普通高等教育“十二五”规划教材
全国普通高等学校优秀教材
高等学校公共课*计算机*规划教材

C语言程序设计案例教程

■ 王端理　主编

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY [http://www.phei.com.cn]

普通高等教育“十二五”规划教材
全国普通高等学校优秀教材
高等学校公共课计算机规划教材

C 语言程序设计案例教程

王端理 主编

付苏嘉 冉桂萍 杨 焱 编

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是全国普通高等学校优秀教材，从实用性出发，针对初学者较全面地介绍C语言的语法规则、编程思路、编程方法和程序设计具体应用方面的技能。全书共8章，主要内容包括：算法与C语言程序设计、C语言概述、顺序结构程序设计、选择结构程序设计、循环结构程序设计、简单构造数据类型、模块化程序设计和复杂构造数据类型，以及C语言库函数，每章后附实验指导和上机实战。本书提供配套电子课件、程序代码和习题参考答案等。

本书可作为高等学校本科理工类专业C语言程序设计课程的教材，也可作为高职高专及C语言自学者的教材和参考书。

图书在版编目（CIP）数据

C语言程序设计案例教程 / 王端理主编. —北京：电子工业出版社，2014.1
高等学校公共课计算机规划教材
ISBN 978-7-121-22084-5

I. ①C… II. ①王… 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（2013）第291523号

策划编辑：王羽佳
责任编辑：王羽佳　　文字编辑：王晓庆
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　　邮编：100036
开　　本：787×1092　1/16　印张：10　　字数：237千字
印　　次：2014年1月第1次印刷
印　　数：5000册　　定价：27.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

随着我国计算机技术的迅猛发展，社会对具备计算机基本能力的人才的需求急剧增加，具备计算机基本知识与能力已成为21世纪人才的基本素质之一。

未来社会利用计算机解决问题已经成为了一种主流。要想做到这一点，人类必须首先将现实世界的事物抽象成计算机能够识别并加工的数据，接着抽象出加工的流程，然后用计算机语言描述加工流程，最后提交给计算机执行。这就需要我们具备相应的计算思维能力，因此计算思维在人类未来的工作和生活中极为重要。程序设计是培养计算思维能力的一个很好的平台。

为了适应高等学校正在开展的以计算思维能力培养为重点的大学计算机教育的教学改革，及时反映计算机基础教学的研究成果，积极探索适应21世纪人才培养的教学模式，我们编写了这本C语言程序设计案例教程。

C语言是目前世界上使用最广泛的计算机程序设计语言。由于其强大的功能，特别是其高级语言的表示风格和低级语言的特性，使得利用C语言在编写应用程序和系统软件方面都得天独厚，成为目前最实用且功能强大的编程语言，因而被大多数高等学校采用作为理工科的公共必修课程。然而其精细的语言规则和强悍的计算思维体系成为初学者学习C语言的两道屏障。对于C语言的初学者而言，必须通过大量的程序实例，由浅入深地逐步体会C语言的语法规则和计算思维，才能达到具有使用C语言编写程序的基本能力。

本书采用“知识讲解—程序案例—实验跟进—引发思考”的形式，将知识点融入程序案例，以程序案例带动知识点的学习，并在关键点上通过“想一想，做一做”的方式引发思考，来提高读者对该课程的学习兴趣，同时配以一定的实验，四者相辅相成。在具体程序案例的讲解中，通过阅读问题、展开分析、给出解题思路并结合C语言的语法规则，让读者理解并掌握C语言程序设计思想的具体实现过程，通过实验中的实验目的和具体要求，将问题由易到难逐步进行编程，从而掌握C语言。

本书是全国普通高等学校优秀教材，从实用性出发，针对初学者较全面地介绍C语言的语法规则、编程思路、编程方法和程序设计具体应用方面的技能。全书共8章，主要内容包括：算法与C语言程序设计、C语言概述、顺序结构程序设计、选择结构程序设计、循环结构程序设计、简单构造数据类型、模块化程序设计和复杂构造数据类型，以及C语言库函数。

通过学习本书，你可以：

- 了解结构化程序设计的基本原理和技术；
- 在实验引导下逐步认识VC++的编程环境；
- 在简单易懂的案例驱动下掌握程序设计的思维和方法；
- 自己动手设计一个简单的程序；
- 结合实际问题做好分析——进行简单的应用程序的开发。

本书可作为高等学校本科理工类专业 C 语言程序设计课程的教材，也可作为高职高专及 C 语言自学者的教材和参考书。

为适应教学模式、教学方法和手段的改革，本书提供配套电子课件、程序代码和习题参考答案等教学资源，请登录华信教育资源网（http://www.hxedu.com.cn）注册下载。

本书第 1～4 章由王端理编写，第 5、8 章由付苏嘉编写，第 6 章和第 2 章的 2.2 节由杨焱编写，第 7 章和附录 A 由冉桂萍编写。本书各章的实验指导由王端理编写，全书由王端理统稿。

本书在编写过程中一直坚持将理论与实践紧密结合的原则，但由于成书时间较为仓促，加之编写者水平有限，书中难免出现错误和不妥之处，敬请读者批评指正。

作　者

目　录

第1章　算法与C语言程序设计

在现实世界做任何事情都有相应的方法和步骤，这种方法和步骤就称为算法。计算机所做的工作就是按照人们事先指定的步骤按部就班地执行相应的操作，从而完成特定的任务。因此，要让计算机为我们做事，就必须事先设计出一系列操作步骤，并用计算机语言写成程序，这就是程序设计，而解决问题的方法和步骤，就称为算法。它是程序设计的关键，所以在学习C语言程序设计之前，必须掌握算法的基本知识。

1.1　算　　法

1.1.1　基本概念

1．算法的定义

所谓算法，是指为解决问题而采用的方法和步骤。算法最早来源于算术运算，如算式(3+4)×5−(7−3)÷2的运算就是数字在运算符的操作下，按照规则进行的数值变换，而这个“规则”就是算法。随着人类社会问题的不断增加和复杂化，要想很好地解决问题，均需要按照一定规则逐步展开。比如去食堂吃饭、去阅览室看书、找工作等都有相应的操作步骤，因此算法一词也从数值计算领域延伸到了非数值计算领域。

随着信息社会的发展，计算机已成为人们日常生活和工作中不可缺少的工具。听音乐、看电影、玩游戏、打字、画卡通画、处理数据，计算机几乎渗透到了人们生活的所有领域，那计算机是怎样工作的呢？要想弄清楚这个问题，算法的学习是一个开始。在计算机中，利用计算机解决问题的方法和步骤称为计算机的算法。如求两个正整数的最大公约数、线性方程的求解方法、文字处理方法、信息查找方法等都是算法。然而不是所有问题计算机都能够解决，根据图灵理论：只要能够分解为有限步骤，并且每一步骤都可以转化为计算机可以执行的程序指令的问题，才是计算机可以解决的问题。这里面包含两层含义，一是算法的步骤必须是有限的，二是算法最终可以转化为计算机所执行的程序。因此算法设计是程序设计的基础，算法研究成为了计算机科学的核心课题之一。

综上所述，算法是求解问题步骤的有序集合。

2．算法的基本特征

作为一个算法，一般应具有以下5个特征。

（1）可行性

任何一个算法的执行过程往往都要受到计算工具和相应环境的限制。比如“骑自行车

比赛”在平原、丘陵的方案就会有所不同，而到青藏高原骑着自行车爬喜马拉雅山就只能是梦想了。因此，在设计一个算法时，必须要考虑它的可行性，否则就得不到满意的结果，甚至得不到结果。

（2）确定性

算法中的每一步最终都要落到实处，因此其每一步都必须有明确的定义，决不允许有含糊其辞和模棱两可的解释。

（3）有穷性

算法必须在有限、合理的时间内做完。即算法的执行步骤必须是有限的，其执行时间必须在合理的范围内，不能没完没了地拖几千几万年，这就失去了算法的使用价值。

（4）零到多个输入

算法的执行总是会接触到相应的数据对象，而这些数据对象的初始状态可能需要动态获取，这就需要有相应的输入才能保证算法执行有起点。当然，如果数据对象的初始状态是静态的，那就可以在算法操作步骤中直接给出，而无须再输入。

（5）至少一个输出

执行算法的最终目的是得到相应的结果。如果一个算法执行后没有任何结果，这样的算法就失去了实用价值。

3. 算法的组成要素

一个算法通常由两种基本要素组成：一是对数据对象的运算和操作；二是算法的控制结构。

（1）对数据对象的运算和操作

每个算法实际上是根据题意并结合环境选择合适的操作所组成的一组指令序列，因此，计算机算法就是由计算机处理的操作所组成的指令序列。而指令是计算机可以执行的基本操作。

操作离不开运算。在一般的计算机系统中，基本的操作运算有以下 4 类。

① 算术运算：主要包括加、减、乘、除等运算。

② 逻辑运算：主要包括“与”、“或”、“非”等运算。

③ 关系运算：主要包括“大于”、“小于”、“等于”、“不等于”等运算。

④ 数据传输：主要包括赋值、输入、输出等操作。

在编制计算机的算法时通常要考虑很多与方法和分析无关的细节问题（如语法规则），因此在设计算法的开始，通常并不直接利用计算机来描述算法，而是用别的描述工具（如流程图、专门的算法描述语言，甚至用自然语言）来描述算法。但不管用哪种工具来描述算法，算法的设计一般都要从上述 4 类基本操作运算中考虑，根据题意从这些基本操作运算中选择合适的操作组成解题的操作序列。计算机算法的主要特征着重于算法的动态执行，它区别于传统的着重于静态描述或按演绎方式求解问题的过程。传统的演绎数学是以公理系统为基础的，问题的求解过程是通过有限次推演来完成的，每次推演都将进一步描述问题，如此不断推演，直到直接将结果描述出来为止。而计算机算法则是使用一些最基本的

操作，通过对已知条件一步一步地加工和变换，从而实现解题目标。这两种方法的解题思路是不同的。

（2）算法的控制结构

一个算法的功能不仅取决于所选用的操作，而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。

算法的控制结构给出了算法的基本框架，它不仅决定了算法中各操作的执行顺序，而且也直接反映了算法的设计是否符合结构化原则。描述算法的工具通常有自然语言、传统流程图、N-S 结构化流程图、算法描述语言等，而流程图特别是传统流程图是初学者最喜欢的算法表示。一个算法一般都可以用顺序、选择、循环这 3 种基本控制结构组合而成。我们通过如图 1.1～图 1.3 所示的传统流程图，直观地来了解这 3 种结构及图框。

顺序结构：

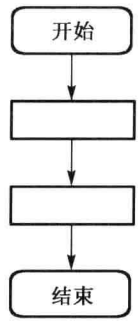

图 1.1　顺序结构

选择结构：

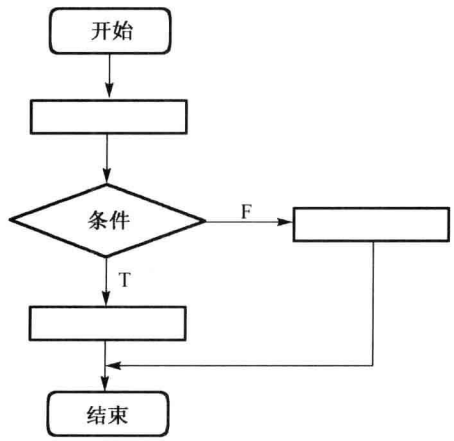

图 1.2　选择结构

循环结构：

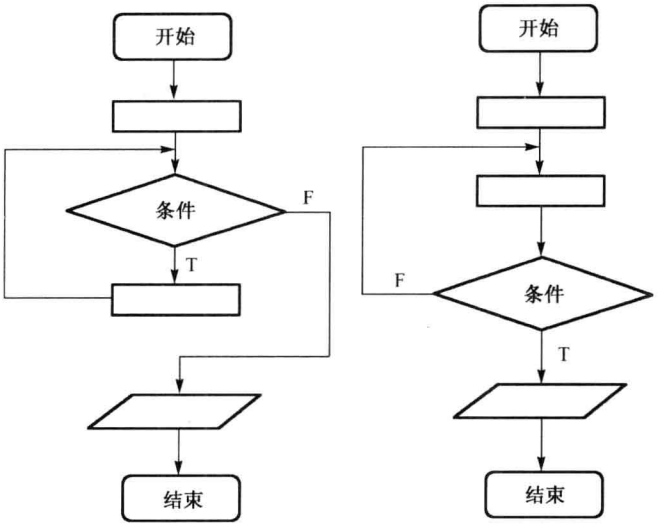

图 1.3 当型循环和直到型循环

流程图常用图框与符号如图 1.4 所示。

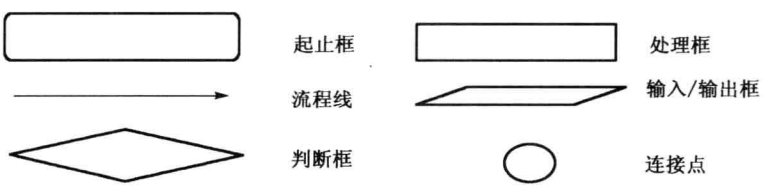

图 1.4 流程图常用图框与符号

4．算法设计基本方法

计算机解题的过程实际上是在实施某种算法，这种算法称为计算机算法。计算机算法不同于人工处理的方法。

本节介绍工程计算上常用的几种算法设计基本方法，在实际应用时，各种方法之间往往存在着一定的联系。

（1）穷举法

穷举法是基于计算机特点而进行解题的思维方法。其基本思想是：根据提出的问题列举所有可能的情况，然后通过一一验证是否符合整个问题的求解要求，而得到问题的解。因此，穷举法常用于解决“是否存在”或“有多少种可能”等类型的问题，如求解不定方程的问题。

穷举法的特点是算法简单，但运行时所花费的时间量大。有些问题所列举出来的情况数目会大得惊人，即使用高速的电子计算机运行，其等待运行结果的时间也将使人无法忍受。因此，在用穷举法解决问题时，应尽可能将明显的不符合条件的情况排除在外，以尽快取得问题的解。通常，在使用穷举法时，只要对实际问题进行详细的分析，将与问题有

关的知识条理化、完备化、系统化，从中找出规律；或将所有可能的情况进行分类，引出一些有用的信息，是可以大大减少列举量的。

穷举法是计算机应用领域中使用极为广泛的方法。许多实际问题，若采用人工列举是不可想象的，但由于计算机的运算速度快，可以很方便地进行大量列举。穷举法虽然是一种比较笨拙而原始的方法，且其运算量比较大，但在有些实际问题中（如寻找路径、查找、搜索等问题），局部使用穷举法却是很有效的，因此，穷举法是计算机算法中的一个基础算法。

（2）归纳法

归纳法是从个别性知识引出一般性知识的推理方法。其基本思想是：通过列举少量具有代表性的信息，经过分析，最后找出一般性的结论。显然，归纳法要比穷举法更能反映问题的本质，并且可以解决穷举法无法解决的问题。但是，根据几个特殊情况总结归纳出一般的关系，并不是一件容易的事情，尤其是要归纳出一个数学模型更为困难。

归纳是一种抽象，即从特殊现象中找出一般关系。但由于在归纳的过程中不可能对所有的情况进行列举，而通过精心观察得到的猜测常常会是错的，因此，最后由归纳得到的还只是一种猜测，还需要对这种猜测加以必要的证明才能得到结论。

（3）递推法

所谓递推，是指从已知的初始条件出发，逐次推出所要求的各序列结果和最后结果。其中，初始条件或问题本身已经给定，或通过对问题的分析与化简而确定。递推本质上也属于归纳法，工程上许多递推关系式实际上是通过对实际问题的分析与归纳而得到的，因此，递推关系式往往是归纳的结果。

递推法在数值计算中是极为常见的。其思想是把一个复杂的庞大的计算过程转化为简单过程的多次重复。它是计算机中的一种常用算法。该算法利用了计算机速度快和不知疲倦的机器特点。

（4）递归法

人们在解决一些复杂问题时，为了降低问题的复杂程度（如问题的规模等），一般总是将问题逐层分解，最后归结为一些规模较小的同类简单的问题。这种将问题逐层分解的过程，实际上并没有对问题进行求解，而只是当解决了最后那些同类简单的问题后，再沿着原来分解的逆过程逐步进行综合，这就是递归的基本思想。在工程实际中，有许多问题就是用递归来定义的，数学中的许多函数也是用递归来定义的。递归在计算机程序设计中占据很重要的地位。

递归分为直接递归与间接递归两种。如果一个函数显式地调用自己，则称为直接递归；如果一个函数调用另一个函数，而另一个函数又调用该函数，则称为间接递归。

有些实际问题可以归纳为递归法。但递推与递归的实现方法是大不一样的。递推是从初始条件出发，逐次推出所需求的结果；而递归则是从算法本身到达递归边界的。通常，递归法要比递推法清晰易读，其结构比较简练。特别是在许多复杂的问题中，很难找到从初始条件推出所需结果的全过程，此时，使用递归法要比递推法容易得多。但递归法开销较大、执行效率较低。

（5）回溯法

回溯法也称为试探法，它是一种系统地搜索问题的解的方法。其基本思想是：从一条

路往前走，能进则进，不能进则退回来，换一条路再试。它是工程中既不能用归纳法，又不能用递推法、递归法，也不能进行无限列举的一类问题所采用的一种方法。对于这类问题，一种有效的方法就是“试”。通过对问题的分析，找出一个解决问题的线索，然后沿着这个线索逐步试探，对于每一步，若试探成功，就得到问题的解，若试探失败，就逐步回退，换别的路线再进行试探。回溯法在处理复杂数据结构方面有着广泛的应用。

想一想 做一做：

1. 将5个苹果放入x个篮子里，发现必有一个篮子的苹果数不少于3个，用穷举法求篮子数最多为几个？

2. 有一位师傅想考考他的两个徒弟，看谁更聪明一些。他给每人一筐花生去剥皮，看看每一粒花生是否都有粉衣包着，看谁先给出答案。大徒弟费了很大的力气将花生全部剥完了；二徒弟只捡了几个饱满的、几个干瘪的、几个熟透的、几个没熟的、几个三仁的、几个两仁的和几个一仁的，从总共不过一把花生中可以得出结论，显然二徒弟比大徒弟聪明，那么二徒弟用的是什么方法？大徒弟用的是什么方法？

1.1.2 算法表示案例

【案例1.1】 互换变量x和变量y的值。

案例分析：

在计算机中，一个变量在内存中均对应相应的存储单元。而存储器的性质是：一个存储单元一次只能存储一个值，当新值进入时，则原值就被覆盖。因此不能直接将 x 的值赋给y，或直接将y的值赋给x，而应借助一个中间变量z过渡。

具体算法如图1.5所示。

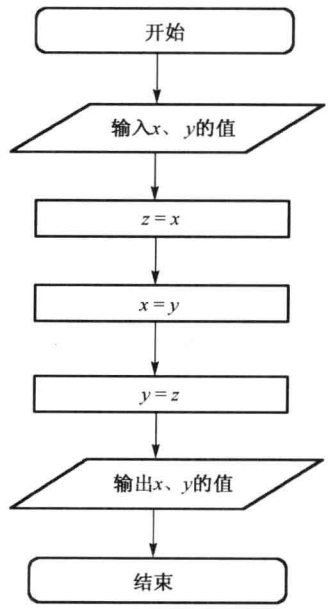

图1.5 实现两变量值互换的流程图

想一想 做一做：

1. 若案例 1.1 缺少第 1 步：输入 x、y 的值，算法是否会出问题？若出了问题，问题出在哪里？

2. 变量 z 的作用是什么？

3. 如果 x、y 都是数值型数据，是否可以不用 z？如果可以，该怎么做（请尝试利用+、-运算实现交换）？

【案例 1.2】 求 $ax+b=0$ 的解。

案例分析：

对于方程 $ax+b=0$ 来讲，应该分情况讨论方程的解。

对一次项系数 a 和常数项 b 的取值情况进行分类，分类如下：

（1）当 $a\neq 0$ 时，方程有唯一的实数解 $-\dfrac{b}{a}$；

（2）当 $a=0$，$b=0$ 时，全体实数都是方程的解；

（3）当 $a=0$，$b\neq 0$ 时，方程无解。

具体算法如图 1.6 所示。

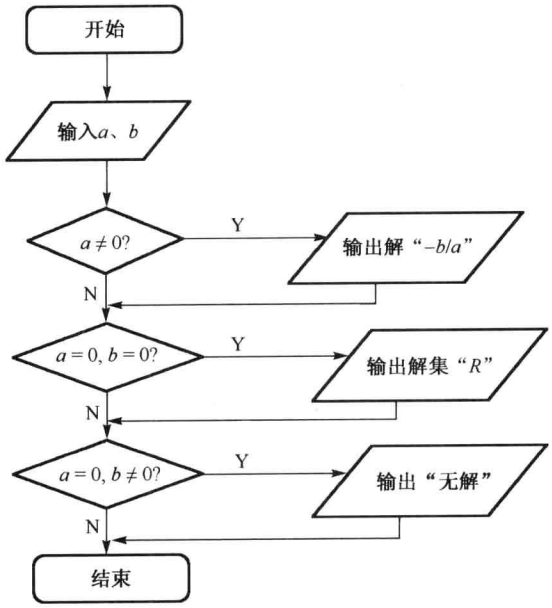

图 1.6　求 $ax+b=0$ 解的流程图

想一想 做一做：

1. 绘制景点买票流程图。注意：儿童老人免票、学生半票、成人全票。

2. 绘制 "红灯停绿灯行" 场景对应的流程图。

【案例 1.3】 求 1+2+…+100 的值。

案例分析：

通常，按照下列过程计算 1+2+…+100 的值。

第 1 步　0+1=1

第 2 步　1+2=3

第 3 步　3+3=6

第 4 步　6+4=10

……

第 100 步　4 950+100=5 050

显然，这个过程中包含重复操作的步骤，可以用循环结构表示上述计算过程，可以发现每一步都可以表示为：第(i−1)步的结果 ＋i＝ 第 i 步的结果。

为了方便、有效地表示上述过程，用一个累加变量 S 来表示第 1 步的计算结果，即把 S+i 的结果仍记为 S，从而把第 i 步表示为 S=S+i，其中 S 的初始值为 0，i 依次取 1, 2, …, 100，由于 i 同时记录了循环的次数，所以也称为计数变量。

具体算法如图 1.7 所示。

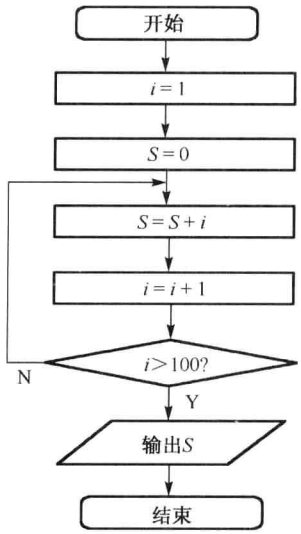

图 1.7　求 1+2+…+100 的流程图

想一想 做一做：

1. 将案例 1.3 改为求 1+3+5+7+…+99，算法该如何？
2. 将案例 1.3 改为求 2+4+6+8+…+100，算法该如何？
3. 将案例 1.3 改为求 1+1/2+1/3+1/4+…+1/100，算法该如何？
4. 将案例 1.3 改为求 1−1/2+1/3−1/4+…−1/100，算法该如何？

【案例 1.4】 将十进制整数 a 转化成二进制数。

案例分析：

将十进制整数转化成二进制数的方法是：除 2 求余反向取。

具体算法如图 1.8 所示。

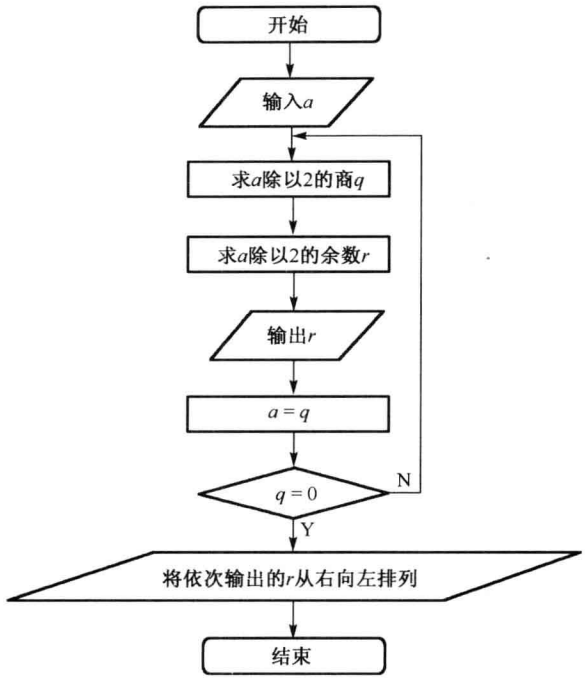

图 1.8　将十进制整数 a 转化成二进制数的流程图

1.2　C 语言程序设计

随着计算机应用领域的不断深入和发展，利用计算机解决的问题也越来越多、越来越复杂。要想让计算机解决一个问题，首先需要把解决问题的方法和步骤设计出来，并且将解决问题所需要的数据合理组织，最后用计算机的语言表示出来，所以有人将程序描述为：程序=算法+数据结构+语言，而编写程序的过程就称为程序设计。由此可见，一个程序设计离不开对问题的分析、程序设计的基本方法和程序设计语言。

1.2.1　分析问题

为了能编写出解决问题的程序，首先应该分析问题，然后设计算法，组织数据并用高级语言编写程序指令。分析问题是第一步，也是最重要的步骤，这一步需要做以下几方面的工作。

1．明确问题的性质

计算机能够解决的问题不外乎两类：一类是数值计算问题，一类是非数值计算问题。不管是哪类问题，其所用的方法、工具以及输入/输出的形式都会有所不同，因此明确问题的性质是分析问题的基础。

2．了解解决问题的必要条件

这些必要条件包括：程序是否需要与用户建立联系，程序是否要处理数据，程序是否有输

出，需要什么样的结果。如果程序要对数据进行操作，那么还必须知道数据是什么，以及这些数据代表什么。如果程序产生输出结果，还应该知道怎样产生结果，以及以何种形式来输出结果。

3．合理分解问题

如果问题比较复杂，应该把复杂问题分解为若干子问题，每个子问题只完成一项简单的功能，并且每个子问题都重复步骤1和步骤2。

1.2.2 C语言程序设计的基本方法

C语言程序设计的基本方法就是结构化程序设计法。结构化程序设计法也称为自上而下的设计、逐步细化和模块化编程，即把问题分解为若干子问题的方法。在结构化程序设计中，问题被分解为若干较小的问题，然后把所有子问题的解决方法结合在一起来解决整个问题。执行结构化程度设计的过程称为结构化编程。结构化程序设计要求把程序的结构限制为顺序、选择和循环3种基本控制结构，以便提高程序的可读性。采用这种结构开发出来的程序具有清晰的结构层次，易于理解并便于修改调试。

结构化程序设计的模块化编程是指将一个大问题按照人们能理解的大小进行分解。如果分解的问题较小，则较为容易实现。在进行模块化编程时，要重点考虑以下两个问题。

1．按功能划分模块

按功能划分模块的基本原则是：按照人们解决复杂问题的普遍规律，使每个模块都易于理解，同时各模块的功能尽可能单一，各模块之间的联系尽量少，从而保证各模块的可读性和可理解性。

2．按层次组织模块

按层次组织模块是划分模块的最好形式之一。在按层次组织模块时，一般上层模块只指出“做什么”，具体“怎么做”由下层模块精确描述。在图1.9所示的层次结构中，上层模块给出任务，最后一层模块才精确描述“怎么做”。

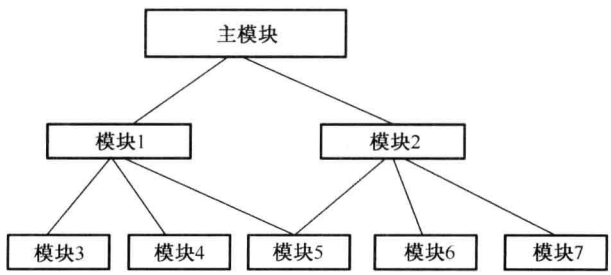

图1.9 按层次组织的模块划分图

1.2.3 C语言程序的构成和基本格式

1．一个简单的C语言程序案例

【案例1.5】 求半径为2的圆的面积。