高等学校“十三五”规划教材

GAODENG XUEXIAO “13·5” GUIHUA JIAOCAI

C语言程序设计

C YUYAN CHENGXU SHEJI

邓达平　谢小云　彭　洁　编著

冶金工业出版社
www.cnmip.com.cn

Metallurgical Industry Press
冶 金 工 业 出 版 社

C语言程序设计

体验更多精彩阅读
尽在冶金工业出版社微信平台

ISBN 978-7-5024-6165-2
9 787502 461652 >

定价59.80元
销售分类建议：计算机

高等学校“十三五”规划教材

C语言程序设计

C YUYAN CHENGXU SHEJI

邓达平　谢小云　彭　洁　编著

北　京
冶　金　工　业　出　版　社
2019

内 容 简 介

C 语言是一门重要的计算机语言，广泛应用于系统软件、应用软件、嵌入式系统软件、物联网设备相关软件等的开发，是高等院校工科类专业的必修基础课，对学生后续相关课程的学习影响较大。

本书基于 ANSI C 标准，突出基础知识的讲解和应用，内容分为三部分：第一部分（第 1 章和第 2 章）主要介绍计算机基础知识、C 语言发展历史、程序设计方法及 C 语言程序的开发环境；第二部分（第 3～13 章）重点介绍 C 语言的语法，并通过大量实例对重要知识进行说明，是本书的主要内容；第三部分（附录）给出了 C 语言程序设计所需的 ASCII 码、库函数等参考资料，便于读者查阅。

本书内容详细，以实例为主线，适合高等院校工科类专业学生学习使用，也可作为 C 语言程序员的参考手册。

图书在版编目(CIP)数据

C 语言程序设计/邓达平，谢小云，彭洁编著. —
北京：冶金工业出版社，2019.6
高等学校“十三五”规划教材
ISBN 978-7-5024-6165-2

Ⅰ.①C… Ⅱ.①邓… ②谢… ③彭… Ⅲ.①C 语言—程序设计—高等学校—教材 Ⅳ.①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 057100 号

出 版 人 谭学余
地 址 北京市东城区嵩祝院北巷 39 号 邮编 100009 电话 (010)64027926
网 址 www.cnmip.com.cn 电子信箱 yjcbs@cnmip.com.cn
责任编辑 纵晓阳 美术编辑 易 帅 版式设计 刘 芬
责任校对 何立兵 责任印制 李玉山
ISBN 978-7-5024-6165-2
冶金工业出版社出版发行；各地新华书店经销；三河市鑫鑫科达彩色印刷包装有限公司印刷
2019 年 6 月第 1 版，2019 年 6 月第 1 次印刷
787mm×1092mm 1/16；21.5 印张；548 千字；334 页
59.80 元
冶金工业出版社 投稿电话 (010)64027932 投稿信箱 tougao@cnmip.com.cn
冶金工业出版社营销中心 电话 (010)64044283 传真 (010)64027893
冶金工业出版社天猫旗舰店 yjgycbs.tmall.com
（本书如有印装质量问题，本社营销中心负责退换）

Preface

前言

众所周知，软件已成为我国新一代信息技术产业重点发展的内容之一，这使得软件研发成为一个非常具有发展前景的技术领域，也是相关专业大学毕业生就业的重要方向。

就本书所介绍的计算机语言——C语言来说，它是一门应用广泛的计算机语言，已成为很多理工类专业学生必须掌握的基础知识。C语言具有功能完善、灵活性强、目标程序效率高、可移植性强等优点，因此获得了众多工程师的喜爱。学习C语言程序设计入门容易，但要达到精通的程度，就需要进行大量的实际应用，并不断总结应用经验。为帮助读者学好C语言及其他计算机语言，作者提出以下几点编程经验，供读者参考。

(1)应当做到“两多”，即多读代码、多写代码

C语言初学者容易陷入对语法的“死记硬背”之中，从而偏离了计算机语言学习必须“以实际编程为重点”的正确思路。正确的学习方法是在掌握初步的语法知识之后，多读代码、多写代码，从而循序渐进地进行程序编写；在编写代码的过程中，发现语法错误或者发现遗忘的语法知识后，再回过头去查看相关内容，这样更容易掌握相关知识。

(2)注重培养逻辑思维能力

学习计算机语言还要注重培养自己的逻辑思维能力，学习如何使用计算机语言去描述需要解决的问题，借助一定的工具进行问题建模，最后用C语言编写代码。

(3)学习算法及数据结构的相关知识

算法解决的是程序处理问题的思路和方法，数据结构解决的是程序中数据的组织方式。学习算法及数据结构的知识对于编写出优质

的程序是非常重要的。此外，即使对于较为简单的问题，所编写的代码也需要不断优化。因此，应当研究如何提升程序的执行效率。

(4)重视程序的分析与设计

有很多初学者面对编程的需求，如编程解答一道题，往往稍加思索就开始动手编程，以致问题考虑不全面，程序无法实现指定功能，甚至出现很多错误。正确的做法是在编写程序之前，首先对问题进行分析，明确需求，然后利用程序流程图等工具进行建模分析，最后按照模型进行代码编写。

(5)按照编程规范养成良好的编程习惯

编程规范是一系列书写代码的原则。与本书配套的《C语言程序设计实验指导与课程设计》一书的第1章中介绍了C语言的相关编程规范，读者可以参考。编程就像写应用文，需要按照既定的规范和格式来书写，这样有利于提高代码的可读性，便于与他人交流，也可以避免一些潜在的错误，提升代码的质量。

(6)掌握好程序调试的方法，实现错误的快速定位

编写C语言程序，很可能会出现问题，甚至是错误。因此，要求程序员必须掌握好程序调试的方法，能够读懂编译器的报错信息，快速地对错误进行定位，从而完成代码的修改。在调试的过程中多思考、多总结，就能较快地掌握常见错误的调试方法。

本书全面介绍ANSI C的语法规则、开发平台的使用方法等内容，并通过大量的实例对C语言程序设计中的重要知识进行说明。书中的例题均使用Visual C++ 6.0进行调试，保证了结果的正确性。

全书共13章，各章的主要内容如下：

第1章：主要介绍程序设计的基本概念、C语言发展历史、算法的基本概念以及程序设计方法。

第2章：主要介绍C语言程序的编写与运行、Visual C++ 6.0的安装与使用方法。

Preface

第3章：主要介绍C语言的标识符、常量、变量、数据类型、表达式、类型转换、赋值运算及逗号运算符等基础知识。

第4章：主要介绍C语言的3种基本结构形式，并介绍scanf函数、printf函数、getchar函数及putchar函数的使用方法。

第5章：主要介绍关系运算符及逻辑运算符，重点介绍多种if语句的格式及使用方法。此外，还介绍条件运算符的使用方法。

第6章：主要介绍3种循环结构的语法规则、使用方法等，并对break、continue语句进行对比介绍，最后介绍嵌套循环的使用方法。

第7章：在介绍数组基本概念的基础上，介绍一维数组、二维数组、字符数组的定义、初始化、引用及使用等内容。

第8章：主要介绍模块化设计的思想、函数的基本概念，重点介绍函数的定义及使用方法，最后对C语言中的变量类型进行介绍。

第9章：主要介绍指针的概念、定义、初始化及引用，重点介绍指针及指针数组的使用方法，最后对动态内存分配的相关操作进行介绍。

第10章：主要介绍结构体及共用体的定义及使用方法，重点介绍结构体变量的使用及结构体指针的使用等内容。

第11章：主要介绍文件的分类、文件类型及文件指针等基本概念，重点介绍文件打开、关闭、读写、定位及出错检测等函数的使用方法。

第12章：在介绍预编译处理流程的基础上，详细介绍宏定义、文件包含及条件编译等预编译处理方法。

第13章：主要介绍位运算的概念及基于字节、字等单位的编程的异同，并对位段的概念、使用方法及使用技巧等进行介绍。

本书由邓达平、谢小云和彭洁编写。其中，第1～2、8～10章及附录部分由邓达平编写，第3～6章由彭洁编写，第7、11～13章由谢小云编写。全书由邓达平统稿。本书在编写过程中得到了江西理工大学应用

Preface

科学学院信息工程系计算机教研室相关教师的大力支持，在此深表感谢！本书内容所涉及的研究获得江西省高等学校教学改革研究课题（项目编号：JXJG-18-36-1）、江西省教育厅科学技术研究项目（项目编号：GJJ181504）的支持，本书的出版获得江西理工大学应用科学学院的资助，在此一并表示感谢！

由于作者水平所限，书中不妥之处，恳请广大读者批评指正。

作　者

2019 年 1 月

Contents
目 录

第1章　程序设计基础

本章主要内容：

① 计算机发展简史。
② 程序设计的相关概念及步骤。
③ 程序设计语言发展简史。
④ C语言发展简史。
⑤ 算法的概念及算法的五个特性。
⑥ 算法的度量。
⑦ 程序流程图的概念及基本框图。
⑧ 三种基本控制结构。

1.1　C语言概述

1.1.1　从算盘到计算机

人类计算工具经历了由简单到复杂、从低级到高级的演化，从最开始的绳结到算筹、算盘、机械计算器等(见图1.1)。它们在不同的历史时期发挥了各自的历史作用，同时也启发了电子计算机的研制和设计思路。它们的计算速度由低速到高速，由几秒一次到每秒上百次，特别是电子计算机的发明，使得计算速度有了质的飞跃。

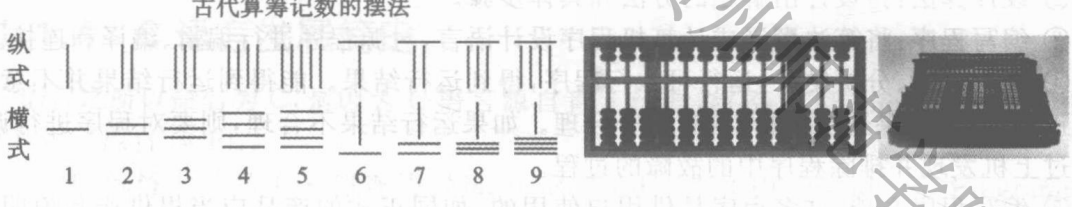

图1.1　算筹、算盘和机械计算器

1642年，法国哲学家、数学家布莱士·帕斯卡(Blaise Pascal，1623—1662)发明了世界上第一台手摇式机械计算器，如图1.2(a)所示。它利用齿轮传动原理制成，能做加减法。

1889年，美国科学家赫尔曼·何乐礼(Herman Hollerith，1860—1929)研制出以电力为基础的电动制表机[见图1.2(b)]，用于存储计算资料。

1930年，美国科学家范内瓦·布什(Vannevar Bush，1890—1974)制造出世界上首台模拟电子计算机，如图1.2(c)所示。

(a)手摇式机械计算器　(b)电动制表机　(c)模拟电子计算机

图1.2　手摇式机械计算器、电动制表机和模拟电子计算机

1946年2月14日，由美国军方定制的世界上第一台电子计算机——电子数字积分计算机(Electronic Numerical Integrator And Calculator，ENIAC)在美国宾夕法尼亚大学问世。ENIAC是美国奥伯丁武器试验场为了满足计算弹道需要而研制的，这台计算机使用了17 840支电子管，尺寸为80ft(约24.38m)×8ft(约2.438m)，重达28t，功耗为170kW，其运算速度为每秒5000次的加法运算，造价约为487 000美元。ENIAC的问世具有划时代的意义，表明了电子计算机时代的到来。

电子计算机在短短的70多年里经历了电子管、晶体管、集成电路和超大规模集成电路四个阶段的发展，体积越来越小，功能越来越强，价格越来越低，应用越来越广泛，目前正朝智能化(第五代)计算机方向发展。

1.1.2 程序设计的相关概念及步骤

程序(Program)：指人们事先准备好的、用来指挥计算机工作的、描述工作步骤的指令序列。

计算机程序(Computer Program)：按照程序设计语言的规则组织起来的一组计算机指令。

程序设计语言(Programming Language)：是计算机能够理解和识别的一种语言体系，它按照特定的规则组织计算机指令，使计算机能够自动进行各种操作处理。

程序设计(Programming)：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。程序设计往往以某种程序设计语言为工具(如C语言)，给出这种语言下的程序。专业的程序设计人员常被称为程序员(Programmer)。

程序设计的步骤包括分析问题、设计算法、编写程序、运行程序、分析结果和编写程序文档等。

① 分析问题：对于接受的任务要进行认真分析，研究所给定的条件，分析最后应达到的目标，找出解决问题的规律，选择解题的方法，解决实际问题。

② 设计算法：指设计出解题的方法和具体步骤。

③ 编写程序：将算法翻译成计算机程序设计语言，对源程序进行编辑、编译和连接。

④ 运行程序、分析结果：运行可执行程序，得到运行结果。能得到运行结果并不意味着程序正确，要对结果进行分析，看它是否合理。如果运行结果不合理，则要对程序进行调试，即通过上机发现并排除程序中的故障的过程。

⑤ 编写程序文档：许多程序是供用户使用的，如同正式的产品应当提供产品说明书一样，正式提供给用户使用的程序，必须向用户提供程序说明书。程序说明书应包括程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据以及使用注意事项等内容。

1.1.3 程序设计语言发展简史

程序设计语言是用于书写计算机程序的语言。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。程序设计语言有三方面的因素，即语法、语义和语用。语法表示程序的结构或形式，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，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。

自20世纪60年代以来，世界上公布的程序设计语言已达上千种，但是只有很小一部分得到了广泛应用。从发展历程来看，程序设计语言大致可以分为四代。

1. 第一代：机器语言

机器语言是由二进制0、1代码指令构成的，不同的CPU具有不同的指令系统。机器语

言程序难编写、难修改、难维护，需要用户直接对存储空间进行分配，编程效率极低。这种语言已经被淘汰了。

2. 第二代：汇编语言

汇编语言指令是机器指令的符号化，与机器指令存在着直接的对应关系，因此汇编语言同样存在着难学难用、容易出错、维护困难等缺点。但是汇编语言也有自己的优点，即可直接访问系统接口，由汇编语言程序翻译成机器语言程序的效率高。从软件工程的角度看，只有当高级语言不能满足设计要求，或不具备支持某种特定功能的技术性能（如特殊的输入/输出）时，汇编语言才被使用。

3. 第三代：高级语言

高级语言是面向用户的、基本上独立于计算机种类和结构的语言。其最大的优点是形式接近于算术语言和自然语言，概念接近于人们通常使用的概念。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言指令。因此，高级语言易学易用，通用性强，应用广泛。高级语言种类繁多，从描述客观系统来看，可以分为面向过程语言和面向对象语言。面向过程语言是以“数据结构＋算法”程序设计范式构成的程序设计语言，比较有影响力的有FORTRAN、COBOL、BASIC、ALGOL、PASCAL、C、Ada 等；面向对象语言是以“对象＋消息”程序设计范式构成的程序设计语言，比较常见的面向对象语言有 Delphi、Visual Basic、Java、C++等。

4. 第四代：非过程化语言

第四代的非过程化语言，编码时只需说明“做什么”，而不需描述算法细节。数据库查询和应用程序生成器是它的两个典型应用。用户可以用数据库查询语言（SQL）对数据库中的信息进行复杂的操作。它具有缩短应用开发过程、降低维护代价、最大限度地减少调试过程中出现的问题，以及对用户友好等优点。

1.1.4　C 语言发展简史

C 语言之所以命名为 C，是因为 C 语言源自肯·汤普森（Ken Thompson）发明的 B 语言，而 B 语言源自 BCPL。

1967 年，剑桥大学马丁·理查德（Martin Richards）对 CPL（Combined Programming Language）进行了简化，于是产生了 BCPL（Basic Combined Programming Language）。

1970 年，美国贝尔实验室的肯·汤普森以 BCPL 为基础，设计出简单且接近硬件的 B 语言（取 BCPL 的首字母）。

1972 年，美国贝尔实验室的丹尼斯·M. 里奇（Dennis M. Ritchie）在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。

1977 年，丹尼斯·M. 里奇发表了不依赖于具体机器系统的 C 语言编译文本——《可移植的 C 语言编译程序》。

1978 年，布莱恩·W. 克尼汉（Brian W. Kernighan）和丹尼斯·M. 里奇出版了《C 程序设计语言》（*The C Programming Language*），从而使 C 语言成为目前世界上流行广泛的高级程序设计语言之一。

1982 年，很多有识之士和美国国家标准协会（ANSI）为了使 C 语言健康地发展下去，决定成立 C 标准委员会，建立 C 语言的标准。C 标准委员会成员包括硬件厂商、编译器及其他软件工具生产商、软件设计师、顾问、学术界人士、C 语言作者和应用程序员。1989 年，ANSI 发布了第一个完整的 C 语言标准——ANSI X3.159—1989，简称 C89，但是人们仍然习惯称其为 ANSI C。C89

在1990年被国际标准化组织(International Organization for Standardization,ISO)所采纳,ISO官方给予的名称为ISO/IEC 9899,所以ISO/IEC 9899:1990也通常被简称为C90。1999年,在做了一些必要的修正和完善后,ISO发布了新的C语言标准,命名为ISO/IEC 9899:1999,简称C99。2011年12月8日,ISO又正式发布了新的标准,称为ISO/IEC 9899:2011,简称C11。

1.1.5 C语言的特点

C语言是一种计算机程序设计语言,它既具有高级语言的特点,又具有汇编语言的特点,因而有很多人将其视为一种中间语言。

1.C语言的主要优点

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,程序书写形式自由,区分大小写。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了起来。

② 运算符丰富。C语言的运算符包含的范围很广泛,共有34种运算符。C语言把括号、赋值和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他高级语言中难以实现的运算。

③ 数据类型丰富。C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型和共用体类型等,能用来实现各种复杂的数据结构的运算。此外,它还引入了指针概念,使程序效率更高。

④ 表达方式灵活实用。C语言提供了多种运算符和表达式值的方法,对问题的表达可通过多种途径获得,其程序设计更主动、灵活。其语法限制不太严格,程序设计自由度大,如对整型量与字符型数据及逻辑型数据可以通用等。

⑤ 允许直接访问物理地址,对硬件进行操作。C语言允许直接访问物理地址,可以直接对硬件进行操作,故它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(bit)、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,可用来编写系统软件。

⑥ 生成目标代码质量高,程序执行效率高。C语言描述问题比汇编语言迅速,工作量小,可读性好,易于调试、修改和移植,而代码质量与汇编语言相当。C语言一般只比汇编程序生成的目标代码效率低10%~20%。

⑦ 可移植性好。C语言在不同机器上的C编译程序,86%的代码是公共的,所以C语言的编译程序便于移植。在一个环境上用C语言编写的程序,不改动或稍加改动,就可移植到另一个完全不同的环境中运行。

⑧ 表达力强。C语言既可用来编写系统软件,又可用来开发应用软件,已成为一种通用程序设计语言。另外,C语言具有强大的图形功能,支持多种显示器和驱动器,且计算功能和逻辑判断功能强大。

2.C语言的主要缺点

① C语言缺少面向对象编程的特性(具体表现在其数据的封装性上),这一点使得C语言在数据的安全性上有很大缺陷,这也是C和C++的一大区别。

② 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,且不检查数组下标越界问题。从应用的角度看,C语言比其他高级语言更难掌握。也就是说,使用C语言的人,要对程序设计更熟练一些。

③ C程序的错误更隐蔽。C语言的灵活性使得用它编写程序时更容易出错,而且C语言的编译器不检查这样的错误,因此要求程序员予以重视。与汇编语言类似,这些逻辑错误

需要程序运行时才能发现。比如将比较的“==”写成赋值“=”，这样的逻辑错误就不易被发现，要找出来往往十分费时。

④ C程序有时会难以理解。C语言语法成分相对简单，是一种小型语言。但是，它的数据类型多，运算符丰富且结合性多样，理解起来有一定难度。

1.2 算法

1.2.1 算法的概念

算法(Algorithm)是对解题方案准确而完整的描述，是一系列解决问题的清晰指令。算法代表用系统的方法描述解决问题的策略机制。也就是说，它能够对一定规范的输入，在有限时间内获得所要求的输出。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。瑞士著名的计算机科学家尼古拉斯·沃斯(Nicklaus Wirth)提出了一个著名的公式“算法＋数据结构＝程序”，这个公式展示了程序的本质，正是凭借这个公式使得他获得了图灵奖。这个公式对计算机科学的影响程度足以媲美物理学中爱因斯坦的能量公式“$E=MC^2$”。

1.2.2 算法的特征

一个算法应该具有以下5个特征：

(1)有穷性

一个算法必须总是(对任何合法的输入值)在执行有穷步之后结束，且每一步都可在有穷时间内完成。这里的有穷不是数学上的有穷，而是指在实际上是合理的、可接受的。

(2)确定性

算法中每一条指令必须有确切的含义，读者理解时不会产生二义性。并且，在任何条件下，算法只有唯一的一条执行路径，即对于相同的输入只能得出相同的输出。

(3)可行性

一个算法是可行的，即算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。

(4)输入

一个算法可以没有输入，也可以有多个输入，这些输入取自某个特定的对象的集合。

(5)输出

一个算法至少要有一个输出，也可以有多个输出，这些输出是同输入有着某些特定关系的量。

1.2.3 算法的度量

对于同一个问题，有时可用许多不同的方法来解决，而这些不同方法的时间效率和成本可能相差极大。下面介绍两个生活中数学计算的例子。

【例1.1】 求1＋2＋3＋…＋100的值。

解 解决方案一：准备好一支笔和一张白纸，开始计算1＋2＝3，再算3＋3＝6，再算6＋4＝10，再算10＋5＝15……最后算4950＋100＝5050。显然，这一共需要进行99次运算。

解决方案二：直接利用公式 $s=100*(1+100)/2=5050$，只需进行一次运算。

【例 1.2】 猜价格游戏。主持人展示一件物品，并宣称其价格为 1～100 元，请你猜价格。如果你猜的价格刚好等于物品的实际价格，则游戏结束；如果你猜的价格比实际价格低，则主持人会告诉你“低了”；如果你猜的价格比实际价格高，则主持人会告诉你“高了”。

解 解决方案一：瞎猜法。随意选取 1～100 中的一个数字作为价格，显然这种方法一方面很难猜对，另一方面也很可能会重复自己已经报过的价格。

解决方案二：顺序法。从 1 元开始报价格，依次报 1 元、2 元、3 元、4 元…，主持人都反馈“低了”，一直报到物品的实际价格。这种方法一定能猜出价格，但是平均需要猜 50 次。

解决方案三：折半法。每次都先根据物品的一个价格范围，计算出该范围的中间价（折半数，即最低数和最高数的中位数），再根据主持人的反馈调整这个价格范围并重新计算。假设该物品的实际价格为 66 元，则具体方法如下：

① 第 1 次时，物品的价格范围为 1～100 元，则计算出中间价格＝(1＋100)/2＝50 元（取整，不四舍五入，下同），主持人反馈“低了”。显然，其价格必定为 51～100 元（因为 50 元已经猜过了，可以剔除，下同）。可以看出，此时所需要猜的范围为原范围的一半。

② 第 2 次，调整当前范围为 51～100 元，计算出中间价格＝(51＋100)/2＝75 元，主持人反馈“高了”。显然，该价格必定为 51～74 元。此时所需要猜的范围又缩小为上一次的一半。

③ 第 3 次，调整当前范围为 51～74 元，计算出中间价格＝(51＋74)/2＝62 元，主持人反馈“低了”。显然，该价格必定为 63～74 元。

④ 第 4 次，调整当前范围为 63～74 元，计算出中间价格＝(63＋74)/2＝68 元，主持人反馈“高了”。显然，该价格必定为 63～67 元。

⑤ 第 5 次，调整当前范围为 63～67 元，计算出中间价格＝(63＋67)/2＝65 元，主持人反馈“低了”。显然，该价格必定为 66～67 元。

⑥ 第 6 次，调整当前范围为 66～67 元，计算出中间价格＝(66＋67)/2＝66，主持人反馈“猜中”，游戏结束。

显然，使用方案一可能需要猜 100 多次，甚至永远都猜不出来；方案二平均需要猜50 次；而方案三最多只需要猜 $\log_2 100+1=7$ 次。其中，$\log_2 100$ 不四舍五入取整的值为 6。如果这个物品的价格范围不是 1～100 元，而是 1～10 000 元，则方案二平均需要 5000 次，方案三最多只需要 $\log_2 10\,000+1=14$ 次。显然，数字范围越大，方案三的效率优势越明显。

从上面两个实例可以看出，同一个问题不同解决方案，其效率可能会相差很大。同样，对于用计算机编写的程序来解决问题，不同方法（算法）的效率也不同。要度量算法的效率，可以从时间和空间上进行。从时间上度量算法的效率，需要使用“时间复杂度”；从空间上度量算法的效率，需要使用“空间复杂度”。

(1)时间复杂度

时间复杂度是指执行算法所需要的计算工作量。一般情况下，算法中基本操作重复执行的次数是问题规模 n 的某个函数，用 $T(n)$ 表示，若有某个辅助函数 $f(n)$，使得当 n 趋近于无穷大时，$T(n)/f(n)$ 的极限值为不等于零的常数，则称 $f(n)$ 是 $T(n)$ 的同数量级函数，记作 $T(n)=O(f(n))$。$O(f(n))$ 被称为算法的渐进时间复杂度，简称时间复杂度。

显然，随着规模 n 的增大，算法执行时间的增长率和 $f(n)$ 的增长率成正比，所以 $f(n)$ 越小，算法的时间复杂度越低，算法的效率越高。

度量算法的时间复杂度常用的 $f(n)$ 函数有 $O(1)$、$O(\log_2 n)$、$O(n)$、$O(n\log_2 n)$、$O(n^2)$、$O(n^3)$、$O(2^n)$ 和 $O(n!)$。例如在例 1.1 中，方案一的计算量显然是与从 1 加到 n