用注释符括起来，使编译器跳过不作处理，待调试结束后再去掉注释符。另外，C++编译器提供了单行注释，即以双斜线（//）开头，同一行中斜线右侧的所有内容都是注释。

在书写程序时，从清晰和便于阅读、理解、维护的角度出发，应尽量遵循以下规则，以养成良好的编程习惯。

（1）一个声明或一条语句占一行。当然 C 程序允许一行写多条语句，也允许一条语句写在多行上，且无需续行符。

（2）用{ }括起来的部分，通常表示程序的某一层次结构。{ }一般与该结构语句的第一个字母对齐，“}”最好单独占一行。

（3）低一层次的语句或声明可比高一层次的语句或声明缩进若干格后书写，以便看起来更加清晰，增加程序的可读性。

1.3　算法及其描述

1.3.1　算法的概念

1. 算法——程序的灵魂

一个程序应包括如下两种描述。

（1）对数据的描述，在程序中要指定数据的类型和数据之间的组织形式，即数据结构。在 C 语言中，系统提供的数据结构是以数据类型的形式出现的。

（2）对数据处理的描述，即计算机算法。广义地说，为解决一个问题而采取的方法和步骤，就称为“算法”，它是程序的灵魂。因此，著名计算机科学家沃思（Nikiklaus Wirth）提出一个公式：

程序 = 数据结构 + 算法

实际上，一个程序除了数据结构和算法外，还必须使用一种计算机语言，并在必要的环境支持下，采用合适的程序设计方法来设计。因此，程序可以更完整地表达为：

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境

这四个方面都是一个程序设计人员所应具备的知识。其中，算法是灵魂，数据结构是加工对象，语言是工具，编程需要采用合适的方法。本书旨在叙述怎样编写 C 语言程序，这里只介绍算法的初步知识。

从事各种工作和活动，都必须事先想好步骤，然后一步一步地进行，才能避免产生错乱。对同一个问题，又可以有不同的解决方法和步骤。方法有优劣之分，有的方法只需进行很少的步骤，而有些方法则需要较多的步骤。一般来说，希望采用简单的和步骤少的方法。因此，为了有效地解决问题，不仅需要保证算法的正确性，还要考虑算法的质量。

本书所关心的仅限于计算机算法，即计算机能执行的算法。

2. 算法的分类

计算机算法可分为两大类：数值算法和非数值算法。

数值运算的目的是求数值解，一般的数值运算都有现成的模型，可以运用数值分析方法进行解决，因此对数值运算的算法研究比较深入，算法也比较成熟。人们常常把这些成熟的算法汇编成册（写成程序形式），或者将这些程序存放在磁盘或磁带上，供用户调用。

非数值运算包括的面十分广泛，已远远超过了数值运算，最常见的是用于事务管理领域。非数值运算的种类繁多，要求各异，难以规范化，因此只对一些典型的非数值运算算法（例如排序算法）作比较深入的研究。其他的非数值运算问题，往往需要使用者参考已有的类似算法重新设计解决特定问题的专门算法。

本书不可能罗列所有算法，只是通过一些典型算法的例子，介绍如何设计一个算法，如何描述一个算法。

3. 算法的特性

一个算法应该具有以下特性。

（1）有穷性。一个算法应包含有限的操作步骤，而不能是无限的。事实上，“有穷性”往往指“在合理的范围之内”。如果让计算机执行一个历时上千年才能结束的算法，这虽然是有穷的，但超出了合理的限度，实际上是不可能实现的，因而不能把它看作是有穷的。究竟什么算“合理限度”并无严格标准，由人们的常识和需要而定。

（2）确定性。算法中的每一个步骤都应当是确定的，而不应当是含糊的、模棱两可的。也就是说，算法的含义应当是唯一的，而不应当具有“歧义性”。所谓“歧义性”是指可以被理解为两种以上的可能含义，这样可能导致计算机不知所措。

（3）有零个或多个输入。所谓输入是指在执行算法时需要从外界取得的必要信息，可以理解为算法运行的初始数据。但是，一个算法可以没有输入。

（4）有一个或多个输出。算法的目的是为了求解，“解”就是输出。输出可以是打印、显示，也可以是磁盘输出，甚至是中间结果等。总之，没有输出的算法是没有意义的。

（5）有效性。算法中的每一个步骤都应当能有效地执行，并得到确定的结果。例如，在实数范围内对负数开平方根是不能被有效执行的，只能得到荒谬的结果。

对于不熟悉计算机程序设计的人来说，他们可以只使用别人已设计好的现成算法，只需根据算法的要求给以必要的输入，就能得到输出的结果。对他们来说，算法如同一个“黑盒子”一样，

可以不去了解“黑盒子”内部的结构，只要从外部特性上掌握了算法的作用，即可方便地使用算法。但对于程序设计人员来说，必须学会设计算法，并且根据算法编写程序。

1.3.2　算法的描述方法

描述一个算法，可以用不同的方法来实现。常用的有自然语言、传统流程图、N-S 流程图、伪代码、计算机语言等。下面用一个例子对这几种常见的算法描述方法作一介绍。

例 1.3　求 $1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\cdots+\frac{1}{99}-\frac{1}{100}$ 的值。

本题是求一个级数的值，通过观察可以发现，它是从 1 到 100 作为分母，构成级数的 100 项，而项的符号正负交替出现，所以，可以采取循环的思想，对这 100 项逐一累加，最后的累加和便是所求。

1. 用自然语言描述算法

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。例 1.3 的算法用自然语言描述如下：

步骤 1：预设 sign 为 1（sign 代表项的符号，第一项为正，值为 1）；

步骤 2：累加和 sum 置初值 1；

步骤 3：将之后要加的分母用 deno 表示，赋初值 2（即下一步加的是第二项）；

步骤 4：将 sign 乘以-1 后再赋给 sign（实现正负交替）；

步骤 5：用当前符号 sign 与当前基项（1/deno）相乘得到当前项 term；

步骤 6：将当前项 term 与累加和 sum 相加得新的累加和 sum；

步骤 7：分母 deno 加 1，得下一项分母 deno；

步骤 8：若分母 deno≤100 返回步骤 4；否则输出 sum，算法结束。

用自然语言描述算法通俗易懂，但文字冗长，容易出现“歧义性”。自然语言表达的含义往往不太严格，需要根据上下文才能判断其正确含义。此外，用自然语言描述包含分支和循环的算法，很不方便。因此，除了很简单的问题以外，一般不用自然语言描述算法。

2. 用流程图描述算法

流程图是用一些图框表示各种操作的算法描述方法。用图形描述算法，直观形象，易于理解。ANSI 规定了一些常用的流程图符号（如图 1.1 所示）。

图 1.1 中的连接点（小圆圈）是用于将画在不同地方的流程线连接起来。实际上它们是同一个点，只是一张图上画不下才分开来画。当流程图比较大时，使用连接点，可以避免流程线的交叉或过长，使流程图清晰直观。

1966 年，Bohra 和 Jacopini 提出了以下三种基本结构，用这三种基本结构作为描述一个良好算法的基本单元。

（1）顺序结构。如图 1.2 所示。A 和 B 两个框组成一个顺序结构。表示当程序段 A 执行完后接着执行程序段 B。

（2）选择结构。如图 1.3 所示。当条件 P 成立时执行程序段 A，条件 P 不成立则执行程序段 B。请注意图 1.3 是一个整体，代表一个基本结构。

（3）循环结构。循环结构分为当型和直到型两种。当型循环结构如图 1.4 所示。表示当条件 P 成立时反复执行程序段 A，直到条件 P 不成立为止，跳出循环。直到型循环结构如图 1.5 所示。表示反复执行程序段 A，直到条件 P 成立为止，跳出循环。

运用以上基本结构，可以画出例 1.3 的流程图，如图 1.6 所示。