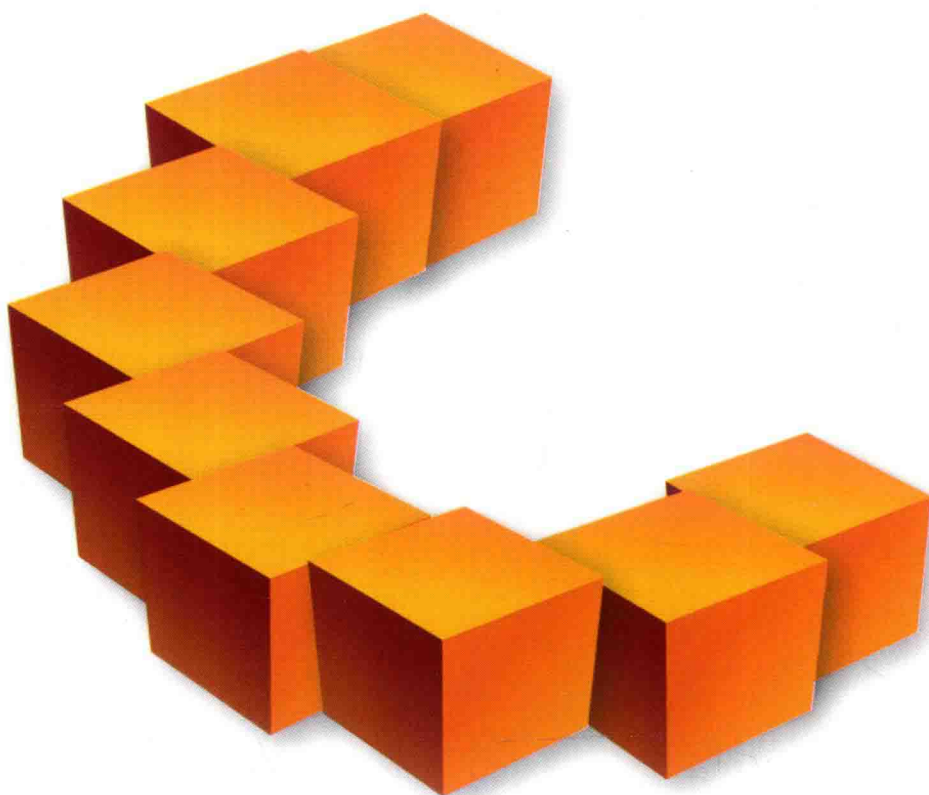




普通高等教育“十三五”规划教材



C语言 程序设计教程

韩雪娜 金一宁 主编

 科学出版社

普通高等教育“十三五”规划教材

C 语言程序设计教程

韩雪娜 金一宁 主编

李俊玲 孔庆彦 康 婧 副主编

哈尔滨商业大学教学改革与教学研究项目
“互联网+现代教育”财经类高校计算机基础课程建设的研究
与实践（项目号：HSDJY03（Z））

科学出版社

内 容 简 介

本书介绍了利用 C 语言进行程序设计的基本知识。本书共 14 章, 主要包括算法与程序设计基础, C 语言程序设计概述, 数据类型、运算符与表达式, 顺序、选择和循环结构程序设计, 数组, 函数, 指针, 结构体与共用体, 编译预处理, 位运算, 文件以及面向对象程序设计与 C++ 基础。每章都配有课后习题。同时, 为方便广大教师教学和学生学学习, 本书配有相应的《C 语言程序设计习题及实验指导教程》(韩雪娜、金一宁主编, 科学出版社) 一书, 便于读者掌握程序设计的基本方法和编程技巧。

本书力求概念叙述准确、严谨, 程序设计简明扼要、通俗易懂、例题丰富。本书可作为高等院校各专业学生学习 C 语言程序设计的教材, 也可作为全国计算机等级考试二级 C 语言程序设计的参考用书, 还可作为程序设计人员的自学参考用书。

图书在版编目 (CIP) 数据

C 语言程序设计教程/韩雪娜, 金一宁主编. —北京: 科学出版社, 2017.8
(普通高等教育“十三五”规划教材)

ISBN 978-7-03-054197-0

I. ①C… II. ①韩… ②金… 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 (2017) 第 201715 号

责任编辑: 宋 丽 袁星星 / 责任校对: 王万红
责任印制: 吕春珉 / 封面设计: 耕者工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

三河市良远印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2017 年 8 月第 一 版 开本: 787×1092 1/16

2017 年 8 月第一次印刷 印张: 19 3/4

字数: 465 000

定价: 48.00 元

(如有印装质量问题, 我社负责调换〈良远印务〉)

销售部电话 010-62136230 编辑部电话 010-62135763-2047

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

C 语言是目前国内外使用较广泛的程序设计语言，C 语言程序设计课程是国内外大学普遍开设的程序设计课程。C 语言具有丰富的数据类型，它提供的数据结构和控制结构适合于进行结构化程序设计，还可以实现汇编语言的大部分功能，使用灵活。C 语言程序具有完善的模块程序结构，可移植性好，而且可以直接实现对系统硬件及外部设备接口的控制，具有较强的系统处理能力。

本书不仅详细地介绍了 C 语言的语法规则，还对某些功能的系统实现及程序的执行过程做了必要的分析。本书着重强调 C 语言的基本概念，通过大量的例题分析和算法实例，使读者理解和掌握利用 C 语言进行程序设计的方法。

本书概念清楚、内容全面、题例和习题丰富，书中所有示例程序均给出了算法思路和算法步骤，并上机调试运行后给出结果，每个程序都遵循标准化的编程风格，便于学生理解和自学。

本书由多年讲授 C 语言程序设计课程的教师编写，由哈尔滨商业大学的韩雪娜、金一宁担任主编，李俊玲、孔庆彦、康婧担任副主编，张洪瀚担任主审。具体编写分工如下：第 1 章、第 9 章、第 11 章由韩雪娜编写，第 2 章、第 3 章、第 6 章由康婧编写，第 4 章、第 5 章、第 10 章由金一宁编写，第 7 章、第 8 章由李俊玲编写，第 12 章、第 13 章、第 14 章由孔庆彦编写。

在编写本书的过程中，编者得到了哈尔滨商业大学各级领导的大力支持和帮助，同时得到了哈尔滨商业大学计算机与信息工程学院教师们的关心和支持，在此向他们表示衷心的感谢。

由于编者水平有限，书中不妥之处在所难免，欢迎广大读者对本书提出宝贵意见和建议。

编 者
2017 年 5 月

目 录

第 1 章 算法与程序设计基础	1
1.1 算法概述	1
1.2 算法的常用表示方法	2
1.3 程序设计的典型算法	5
1.4 结构化程序设计方法	7
小结	8
习题	9
第 2 章 C 语言程序设计概述	10
2.1 C 语言发展简史	10
2.2 C 语言的特点	11
2.3 C 语言的构成和书写形式	12
2.4 C 语言程序的开发步骤	16
小结	18
习题	19
第 3 章 数据类型、运算符与表达式	21
3.1 C 语言的基本符号	21
3.2 数据类型	26
3.3 运算符和表达式	36
3.4 自增运算和自减运算	41
3.5 数据类型转换	42
小结	46
习题	46
第 4 章 顺序结构程序设计	49
4.1 数据的输出	49
4.2 数据的输入	52
4.3 复合语句和空语句	54
4.4 顺序结构程序设计实例	54
小结	56
习题	57

第5章 选择结构程序设计	59
5.1 关系运算符和关系表达式	59
5.2 逻辑运算符和逻辑表达式	60
5.3 条件运算符和条件表达式	61
5.4 if 语句	61
5.5 switch 语句	66
5.6 选择结构程序设计实例	68
小结	71
习题	71
第6章 循环结构程序设计	73
6.1 while 语句	73
6.2 do-while 语句	78
6.3 for 语句	80
6.4 三种循环语句的比较	85
6.5 循环嵌套	86
6.6 转移控制语句	89
6.7 程序举例	92
小结	95
习题	96
第7章 数组	100
7.1 数组概述	100
7.2 一维数组	101
7.3 二维数组	108
7.4 字符数组	113
小结	126
习题	127
第8章 函数	131
8.1 函数的基本概念	131
8.2 数组作为函数参数	137
8.3 函数的嵌套调用和递归调用	142
8.4 变量的作用域和存储	149
8.5 内部函数和外部函数	158
小结	160
习题	160

第 9 章 指针	165
9.1 指针的基本概念及定义	165
9.2 指针运算	167
9.3 指针变量与一维数组	174
9.4 指针与函数	182
9.5 指针与二维数组	189
9.6 二级指针	198
9.7 内存空间的动态分配	199
9.8 main 函数的参数	202
9.9 指针应用实例	203
小结	208
习题	209
第 10 章 结构体与共用体	212
10.1 结构体类型与结构体变量	212
10.2 结构体数组	215
10.3 结构体指针	216
10.4 共用体	219
10.5 枚举类型	220
10.6 用 typedef 定义类型	221
10.7 链表及其简单操作	222
小结	225
习题	225
第 11 章 编译预处理	229
11.1 宏定义	229
11.2 文件包含	232
11.3 条件编译	235
11.4 行控制	237
小结	238
习题	238
第 12 章 位运算	241
12.1 位运算符与位运算	241
12.2 位域（位段）	248
小结	251
习题	251

第 13 章 文件	254
13.1 文件概述	254
13.2 文件指针	255
13.3 文件的打开与关闭	256
13.4 文件的顺序读写	257
13.5 文件的定位与随机读写	260
13.6 其他相关函数	261
13.7 综合应用	262
小结	265
习题	265
第 14 章 面向对象程序设计与 C++ 基础	267
14.1 面向对象程序设计的基本概念	267
14.2 面向对象程序设计语言 C++ 简介	269
小结	286
习题	287
附录 1 习题参考答案	288
附录 2 C 语言的关键字	292
附录 3 常用字符与 ASC II 代码对照表	293
附录 4 C 语言运算符优先级和结合性	295
附录 5 C 语言常用库函数	296
附录 6 常见错误中英文对照表	300
参考文献	305



第1章 算法与程序设计基础

为了使计算机能够按人的意图处理事务，人们必须事先设计好完成各种任务的程序，并预先将它们存放在存储器中。程序主要包括对数据及处理问题的方法和步骤的完整而准确的描述。对数据的描述，就是确定在程序中要用到的数据的类型和数据的组织形式，即数据结构。对处理问题的方法和步骤的描述就是计算机进行操作的步骤，也就是所采用的算法。有人说过：“计算机科学就是研究算法的科学”，可见算法在程序设计中的重要性。

1.1 算法概述

什么是算法？瑞士著名的计算机科学家、Pascal 语言的发明者 N.沃思（Niklaus Wirth）教授提出了程序定义的著名公式：

程序=算法+数据结构

这个公式的重要性在于它说明了程序与算法的关系，同时说明了数据结构的选择也是十分重要的。对于程序而言，算法与数据结构是统一的关系。

一个算法就是一个有穷规则的集合，其中的规则规定了一个解决某个特定类型的问题的运算序列。简单地说，任何解决问题的过程都是由一定的步骤组成的，我们把为解决问题采取的方法和步骤称为算法。

算法应当具备以下几个特点：

1) 有穷性。一个算法必须在执行有限步骤之后结束，即一个算法必须包含有限的操作步骤，不能无限地执行下去。

2) 确定性。算法的每一个步骤都应当是确定的、无二义性的，而不能是含糊的、模棱两可的。算法应当满足具体问题的要求，对于一切合法的输入都能产生符合要求的结果。

3) 可行性。算法的每一个步骤都应该是能够实现的、可操作的，并能得到确定的结果。

4) 有零个或多个输入。输入是指在执行算法时，对算法给出初始量。一个算法可以有多个输入，也可以没有输入。

5) 有一个或多个输出。算法的目的是解决问题，问题的解决就是输出。一个算法可以有一个或多个输出，没有输出的算法是没有意义的。

算法必须能在有限时间内完成，且对相同的输入具有相同的输出。在程序设计语言中，与算法密切相关的是语句，包括与程序执行处理有关的功能语句（如输入语句、输出语句、赋值语句和调用语句等）和与程序执行流程有关的语句（如条件语句、循环语句等）。对于程序设计而言，算法的确定也就是合理安排这些语句以完成人们要求实现的特定功能。

1.2 算法的常用表示方法

算法是实现某一个问题求解的方法和步骤，是解决问题的框架流程；而程序设计则是根据这一求解的框架流程进行语言细化，实现这一问题求解的具体过程。

一般可以使用以下工具描述算法。

1. 自然语言

自然语言即人们日常进行交流的语言，如英语、汉语等。用自然语言来描述的算法通俗易懂，便于用户之间进行交流。

【例 1-1】 根据半径求圆的面积。

算法：

步骤 1：输入圆的半径 r 。

步骤 2：判断 r 是否大于 0，如果大于 0 执行步骤 3，否则执行步骤 5。

步骤 3：计算 $\pi * r * r$ ，将结果给 S 。

步骤 4：输出圆的面积 S ，程序结束。

步骤 5：输出不是圆，程序结束。

用自然语言描述算法通俗易懂，但自然语言表示的含义往往不太严格，容易产生歧义。另外，将自然语言描述的算法直接在计算机上进行处理目前还有很多困难，不能实现。如果问题复杂，语句会比较烦琐、冗长，并且很难清楚地表达算法的逻辑流程等，往往需要根据上下文判断其含义。尤其对于含有选择结构、循环结构的算法，用自然语言不能方便直观地进行描述。

2. 流程图

流程图是用一些几何图形、线条及文字说明来描述算法的逻辑结构。自 20 世纪五六十年代流程图开始兴起，它几乎成为程序设计及算法描述的必用工具。

美国国家标准学会（American National Standards Institute, ANSI）规定了一些常用的流程图符号，流程图中的元素包括起止框、输入/输出框、判断框、处理框、流向线、连接点、注释框、端点等，它们的形状如图 1-1 所示。流程图的表示方法目前已经被大

多数国家接受和使用。

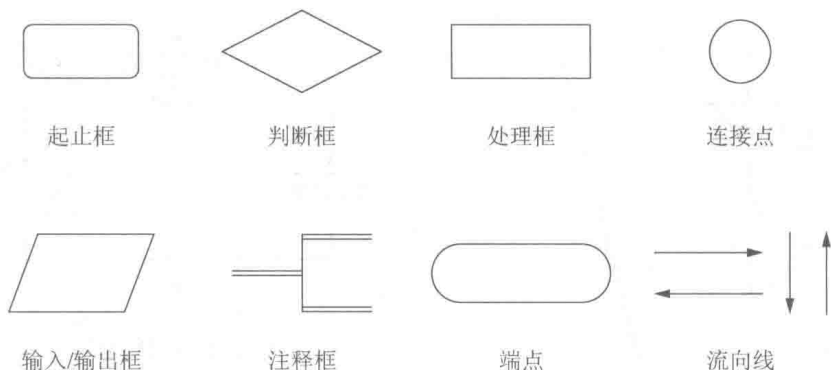


图 1-1 流程图元素

在流程图描述法中，一个流程图只允许有一个开始符和一个结束符，因此在流程图的构成元素中起止框在一个流程图中只允许出现两次，一个用于开始，一个用于结束；输入/输出框用于描述输入/输出数据；判断框用于描述选择结构的程序语句，所以判断框有两个输出，分别对应“是”和“否”两种条件判定结果。

注释框的作用在于为流程图中各个操作添加说明。注释框不是程序流程图的必要构成部分，可以根据需要在流程图的适当位置，为其添加必要的文字说明，以帮助读者更好地理解流程图。

流程图的特点是用一些图框表示各种类型的操作，用流向线表示这些操作的执行顺序。

结构化程序设计的3种基本程序结构是顺序结构、选择结构和循环结构。各种程序设计语言一般都具有这3种基本结构，但它们的表示形式有所差异。已经证明，由3种基本结构组成的算法，可以解决任何复杂的问题。用流程图表示3种基本结构如图1-2～图1-5所示。

(1) 顺序结构

顺序结构是一种简单的程序结构，程序依据语句出现的先后顺序依次执行，即程序执行了语句组1后才能执行语句组2，如图1-2所示。

(2) 选择结构

选择结构也称为分支结构，此结构必须包含一个条件判断框，根据条件判断结果决定程序执行的顺序。当“条件”为“真”（成立）时，执行语句组1；当“条件”为“假”（不成立）时，执行语句组2，如图1-3所示。

(3) 循环结构

循环结构也称为重复结构，通常用来重复执行某些操作语句。循环结构有以下两种形式。

1) “当型”：此种循环结构是先判断循环条件，若条件成立，则执行循环体，如此反复进行，直到条件不成立，跳出循环，如图1-4所示。

2) “直到型”：此种循环结构是先执行循环体，再判断循环条件是否成立，如果条

件成立，则继续执行循环体，直到条件不成立时，跳出循环，如图 1-5 所示。



图 1-2 顺序结构

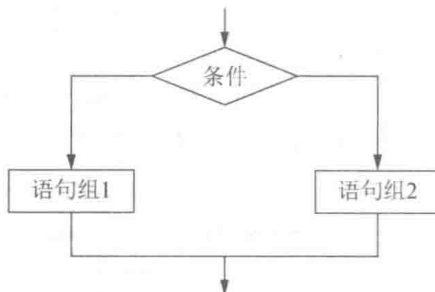


图 1-3 选择结构

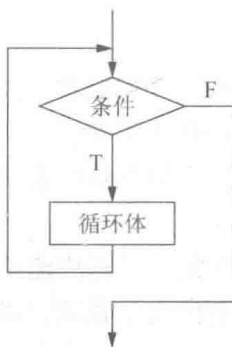


图 1-4 “当型”循环结构

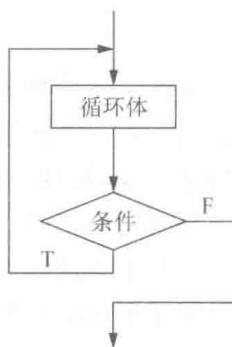
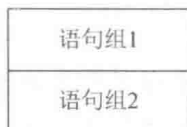


图 1-5 “直到型”循环结构

3. N-S 图

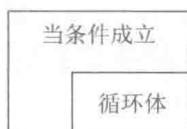
1973 年，美国学者 I.Nassi 和 B.Shneiderman 提出了一种在流程图中完全去掉流向线，全部算法写在一个矩形框内，在框内还可以包含其他框的流程图形式，即由一些基本的框组成一个大的框，这种流程图称为 N-S（两个学者的名字组合）结构流程图，简称 N-S 图（也称为盒图或 CHAPIN 图）。N-S 图包括顺序、选择和循环 3 种基本结构，如图 1-6 所示。



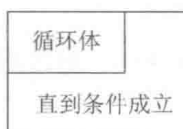
(a) 顺序结构



(b) 选择结构



(c) “当型”循环结构



(d) “直到型”循环结构

图 1-6 3 种基本结构 N-S 图

N-S 图比文字描述更加直观、形象、易于理解，使读者容易理解设计意图，给编程、排错、调试、维护都带来了便利。

用传统流程图和 N-S 图分别表示例 1-1 的算法，如图 1-7 所示。

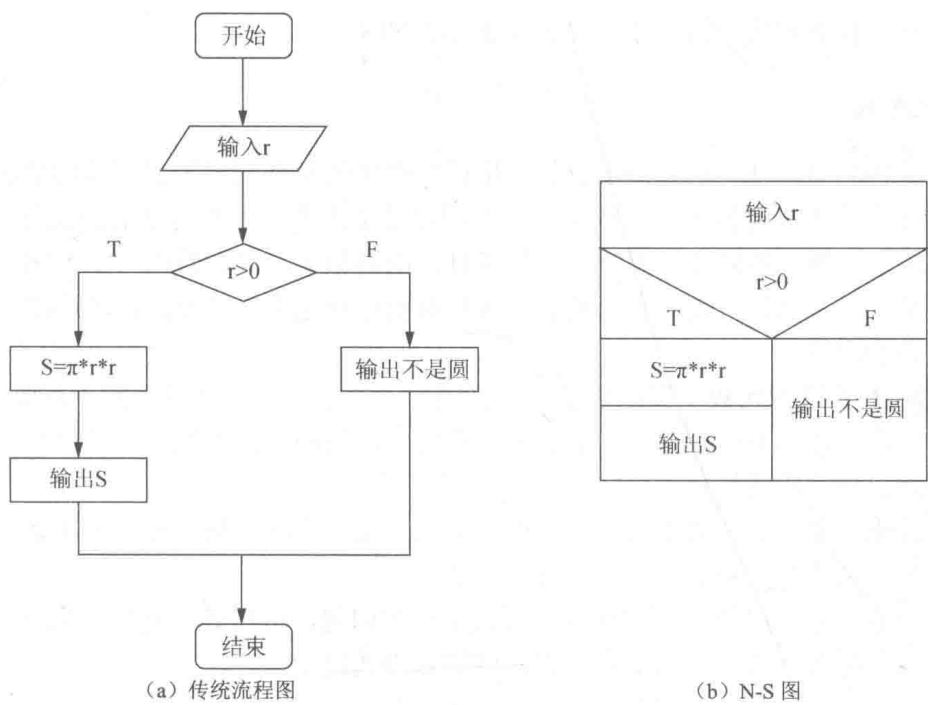


图 1-7 例 1-1 算法流程图

1.3 程序设计的典型算法

在程序设计时，常常要通过一些特定的算法求解。下面介绍几种比较常用的算法。

1. 迭代法

迭代法也称辗转法，是一种不断用变量的旧值递推新值的方法。一般的一元五次或更高次的方程及绝大部分的超越方程、微分方程问题无法用解析方法通过求根公式来求解，人们只能用数值方法求近似解。用事先估计的一个根的初始值 x_0 ，通过迭代公式 $x_{k+1} = f(x_k)$ 求出另一个近似值 x_1 ，再由 x_1 求出 x_2 ，从而获得一个求解的序列 $\{x_0, x_1, x_2, \dots, x_n\}$ 来逼近方程 $f(x) = 0$ 的根。这种求解的过程称为迭代。

例如，求某数 a 的平方根，已知求平方根的迭代公式为 $x_n = \frac{1}{2} \left(x_{n-1} + \frac{a}{x_{n-1}} \right)$ ($n=1, 2, 3, 4, \dots$)，取 $a/2$ 作为迭代的初始值，迭代结束条件为 $|x_n - x_{n-1}| \leq 10^{-5}$ 。

解题思路：根据题目知道 $x_0 = a/2$ ，根据公式可以求出 $x_1 = \frac{1}{2} \left(x_0 + \frac{a}{x_0} \right)$ ，然后判断是否满足结束条件 $|x_n - x_{n-1}| \leq 10^{-5}$ ，如不满足，则继续根据 x_1 求出 x_2 ，然后继续判断结束条件，依此类推，直到求出满足结束条件的 x_n 。

2. 枚举法

枚举法也叫穷举法。它的基本思想是根据所提出的问题列举所有可能的情况，并根据题目的部分条件确定答案的大致范围，然后在此范围内对所有可能的情况逐一验证，直到全部验证完毕。若验证结果符合题目条件，则列举情况是题目的一个答案；若验证结果均不符合题目条件，则题目无解。利用计算机运算速度快的特点，可以很方便地进行枚举问题的求解。

【例 1-2】我国古代数学家张丘建在《算经》一书中曾提出过著名的“百钱买百鸡”问题，该问题叙述如下：鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一；百钱买百鸡，则翁、母、雏各几何？

翻译过来，意思是公鸡 1 只 5 元，母鸡 1 只 3 元，小鸡 3 只 1 元，现在要用 100 元买 100 只鸡，则公鸡、母鸡、小鸡各多少只？

解题思路：如果用数学的方法解决百钱买百鸡问题，可将该问题抽象成方程式组。设公鸡 x 只，母鸡 y 只，小鸡 z 只，得到以下方程式组：

$$\begin{cases} 5x + 3y + \frac{1}{3}z = 100 \\ x + y + z = 100 \\ 0 \leq x \leq 100 \\ 0 \leq y \leq 100 \\ 0 \leq z \leq 100 \end{cases}$$

如果按照方程的解题方法，没有办法得到确定解。可以利用枚举法，对各种可能的组合一一进行测试，输出符合条件的组合，即在各个变量的取值范围内不断变化 x 、 y 、 z 的值，列出 3 个变量全部可能的组合，若满足方程组则是一组解。

3. 递推法

递推法是一种简单的算法，即通过已知条件，利用特定关系逐次推出所需要的结果。递推法就是把一个复杂的计算过程转化为简单过程的多次重复，即找出新旧值之间存在的关系，每次重复都在旧值的基础上推出新值，并由新值代替旧值，直至推出最后值。递推算法分为顺推和逆推两种。

【例 1-3】小猴子吃桃子问题。小猴在一天内摘了若干个桃子，当天吃掉一半多一个，第二天又吃掉剩下桃子的一半多一个，以后每天都吃掉剩余桃子的一半多一个。直到第 7 天早上要吃的时候，发现桃子只剩一个了，则小猴一共摘了多少个桃子？

解题思路：解决这个问题应该使用逆推的方法。先从最后一天推出倒数第二天的桃

子数量,再从倒数第二天推出倒数第三天的桃子……设第 n 天的桃子数为 x ,则前一天的桃子数为 $x_{n-1} = (x_n + 1) \times 2$ 。

4. 递归法

递归法是一种直接或者间接地调用自身的算法。在计算机编写程序中,递归法对解决某一类问题是十分有效的,它往往使算法的描述简洁且易于理解。递归法把问题转化为规模缩小了的同类问题的子问题,然后递归调用函数(或过程)来表示问题的解。数学中很多函数是采用递归法来定义的。

【例 1-4】用递归法求 n 的阶乘。

递归定义函数如下:

$$n! = \begin{cases} 1, & n = 0 \text{ 时} \\ n(n-1)!, & n > 0 \text{ 时} \end{cases}$$

无条件的递归是毫无意义的,递归过程必须有一个递归终止的条件,即设计递归出口。例如,在阶乘定义中,当 $n=0$ 时,定义函数值为 1,这就是阶乘递归定义的递归出口。

有些实际问题既可以归纳为递推问题,又可以归纳为递归问题,但递推和递归的实现方法是不一样的。递推是从初始条件出发,逐次推出所需要的结果;而递归则是从过程本身到达递归边界。通常递归法要比递推法清晰易读,且结构比较简单。特别是在比较复杂的问题中,很难找到从初始条件推出所需结果的全过程,此时,设计递归法要比递推法容易得多。

1.4 结构化程序设计方法

结构化程序设计产生于 20 世纪 70 年代中期,是软件发展的一个重要的里程碑。在结构化程序设计出现之前,程序员按照各自的习惯和思路编写程序,没有统一的标准,也没有统一的方法。同样的一件事情,不同的程序员编写的程序所占用的内存空间、运行时间可能差异很大。更严重的是,这些程序的可读性和修改性很差。

结构化方法是一种传统的软件开发方法,它是由结构化分析、结构化设计和结构化程序设计 3 部分有机组合而成的。它的基本思想:把一个复杂问题的求解过程分阶段进行,而且这种分解是自顶向下,逐层分解的,使得每个阶段处理的问题都控制在人们容易理解和处理的范围内。

一个结构化程序就是用高级语言表示的结构化算法。用 3 种基本结构(顺序、选择、循环)组成的程序必然是结构化的程序,这种程序便于编写、阅读、修改与维护。这就减少了程序出错的机会,提高了程序的可靠性,保证了程序的质量。

结构化设计方法是以自顶向下、逐步求精、模块化为基点,以模块化、抽象、逐层分解求精、信息隐蔽化、局部化和保持模块独立为准则设计软件的数据架构和模块架构

的方法。结构化设计的过程是将问题求解由抽象逐步具体化的过程。采用这种方法便于验证算法的正确性。

结构化设计的步骤如下：

- 1) 评审和细化数据流图；
- 2) 确定数据流图的类型；
- 3) 把数据流图映射到软件模块结构，设计上层模块；
- 4) 基于数据流图逐步分解上层模块，设计中下层模块；
- 5) 对模块结构进行优化，得到更为合理的软件结构；
- 6) 描述模块接口。

结构化设计方法的设计原则如下：

- 1) 使每个模块尽量只执行一个功能（坚持功能性内聚）；
- 2) 每个模块用过程语句（或函数方式等）调用其他模块；
- 3) 模块间传送的参数用作数据；
- 4) 模块间共用的信息（如参数等）尽量少。

结构化程序设计方法的优点很多，但也不是毫无缺点。

1. 优点

结构化程序设计方法主要由 3 种逻辑结构组成，分别为顺序结构、循环结构和选择结构。结构化程序设计的每种结构只有一个入口和一个出口，这是结构化程序设计的一个原则。遵循结构化程序设计的原则，按照结构化程序设计方法设计出的程序具有明显的优点。

1) 程序易于阅读、理解、排错和维护。程序员采用结构化编程方法，将一个复杂的程序分解成若干个子结构，便于控制、降低程序的复杂性，因此容易编写程序，同时便于验证程序。

2) 提高了编程工作的效率，降低了软件开发成本。由于结构化编程方法能够把错误控制到最低限度，每个模块可以独立编制、测试，因此能够减少调试和查错的时间。

2. 缺点

1) 用户的要求难以在系统分析阶段准确定义，从而使系统在交付使用时产生许多问题。

2) 用系统开发每个阶段的成果来进行控制，不能适应事物变化的要求。

3) 系统的开发周期长。对于结构化程度较低的系统，在开发初期难以锁定其功能要求。

小 结

计算机科学是研究算法的科学。一个算法是一个有穷规则的集合，其中的规则规定了一个解决某个特定类型的问题的运算序列。算法的常用表示方法有自然语言、流程图、

N-S 图等。程序设计常用的典型算法有迭代法、枚举法、递推法、递归法等。设计程序时应采用结构化程序设计的方法，这样的程序便于编写、阅读、修改和维护。

习 题

1. 什么是算法？算法的基本特征有哪些？
2. 为什么要提倡结构化程序设计？简述结构化程序设计的 3 种基本结构。
3. 画出以下程序的流程图。
 - (1) 求 1~100 的奇数和和偶数积。
 - (2) 输入某个科目 30 位学生的成绩，统计所有成绩不及格的学生个数。
 - (3) 输出 100 以内所有能同时被 3 和 7 整除的数。
4. 简述结构化程序设计的优缺点。