高等学校“十二五”应用型本科规划教材

C语言程序设计

◎主编　李振富

西安电子科技大学出版社
http://www.xduph.com

高等学校“十二五”应用型本科规划教材

C 语言程序设计

主编　李振富

西安电子科技大学出版社

内 容 简 介

本书由浅入深，以实例教学的方法组织知识点，既利于初学者对基本知识点的理解和掌握，又锻炼了读者的编程能力。本书共 10 章，第 1 章简要介绍了 C 语言程序设计的基本概念与基本方法；第 2～9 章在讲解 C 语言基础知识的同时，循序渐进地引入程序设计的步骤、方法、要领等；第 10 章简单介绍了 C 语言在单片机中的应用并给出了 MCS-51 单片机 C 语言设计实例。

本书体系合理、内容清晰、逻辑性强、通俗易懂，既可作为高等学校的教材，也可以作为自学参考书。

图书在版编目(CIP)数据

C 语言程序设计/李振富主编. —西安：西安电子科技大学出版社，2013. 8
高等学校“十二五”应用型本科规划教材
ISBN 978-7-5606-3140-0

Ⅰ. ① C… Ⅱ. ① 李… Ⅲ. ① C 语言—程序设计—高等学校—教材 Ⅳ. ① TP312

中国版本图书馆 CIP 数据核字(2013)第 180114 号

策　　划　戚文艳
责任编辑　秦志峰　戚文艳
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　西安文化彩印厂
版　　次　2013 年 8 月第 1 版　2013 年 8 月第 1 次印刷
开　　本　787 毫米 × 1092 毫米　1/16　　印　　张　22
字　　数　519 千字
印　　数　1～3500 册
定　　价　39.00 元
ISBN 978-7-5606-3140-0/TP
XDUP 3432001-1

出 版 说 明

本书为西安科技大学高新学院课程建设的最新成果之一。西安科技大学高新学院是经教育部批准，由西安科技大学主办的全日制普通本科独立学院。学院秉承西安科技大学 50 余年厚重的历史文化传统，充分利用西安科技大学优质教育教学资源，闯出了一条以“产学研”相结合为特色的办学路子，成为一所特色鲜明、管理规范的本科独立学院。

学院开设本、专科专业 26 个，涵盖工、管、文、艺等多个学科门类，在校学生 1.2 万余人，是陕西省在校学生人数最多的独立学院。学院是“中国教育改革创新示范院校”，2010、2011 连续两年被评为“陕西最佳独立学院”。学院部分专业现已被纳入二本招生，成为陕西首批纳入二本招生的独立学院。

学院注重教学研究与教学改革，实现了陕西独立学院国家级教改项目零的突破。学院围绕“应用型创新人才”这一培养目标，充分利用合作各方在能源、建筑、机电、文化创意等方面的产业优势，突出以科技引领、产学研相结合的办学特色，加强实践教学，以科研、产业带动就业，为学生提供了实习、就业和创业的广阔平台。学院注重国际交流合作和国际化人才培养模式，与美国、加拿大、英国、德国、澳大利亚以及东南亚各国进行深度合作，开展本科双学位、本硕连读、本升硕、专升硕等多个人才培养交流合作项目。

在学院全面、协调发展的同时，学院以人才培养为根本，高度重视以课程设计为基本内容的各项专业建设，以扎扎实实的专业建设，构建学院社会办学的核心竞争力。学院大力推进教学内容和教学方法的变革与创新，努力建设与时俱进、先进实用的课程教学体系，在师资队伍、教学条件、社会实践及教材建设等各个方面，不断增加投入、提高质量，为广大学子打造能够适应时代挑战、实现自我发展的人才培养模式。为此，学院与西安电子科技大学出版社合作，发挥学院办学条件及优势，不断推出反映学院教学改革与创新成果的新教材，以逐步建设学校特色系列教材为又一举措，推动学院人才培养质量不断迈向新的台阶，同时为在全国建设独立本科教学示范体系，服务全国独立本科人才培养，做出有益探索。

西安科技大学高新学院

西安电子科技大学出版社

2012 年 7 月

高等学校“十二五”应用型本科规划教材
编审专家委员会名单

前　言

本书是以教育部高等学校非计算机专业计算机基础课程教学指导分委会提出的《高等院校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》为指导纲领，融学习指导、实例和测试习题为一体的教材。

C 语言是目前使用最广泛的高级程序设计语言之一，“C 语言程序设计”课程是计算机及相关专业的一门基础主干课程。为了让读者更好地掌握 C 语言及其程序设计的思维方式，培养理论联系实际的能力，本书根据“夯实基础、面向应用”的指导思想，加强了教材的基础性及应用性，并且注重培养读者的程序设计思维方式。

本书的主要特点：

(1) 以软件工程方法为指导，以结构化、模块化程序设计方法为主线。

(2) 注重基础知识，以原理为主线，以实例为引导，以应用为目的。

(3) 概念准确，内容丰富，重点突出，难点分散，由浅入深。

(4) 每章最后一节是经典的程序设计实例分析，实例多源自经典问题或实际问题，并兼具一定的趣味性，对其进行编程分析，可帮助学生进一步理解和掌握编程方法，从而激发学生的学习兴趣。

(5) 每章配有丰富的不同难易程度的测试练习题，以供教师和学生进行测试和练习。习题的题目内容有选择题、填空题、分析程序题、问答题、编写程序题等类型，参考了国家计算机等级考试命题的特点，具有一定的代表性，是学生进行总结复习的实用参考资料。

本书既可作为高等院校和大、中专院校的“C 语言程序设计”课程的主讲教材，也可作为自学用书及相关技术人员的参考书。

本书由李振富担任主编，丁雪芳、许元飞、李立红、崔海文、周燕参加了编写。其中，崔海文编写第 1、2 章，丁雪芳编写第 3、8 章，许元飞编写第 4、6、7 章，李立红编写第 5、9 章，周燕编写第 10 章。李振富审读了全书内容。

在本书的编写过程中参考了大量同类专著和教材，在此对所有参考文献的作者表示诚挚的感谢。

本书虽然经过多次反复修改才定稿，但是限于编者的水平，书中难免有不足之处，恳请读者批评指正，以便进一步修订完善。

编　者

2013 年 5 月

目　录

第 1 章　C 语言与程序设计

教学目标

1. 了解计算机语言及其发展。
2. 了解 C 语言的发展简史与特点。
3. 了解计算机的解题过程。
4. 掌握算法及其表示。
5. 了解常用的算法。
6. 理解结构化的程序设计方法。
7. 了解简单的 C 语言程序设计实例。

1.1　计算机语言及其发展

1. 计算机语言

自世界上第一台电子计算机 ENIAC 于 1946 年问世以来，伴随着计算机硬件的不断更新换代，计算机程序设计语言也有了很大的发展。在过去的几十年间，大量的程序设计语言被发明、被取代、被修改或组合在一起。计算机应用领域的不断扩大，对软件技术的要求越来越高，程序设计语言也在不断进步。

(1) 机器语言。最初的计算机编程语言是所谓的机器语言(也称为第一代语言)，即直接使用机器代码进行编程。机器语言即机器指令的集合。每种计算机都有自己的指令集合，计算机能直接执行使用机器语言所编写的程序。机器语言包括指令系统、数据类型、通道指令、中断字、屏蔽字、控制寄存器的信息等。机器语言用二进制数表示，是计算机能理解和执行的唯一语言。机种不同，其机器语言组合方式也不一样。因此，同一个题目在不同的计算机上计算，必须编写不同机器语言的程序。机器语言是最低级的语言。

(2) 汇编语言。为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一个特定指令的二进制串，比如，用“ADD”代表加法，“MOV”代表数据传递，等等。这样一来，人们就很容易读懂并理解程序在干什么，纠错及维护也变得方便了。这种程序设计语言称为汇编语言。但是，计算机不认识汇编语言符号，这就需要一个专门的程序，负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。汇编语言是第二代计算机语言，同样依赖于机器硬件，移植性不好，但效率十分高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精练而质量高，所以至今汇编语言仍是一种常用的软件开发工具。

(3) 高级语言。虽然汇编语言较机器语言已有很大的改进，但仍有两个主要缺点：一是涉及太多的细节；二是与具体的计算机结构相关。所以，汇编语言也是低级语言，被称为面向机器的语言。为了进一步提高编程效率，改进程序的可读性和可维护性，20世纪50年代以来，相继出现了许多种类的高级计算机编程语言(也称为第三代语言)，例如Fortran、Basic、Pascal、Java、C和C++等。其中，C和C++语言是当今最流行的高级计算机程序设计语言。

高级语言比低级语言更加抽象、简洁。它具有以下特点：① 一条高级语言的指令相当于几条机器语言的指令；② 用高级语言编写的程序同自然英语语言非常接近，易于学习；③ 用高级语言编写程序并不需要熟悉计算机的硬件知识。

同汇编语言类似，高级语言也需要专门的翻译程序(称为编译器或解释器)，即翻译成机器语言后，程序才能运行。因此，实现同样的功能，用高级语言编写的程序执行效率相对于机器语言和汇编语言来说是最低的。

2. 计算机语言的发展

计算机程序设计未来的发展方向是面向对象程序设计以及数据抽象，将会向完全面向对象、更易表达现实世界和更易于编写程序的方向发展。计算机语言发展的特点可以归结为以下几点：

(1) 易实现：只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。

(2) 面向对象：提供简单的类机制以及动态的接口模型。对象中封装状态变量以及相应的方法实现了模块化和信息隐藏，提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的复用。

(3) 安全性：网络、分布环境下有安全机制保证。

(4) 平台无关性：使程序可以方便地被移植到网络上的不同机器、不同平台。

1.2 C语言的发展简史与特点

1. C语言的发展简史

C语言是在20世纪70年代初问世的。1978年，美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。早期的C语言主要用于UNIX系统。由于C语言的强大功能和各方面的优点逐渐为人们认识，到了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。

C语言是从B语言衍生而来的，它的原型是ALGOL 60语言。1963年，剑桥大学将ALGOL 60语言发展为CPL(Combined Programming Language)语言。1967年，剑桥大学的Martin Richards对CPL语言进行了简化，于是产生了BCPL语言。1970年，美国贝尔实验室的Ken Thompson对BCPL进行了修改，提炼出它的精华设计了B语言，并用B语言写了第一个UNIX操作系统。1973年，美国贝尔实验室的D. M. Ritchie在B语言的基础上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。

为了推广UNIX操作系统，1977年D. M. Ritchie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，B. W. Kernighan和D. M. Ritchie

出版了名著——《The C Programming Language》，从而使 C 语言成为目前世界上流行最广泛的高级程序设计语言。

随着微型计算机的日益普及，出现了许多 C 语言版本。由于没有统一的标准，这些 C 语言版本之间出现了一些不一致的地方。为了改变这种情况，美国国家标准化协会(ANSI)对 C 语言进行了标准化，并于 1983 年颁布了第一个 C 语言标准草案(83 ANSI C)，1987 年又颁布了另一个 C 语言标准草案(87 ANSI C)。最新的 C 语言标准是在 1999 年颁布并在 2000 年 3 月被 ANSI 采用的 C99，但由于未得到主流编译器厂家的支持，直到 2004 年 C99 也未被广泛使用。

在 C 语言创建不久，面向对象程序设计的概念出现了，并且相应的面向对象程序设计技术迅速普及。美国贝尔实验室的 Bjarne Stroustrup 在 C 语言的基础上，弥补了 C 语言存在的一些问题，增加了面向对象的特征，于 1980 年开发出一种既支持过程也支持对象的语言，称为“含类的 C”，1983 年将其改名为 C++。

2. C 语言的特点

C 语言作为目前世界上使用最广泛的程序设计语言，被许多程序员用来设计各类程序，它的优势主要在于 C 语言具有结构化、语言简洁、运算符丰富、可移植性好等诸多特点。

(1) 结构化。C 语言是结构化的程序设计语言，其主要结构成分是函数，可通过函数实现不同程序的共享。另外，C 语言具有结构化的控制语句，支持多种循环结构，复合语句也支持程序的结构化。这些特点使得 C 语言层次清晰、结构紧凑，比非结构化的语言更易于使用和维护。

(2) 语言简洁。C 语言简洁、紧凑，在语言的表达方式上尽可能简单。C 语言使用一个运算符能够完成在其他语言中要用多个语句才能完成的操作。简洁的表达方式使程序的编写更加精练，减少了程序员的书写工作量，极大地提高了编程效率。

(3) 功能强大。C 语言不仅具有高级语言的通用性，能完成数值计算及对字符、数据等的处理，同时还具有低级语言的特点，能对物理地址进行访问，对数据的位进行处理和运算。C 语言这种兼具高级语言和低级语言功能的特点，使得它能够代替低级语言开发系统软件和应用软件，著名的 UNIX 操作系统 90% 以上的代码就是用 C 语言实现的。

(4) 数据结构丰富。C 语言不仅具有其他高级语言所具有的各种数据结构，而且 C 语言又赋予了这些数据结构更加丰富的特性，用户能够扩充数据类型，实现各种复杂的数据结构，完成各种问题的数据描述。

(5) 运算符丰富。C 语言除了具有其他高级程序设计语言所具有的运算符外，还具有 C 语言特有的运算符，比如增量运算符、赋值运算符、逗号运算符、条件运算符、移位运算符和强制类型转换运算符等。大量的运算符使得 C 语言绝大多数的处理和运算都可以用运算符来表达，提高了 C 语言的表达能力。

(6) 生成的代码质量高。实验表明，用 C 语言开发的程序生成的目标代码的效率只比用汇编语言开发同样程序生成的目标代码的效率低 10%～20%。由于用高级语言开发程序描述算法比用汇编语言描述算法要简单、快捷，而且编写的程序可读性好，修改、调试容易，所以 C 语言成为人们用来开发系统软件和应用软件的一个比较理想的工具。

(7) 可移植性好。由于 C 语言程序本身不依赖于机器的硬件系统，因此用 C 语言编制的程序只需少量修改，甚至可以不用修改就可以在其他的硬件环境中运行。正因为 C 语言

程序的可移植性好，UNIX 操作系统才可以迅速地在各种机型上得以实现和使用。

C 语言虽然具有上述这些优点，但也并非尽善尽美，它也存在一些缺点。比如，程序设计人员可以利用指针对任意的物理地址进行访问，而且可以不加检验，这就有可能访问到被禁止访问的内存单元，从而造成程序错误甚至系统瘫痪，但是这样的错误却无法被 C 编译系统检验出来。因此程序设计人员必须要透彻地理解 C 语言，才能避免可能出现的错误。

1.3 编制程序过程

软件的开发一般采用软件工程的方法，经过分析、设计、编码、调试和运行等阶段。每个阶段都有明确的任务，后一阶段的工作是在前一阶段结果的基础上进行的。对于初学者而言，需要解决的问题和编写的程序一般比较简单，只要将问题分析清楚，找到解决问题的方法，编写程序就不是一件困难的事情了。

使用 C 语言解题时，在程序中有两个方面的描述，即数据描述和处理步骤(算法)描述，后者处理前者的数据。因此，在编写 C 语言程序之前，应该将问题所涉及的数据、已知条件和解题步骤都分析清楚。

1.3.1 程序设计的过程

程序设计的过程大致分为三个基本步骤：分析问题(Question)、设计算法(Algorithm)及编写程序(Program)，简称 QAP 方法。分析问题，即对问题进行定义与分析；设计算法，即设计程序的轮廓(结构)，并画出程序的流程图；编写程序，即采用一种计算机语言(如使用 C 语言)实现算法编程。

实际上，程序员在编写程序时，一般应该遵循以下步骤：

(1) 了解题目要做什么。

(2) 明确处理的数据及其特征，确定处理方案。

(3) 清楚地描述对数据的处理步骤——算法设计，以便得到预期的结果。算法是解决问题的核心，是程序的灵魂，是程序设计过程中的一个重要环节。

(4) 对处理步骤进一步细化——逐步求精算法，直到能够使用程序设计语言编写程序实现。

(5) 将算法转换为程序。将算法转换为程序是一项包含众多技巧的工作，需要掌握完整的计算机程序设计语言与程序设计的知识。

(6) 运行程序，分析结果。通过对源程序进行编辑、编译和链接，得到可执行的程序，并且运行所得到的可执行程序。如果出现编译错误或运行结果不正确，则要修改源程序，重新进行编译、链接和运行，直到得到满意的结果。

对于一个复杂的问题，可以将其分解成若干个容易处理、可单独解决的子问题，然后分别加以解决。

1.3.2 书写程序时应遵循的规则

每一种程序设计语言都具有自己的语法规则和特定的表达方法。一个程序只有严格按

照语言规定的语法和表达方式编写，按照一定的规则书写，才能保证编写的程序在计算机中正确地执行。从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规则：

(1) 一个说明或一个语句占一行。

(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，并单独占一行。

(3) 低一层次的语句或说明可比高一层次的语句或说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。

在编程时应力求遵循这些规则，养成良好的编程风格。

1.4　算法及其表示

1.4.1　算法的概念

算法是精确定义的一系列规则的集合，这些规则规定了解决特定问题的一系列操作，以便在有限的步骤内产生出问题的答案。简单地说，算法就是为解决一个问题而采取的方法和步骤。

算法也可以理解为由基本运算及规定的运算顺序所构成的完整的解题步骤，或者看成是按照要求设计好的有限的、确切的计算序列，并且这样的步骤和序列可以解决一类问题。例 1-1 中的处理步骤就是一种使用自然语言描述的算法。

例 1.1　求两个正整数 m 和 n 的最大公约数(即同时能够整除 m 和 n 的最大正整数)。

欧几里德阐述了求两个数的最大公约数的过程——欧几里德算法。

第一步：以 n 除 m，并令 r 为所得余数(显然 $n>r\geq 0$)。

第二步：若 $r=0$，算法结束，n 即为 m 和 n 的最大公约数。

第三步：置 m←n，n←r，返回第一步。

算法中的符号“←”表示将该符号右边变量或表达式的值送到左边的变量中。

算法中，不仅各步骤间的顺序重要，在每步内的动作次序同样也很重要。

对于同一个问题，可以有不同的解题方法，即可以有不同的算法。为了有效地解题，不但要求算法正确，还要考虑算法的质量，通常是选择简单明了、结构清晰和计算高效的算法。算法必须具有以下特性：

(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。

(2) 确定性：算法中每一个步骤应当是确定的，而不能是含糊的、模棱两可的。

(3) 可行性：算法中描述的操作都可以通过已经实现的基本操作执行有限次数来实现。

(4) 有效性：算法中的每一个步骤应当能有效地执行，并得到确定的结果。

(5) 输入：有零个或者多个输入，即算法需要的必要信息。

(6) 输出：有一个或者多个输出，输出的结果是与输入有某些特定关系的信息，没有输出的算法是无意义的。

对于程序设计人员，必须会设计算法，并根据算法写出程序。

1.4.2 算法的表示方法

算法的基本组成结构有三种，即顺序结构、分支结构和循环结构。或者说，任何一个算法，无论其多么简单或者多么复杂，都可由这三种基本结构组合而成。算法有多种表示方法，常用的有自然语言描述法、传统流程图法、N-S 流程图法、伪代码法。下面分别简单介绍。

1. 自然语言描述法

自然语言可以是中文、英文、其他民族语言或数学表达式等。用自然语言描述算法通俗易懂，其缺点是文字有可能冗长，不太严格，容易产生歧义，表达分支和循环结构不方便等。除了很简单的问题，一般不采用自然语言描述来表示算法。

2. 传统流程图算法

传统流程图是用约定的图框和流程线来表示运算或操作流程的图示形式。使用流程图表示算法，直观形象，易于理解。美国国家标准学会 ANSI 规定的一些常用流程图符号如图 1.1 所示。这里画出例 1.1 求最大公约数的算法的传统流程图，如图 1.2 所示。

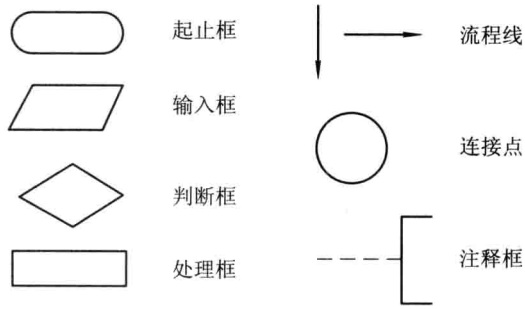


图 1.1 常用流程图符号

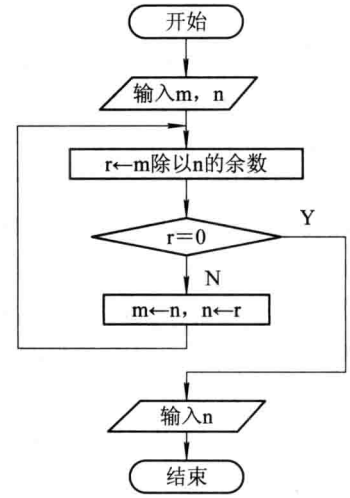


图 1.2 最大公约数的算法传统流程图

3. N-S 流程图法

1973 年，美国学者提出了一种新型流程图——N-S 流程图。N-S 流程图是一种结构化流程图，适合于表示结构化算法，它完全取消了带箭头的流程线，全部算法写在一个矩形框内，框内还可以包含从属于它的框。

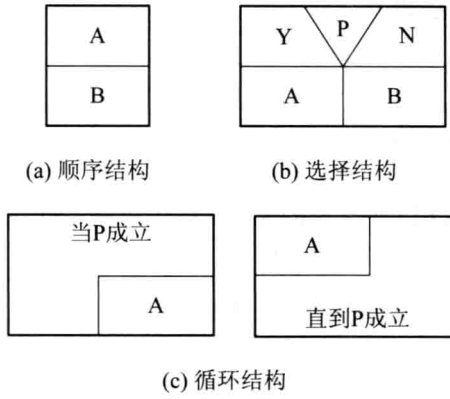
N-S 流程图使用的流程图符号及其基本结构如图 1.3 所示。图中，A 和 B 为某种操作或功能模块，P 为判断条件。

(1) 顺序结构：A 操作和 B 操作按顺序进行，先进行 A 操作，再进行 B 操作。

(2) 选择结构：根据判断条件 P 是否成立，在 A 和 B 两种操作中选择一种。若判断条件 P 成立，选择 A 操作执行；否则，选择 B 操作执行。

(3) 循环结构：根据判断条件 P 是否成立，决定 A 操作是否被重复执行。C 语言中的循环结构分为 while 循环(也称为当型循环)和 do-while 循环(也称为直到型循环)。while 循环式先判断条件 P，若成立，则重复 A 操作；do-while 循环是先执行 A 操作，再判断条件 P，

若成立，则重复 A 操作。在此画出例 1.1 求最大公约数算法的 N-S 流程图，如图 1.4 所示。



(a) 顺序结构　(b) 选择结构

(c) 循环结构

图 1.3　N-S 流程图基本结构

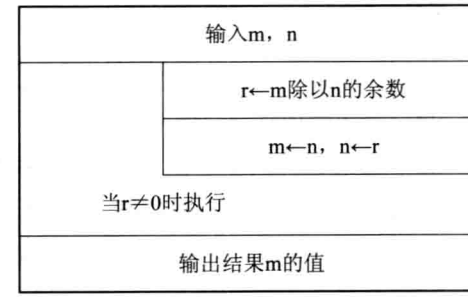


图 1.4　最大公约数的算法 N-S 流程图

4. 伪代码法

伪代码使用介于自然语言和计算机语言之间的文字和符号来描述算法。其采用了类似程序设计语言的语句表示算法，但是伪代码不是一种程序设计语言，不涉及程序设计的细节。伪代码表示算法不使用图形符号，它比使用传统流程图和 N-S 流程图表示算法更方便、紧凑，并且更易懂，便于将算法转换为各种计算机语言的程序。例 1.1 求最大公约数算法的伪代码(伪 C 语言)可以表示如下。

```
算法开始
输入 m,n;
do{
  r←以 n 除 m 的余数;
  m←n;
  n←r;
} while(r≠0);
输出 m;
算法结束
```

可以看出，用伪代码表示的算法与计算机程序设计语言已经很接近，很容易将其改写成计算机程序。对计算机程序设计语言比较熟悉时，使用伪代码(伪 C 语言)表示算法易于转换为实际的计算机语言程序。

1.4.3　简单算法举例

例 1.2　求 1×2×3×4×5。

最原始方法：

步骤 1：先求 1×2，得到结果 2。

步骤 2：将步骤 1 得到的乘积 2 乘以 3，得到结果 6。

步骤 3：将 6 再乘以 4，得 24。

步骤4：将24再乘以5，得120。

这样的算法虽然正确，但太复杂。改进的算法如下：

S1：使t=1

S2：使i=2

S3：使t乘i，乘积仍然放在变量t中，可表示为t乘i→t

S4：使i的值加1，即i加1→i

S5：如果i小于等于5，返回重新执行步骤S3以及其后的S4和S5；否则，算法结束。

如果计算100！只需将步骤S5中的“i小于等于5”改成“i小于等于100”即可。

如果要求1×3×5×7×9×11，算法只需做很少的改动即可，如：

S1：1→t

S2；3→i

S3：t乘i→t

S4：i加2→t

S5：若i小于等于11，则返回S3；否则，算法结束。

该算法不仅正确，而且是计算机较好的算法，因为计算机是高速运算的自动机器，实现循环轻而易举。

思考：将S5写成：“S5：若i小于11，返回S3；否则，算法结束”。那么该算法的操作结果会是什么？

1.5 常用算法介绍

算法是解决问题的方法，不同的领域有不同的方法。根据问题领域的不同，算法可分为数值问题和非数值问题的算法。数值问题的算法解决传统的数学问题，如求方程解的算法和求定积分算法等。与数值问题的算法相比，计算机科学对非数值问题的算法给予了更多的重视。计算机在非数值应用领域应用十分广泛，不同的应用领域有不同的非数值问题算法，已远远超过了在数值问题方面的应用。

算法的种类繁多，常用的算法有枚举法、递推法、递归法等。本节简要介绍这几种在程序设计中常用到的方法。

1. 枚举法

枚举法又称为穷举法。该方法是通过逐一考察问题的所有可能解，最终找出问题真正的解。枚举法要求问题的可能解必须是有限的，而且这些可能解是已知的。

例1.3 给定一个正整数，确定它的整数立方根是否存在，若存在，则找出这个立方根。

问题分析：如果一个正整数的整数立方根存在，则一定在0和这个数之间，而这之间正整数的个数是有限的，所以可以设计一个基于枚举法的算法。

```
算法开始
输入一个正整数给 n;
x←0;
while(x≤n 且 x*x*x≠n){
```