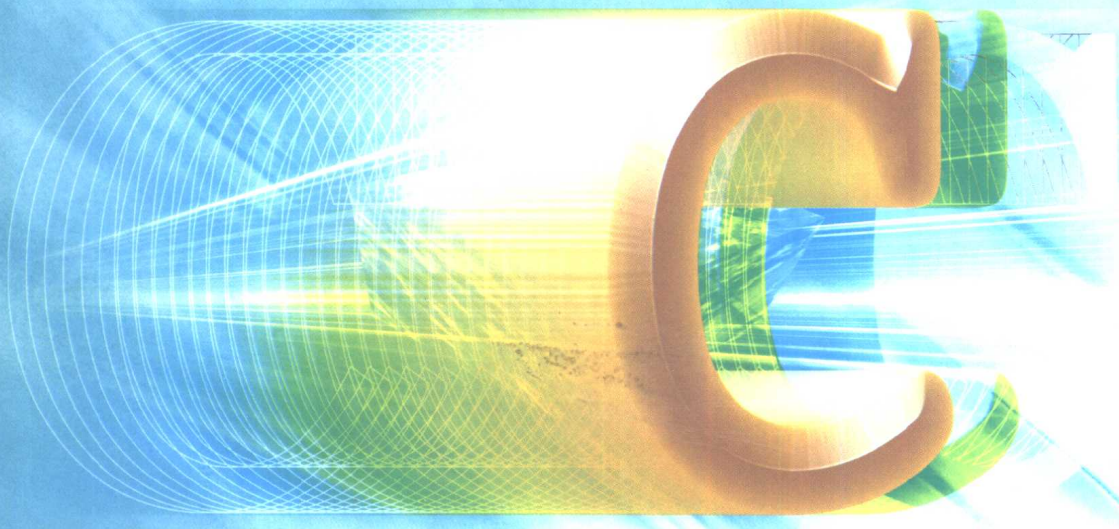
21世纪高职高专系列教材

C语言程序设计

中国机械工业教育协会　组编



机械工业出版社
China Machine Press

21世纪高职高专系列教材

C语言程序设计

中国机械工业教育协会　组编
主　编　厦门大学　朱立秒
副主编　大连理工大学　孟　军

机 械 工 业 出 版 社

C 语言功能丰富，使用灵活，可移植性好，既具有高级语言的优点，又具有低级语言的许多特点，是一种通用的程序设计语言。它不仅可以编写应用软件，而且特别适合于编写系统软件。

本书是针对高等职业教育的特点而专门编写的计算机基础课教学用书，是高职高专学生学习 C 语言程序设计的理想教材。本书共 12 章，按照循序渐进的原则，逐步介绍了 C 语言的基本概念、语法规则、程序设计方法，每章结尾都有一些典型复习思考题。本书着重强调培养学生掌握一些常用的基本算法，掌握阅读、编写 C 语言程序的基本方法、基本能力和初步技巧，掌握 C 语言的初步应用。

本书是作者根据多年教学经验编写而成的，在内容编排上尽量体现易学够用的特点，在文字叙述上条理清晰、简洁，便于读者理解。凡具有计算机初步知识的读者都能读懂本书。本书既可作为高职高专教材使用，也可供自学参考，或作为全国计算机等级考试（二级 C 语言）的参考用书。

图书在版编目（CIP）数据

C 语言程序设计/中国机械工业教育协会组编. —北京：机械工业出版社，2002.1
21 世纪高职高专系列教材
ISBN 7-111-08364-4

Ⅰ. C... Ⅱ. 中... Ⅲ. C 语言 - 程序设计 - 高等学校：技术学校 - 教材
Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2001）第 068750 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
责任编辑：王英杰
封面设计：姚　毅　　责任印制：路　琳
中国建筑工业出版社密云印刷厂印刷·新华书店北京发行所发行
2002 年 2 月第 1 版·第 1 次印刷
787mm×1092mm1/16·14.5 印张·字数 359 千字
0 001－4 000 册
定价：22.00 元

21世纪高职高专系列教材编委会名单

序

1999 年 6 月中共中央国务院召开第三次全国教育工作会议，作出了“关于深化教育改革，全面推进素质教育的决定”的重大决策，强调教育在综合国力的形成中处于基础地位，坚持实施科教兴国的战略。决定中明确提出要大力发展高等职业教育，培养一大批具有必备的理论知识和较强的实践能力，适应生产、建设、管理、服务第一线急需的高等技术应用性专门人才。为此，教育部召开了关于加强高职高专教学工作会议，进一步明确了高职高专是以培养技术应用性专门人才为根本任务；以适应社会需要为目标；以培养技术应用能力为主线设计学生的知识、能力、素质结构和培养方案；以“应用”为主旨和特征来构建课程和教学内容体系；高职高专的专业设置要体现地区、行业经济和社会发展的需要，即用人的需求；教材可以“一纲多本”，形成有特色的高职高专教材系列。

“教书育人，教材先行”，教育离不开教材。为了贯彻中共中央国务院以及教育部关于高职高专人才培养目标及教材建设的总体要求，中国机械工业教育协会、机械工业出版社组织全国部分有高职高专教学经验的职业技术学院、普通高等学校编写了这套《21 世纪高职高专系列教材》。教材首批 80 余本（书目附书后）已陆续出版发行。

本套教材是根据高中毕业 3 年制（总学时 1 600～1 800）、兼顾 2 年制（总学时 1 100～1 200）的高职高专教学计划需要编写的。在内容上突出了基础理论知识的应用和实践能力的培养。基础理论课以应用为目的，以必需、够用为度，以讲清概念、强化应用为重点；专业课加强了针对性和实用性，强化了实践教学。为了扩大使用面，在内容的取舍上也考虑到电大、职大、业大、函大等教育的教学、自学需要。

每类专业的教材在内容安排和体系上是有机联系、相互衔接的，但每本教材又有各自的独立性。因此各地区院校可根据自己的教学特点进行选择使用。

为了提高质量，真正编写出有显著特色的 21 世纪高职高专系列教材，组织编写队伍时，采取专门办高职的院校与办高职的普通高等院校相互协作编写并交叉审稿，以便实践教学和理论教学能相互渗透。

机械工业出版社是我国成立最早、规模最大的科技出版社之一，在教材编辑出版方面有雄厚的实力和丰富的经验，出版了一大批适用于全国研究生、大学本科、专科、中专、职工培训等各种层次的成套系列教材，在国内享有很高的声誉。我们相信这套教材也一定能成为具有我国特色的、适合 21 世纪高职高专教育特点的系列教材。

中国机械工业教育协会

前　言

C语言是目前世界上最为流行的计算机高级程序设计语言之一，它设计精巧，功能齐全，同时兼具高级语言和低级语言的特点，既适合于编写应用软件，又特别适合于编写系统软件，具有短小精悍、表达能力强、数据结构丰富、结构化好、目标代码质量高、可移植性好、提供接近于汇编语言的功能等特点，因此得到广泛应用。尤其在工程领域，由于 C 语言具有汇编语言的特点，能直接访问硬件，因此相对于其它高级语言，如 BASIC、FORTRAN 等，更适合于应用在机电、自动化、通信、化工等领域。

本书根据高职教育的特点，以应用为目的，以必需、够用为度，加强内容的针对性和实用性。在编写时力求适用面广，文字叙述简明，结构清晰、条理分明、概念清楚，使读者在学习本书后掌握基本的程序设计方法，熟练掌握 C 语言的基本概念和语法规则，掌握阅读、编写 C 语言程序的基本方法、基本能力和初步技巧，了解 C 语言的基本应用和综合应用。本书除了可作为高职高专理工科类教学用书外，也可作为成教、夜大、职大、函大等大专层次的教学用书，也可供自学参考，还可作为全国计算机等级考试（二级 C 语言）的参考用书。

全书共 13 章，总课时为 72 学时，上机实验 36 学时，各院校可根据实际情况决定内容的取舍。考虑到许多读者是在学习了计算机操作技术后直接学习 C 语言的，没有高级语言的基础，编者特意在第 1 章中补充简介了有关算法的内容。全书选择了大量的典型例题，每章后附有复习思考题，使读者能通过这些例题、复习思考题掌握 C 语言的语法特点和应用。书末第 13 章简单介绍了 C 语言的综合应用，为读者今后的深入学习和工作打下一定基础。

读者在学习 C 语言时，应注意以下几点：首先，读者要在阅读大量程序的基础上，自己亲自动手编写程序，逐步掌握和提高编程能力。其次，要重视上机训练，只有通过大量的上机练习，才能掌握 C 语言实际应用能力。最后，C 语言结构灵活，容易出错，必须通过大量的练习（包括上机练习）才能掌握 C 语言的一些特殊细微之处。对一些似是而非的问题，读者应亲自上机实验。

本书由厦门大学朱立秒任主编，大连理工大学孟军任副主编。其中第 6 章、第 7 章、第 9 章由大连理工大学孟军编写、其余章节由厦门大学朱立秒编写。

鉴于作者的水平，书中难免有错误或不当之处，敬请专家、同仁和广大读者批评指正。

编　者

目　录

第1章 绪　言

本章简单介绍程序设计及其语言、算法、结构化程序设计和C语言的基础知识。

1.1 程序设计及其语言

计算机已经广泛应用在各个领域。一个完整的计算机系统是由硬件系统和软件系统两大部分组成，计算机硬件由 CPU、存储器、输入设备和输出设备等物理设备组成，而软件由程序、数据和有关的文档组成。所谓程序就是人们为了使用计算机解决某些特定问题而设计的代码化指令序列，或者是符号化指令序列，或者是符号化语句序列。计算机具有特定功能的关键是人们设计了相应的程序，用户在计算机上运行这些程序，使得计算机具有这些功能。例如，要利用计算机来进行文字处理，人们设计了各种各样的文字处理程序（软件），如 MS-Word、WPS、写字板、记事本等等，用户在计算机上安装并运行了这些软件，计算机就具有文字处理能力，用户可以利用这些软件提供的命令进行各种各样的处理，如编辑、排版、打印等。在很多应用中我们看到计算机具有逻辑判断能力，具有较强的智能性。其实，计算机的核心硬件 CPU 本身并没有智能，但是它具有判断能力，且能够按照人们编写的程序来自动执行程序，所以关键在于人们设计的程序使得计算机具有各种各样的智能。

要解决一个实际问题，首先必须分析问题，建立数学模型，再选择计算方法，进行程序设计，然后上机调试运行，对运行结果进行分析，如果不能得到正确结果，则继续前述步骤，直至得到正确结果。用计算机解决实际问题的全过程如图 1-1 所示。

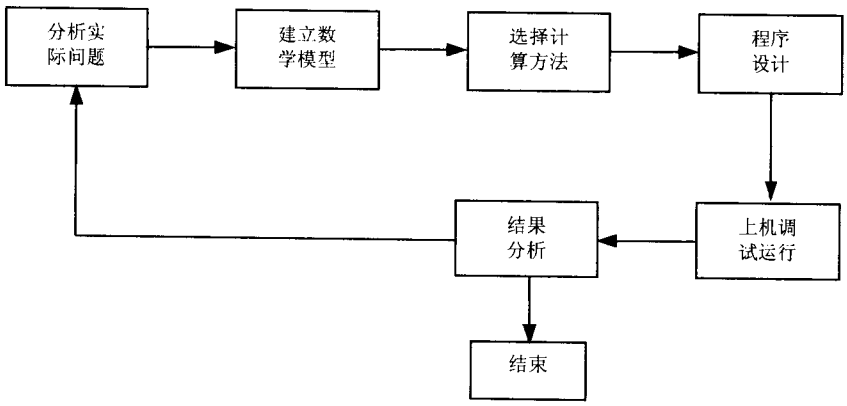


图 1-1　用计算机解决问题的过程

用通俗和不严格的说法，程序设计就是用计算机语言编写程序。著名的计算机科学家沃思提出一个有名的公式

程序=数据结构+算法

也就是说一个程序由数据结构和算法两大要素组成，两者缺一不可。数据是程序操作处理的对象，即指明操作的对象是谁？而算法是对这些数据的操作步骤，即指明对这些数据

进行哪些操作处理？数据结构用于描述数据的组织，即如何组织数据，以方便程序对这些数据进行高效的处理。计算机科学中有一门课程《数据结构》，专门研究对数据的组织和管理，有兴趣的读者可以详细阅读一下本套教材中的《数据结构》一书。

在编写程序时，人与计算机进行信息交换，能沟通人与计算机联系的语言称为“计算机语言”。按照计算机语言对机器的依赖程度分为机器语言、汇编语言和高级语言三种。

早期由计算机机器指令组成的语言称为机器语言，它难读难写，难以修改调试，容易出错，可移植性差，但是它可以被 CPU 直接识别执行，因此程序效率高，执行速度快，占用内存空间小。

汇编语言采用汇编助记符代替机器指令的操作码，用地址符号代替机器指令的地址码。用汇编语言编写的汇编源程序不能直接被 CPU 识别执行，必须经过手工或自动（由汇编程序）翻译成对应的机器指令，才能被计算机执行。汇编语言的特点在于：比较易读易改，也比较容易调试，可移植性仍然差，但是它的程序效率比较高，占用的内存空间较小。

高级语言是由表达各种不同意义的“关键字”和“表达式”按一定的语法规则组合而成的语言，它最接近自然语言。所以用高级语言编写程序，易读易改，容易调试，不易出错，且可移植性好。但是用高级语言编写的源程序必须由解释程序或者编译程序翻译成对应的目标代码（机器指令）再交由计算机执行，所以程序的效率较低，运行速度慢，占用内存空间大。

1.2 有关算法的基本知识

1.2.1 算法的概念

上节已介绍，算法是程序的两大要素之一，程序设计离不开算法的设计。所谓算法，就是解决问题的方法和步骤。解决同一问题，可能有多种不同的方法和步骤，例如：要求 1+2+3+…+100 的和，可以有以下几种方法：

(1) 先计算 1+2 的和为 3，再计算 3+3 的和为 6，一直继续直到求出 100 个数的和。

(2) 把 1+2+3+…+100 重新组合为(1+99)+(2+98)+…+100+50，可以得到和值为 50×100+50，即 5050。

(3) 利用等差级数求和公式 n×(n+1)/2 计算出和值为 100×(100+1)/2，即 5050。

这些方法都是正确的，每一种方法都是解决这一求和问题的算法。但是不同的方法有些简单、方便、运算量少，有些比较繁琐、运算量大，有些适合于计算机解题，有些则不适合。

计算机算法分为两大类：数值运算算法和非数值运算算法。数值运算算法主要用于解决数值计算问题，如求方程的根、求函数值、求定积分等。对于各种数值运算，现在都有比较成熟的算法可供选用。计算机科学中有一门课程《数值计算》就是专门研究此类数值运算的算法问题，有兴趣的读者可自己参阅。在本书中读者也会接触到一些常见、基本、简单的算法。计算机除了应用于科学计算以外，现在更多的应用于科学计算以外的其它领域，如事务处理，实时控制等。因此，非数值运算算法处理的问题非常广泛，除了数值运算以外，其它所有的算法都归入此类。在本书中读者会初步接触到个别非数值运算算法，如排序。

1.2.2 算法的表示和简单算法举例

例 1 交换两个变量 a、b 的值。

初接触计算机的读者可能会想这样的问题非常简单，只要 a=b(把 b 的值赋给 a)，再 b=a(把 a 的值赋给 b)即可。实际上，这种算法是错误的。保存数值的变量的存取特点是：读出时，该变量的值保持不变，写入时，则以新换旧。所以上述算法执行后，a、b 的值相等，且同为变量 b 的原值，a 被覆盖而丢失原值。可以借助于第三个临时变量 c 采用以下步骤实现：

S1：c=a　　S1：c=b
S2：a=b　　或　　S2：b=a
S3：b=c　　S3：a=c

S1、S2、S3…表示步骤 1、2、3…。

在第 2 章我们将看到，不用借助于第三个临时变量 c 也可实现 a、b 的值交换。

学习算法时，一方面要能够根据需要设计出算法，另一方面，在阅读程序时，要能理解算法并了解其用途，如以后在程序中看到类似算法时，应能掌握这种算法的作用和功能。

例 2　求 1+2+3+…+100 的和。

上面已经提到解决这一问题的几种算法，但用计算机解决类似问题时，不用上述算法，而是采用以下方法：

设两个变量 s 和 i，分别用以存放部分和及整数 1～100。采取以下步骤实现目标：

S1：s=0
S2：i=1
S3：s=s+i
S4：i=i+1

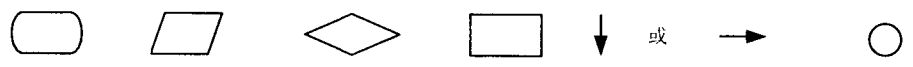
S5：如果 i 不大于 100，则返回重新执行步骤 S3 及其后续的 S4、S5。否则，算法到此结束，变量 s 中保存的就是要求的和。

可以看出：在执行到步骤 S5 时，需要进行判断，根据判断的结果要么返回步骤 S3 再顺序执行，这样构成循环；要么退出循环。

因为计算机是高速进行运算的自动机器，实现循环是轻而易举的，只要设计出采用这种算法的计算机程序，并交由计算机执行即可求解，所以，这种算法比较适合于用计算机来求解。此外，这种算法具有通用性和灵活性，只要对该算法略加修改，即可求解 1+3+5+…+99、2*4*6*8*10*12、10！等类似的数值运算问题。

算法的表示或者描述主要可以用以下几种方法：自然语言、流程图、N-S 结构图（参见下一小节）、伪代码和计算机语言。上述算法的表示是用自然语言描述，用自然语言表示的优点是通俗易懂，但文字冗长，容易出现歧义性，且对于包含分支和循环的算法，就不直观清晰，很不方便，因此一般不用。这时可以使用流程图来表示。

流程图是用一些图框表示各种操作，用图形表示算法，直观形象，易于理解。美国国家标准化协会 ANSI(American National Standard Institute)规定了一些常用的流程图符号，已被大家普遍采用，如图 1-2 所示。



起止框　输入输出框　判断框　处理框　流程线　连接点

图 1-2　常用的流程图符号

例 1 和例 2 的算法用流程图表示，分别如图 1-3 和图 1-4 所示。

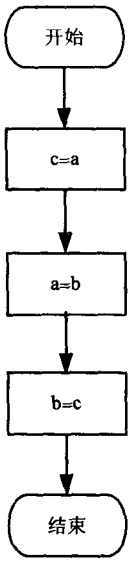


图 1-3　例 1 的流程图

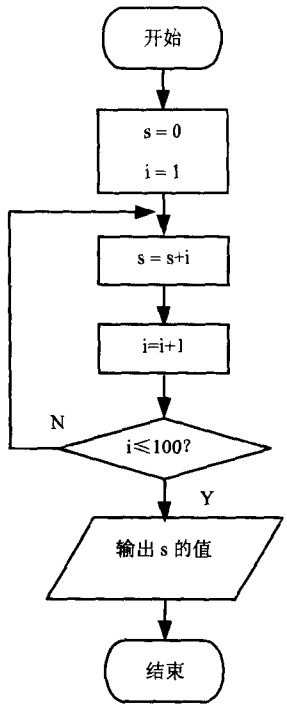


图 1-4　例 2 的流程图

如图 1-4 所示，用流程图表示例 2 的算法时，非常直观清晰，其中明显包含循环过程。对于循环，应该注意：循环的起始条件，即开始循环时 i、s 的初值分别改为 1、2，能否照样得到正确结果？循环的结束条件即 i≤100，可否改为 i<100？循环体中必须包含改变循环变量的操作，如 i=i+1，否则变成死循环。循环体中的 s=s+i 和 i=i+1 能否上下顺序对调？如果它们对调上下位置，则应该如何设定循环的初始条件和结束条件？

例 3　求 100 个数中的最大值。

可以采用日常生活中“打擂台”的方法，设多个变量，具体步骤为：

S1：i=1

S2：max 取值为 100 个数中的第一个数

S3：data 取值为 100 个数中的后续数据

S4：若 max<data，则 max 取 data 的值，否则 max 保持不变

S5：i=i+1

S6：若 i 小于 99，则返回步骤 S3，否则结束算法，此时 max 中保存的就是要求的最大值。

采用这种算法时，请读者注意，循环前 max 的值不能取零。当 100 个数不全为负数时，能给出正确结果；但当 100 个数全为负数时，算法结束后，max 中的值仍为零，不是要求的 100 个数中的最大值。所以，在设计算法时，应确保算法在任何情况下都能给出正确结果，而不是在大部分情况下能给出正确结果，在少数情况下出错。

这种算法同样具有通用性和灵活性，读者只要适当修改上述算法，即可求 1000 个数中的最小值。请读者自己画出该算法的流程图。

例 4 求自然数 m 和 n 的最大公约数。

采用辗转相除法，假设 m=192，n=120，则辗转相除法的步骤为：

		商	余数(r)
S1：m/n	192/120	1	72(r1)
S2：n/r1	120/72	1	48(r2)
S3：r1/r2	72/48	1	24(r3)
S4：r2/r3	48/24	2	0

由上述步骤可以看出，辗转相除法的算法为：先把 m 除以 n，若余数为 0，则 n 就是最大公约数；否则把上次除法运算的除数作为下次除法运算的被除数，把上次除法运算的余数作为下次除法运算的除数，一直相除，直到余数等于零为止，这时除数中保存的就是 m 和 n 的最大公约数。

在上述算法中，每次除法产生的余数分别存放在 r1、r2、r3…中，但针对不同的 m、n 值，无法事先知道除法要进行多少次，因而无法确定存放余数变量 r 的个数，实际上也没有必要设置多个变量 r1、r2、r3…，只需 m、n 和 r 三个变量即可求解。该算法的流程图如图 1-5 所示。注意体会 m、n 和 r 的作用和变化过程，并注意循环体中 m=n 和 n=r 的先后顺序能否对调？为什么？

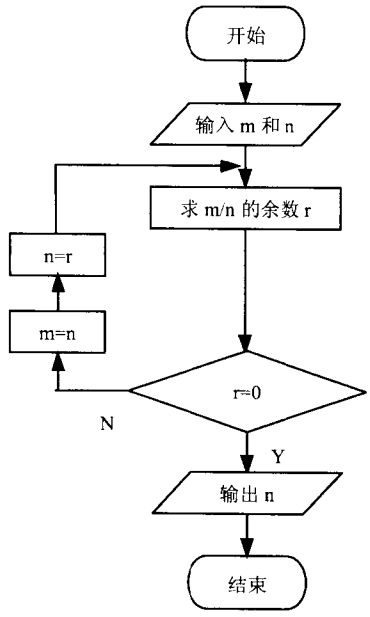


图 1-5 例 5 的流程图

1.2.3 算法的特性和质量

算法一般具有以下特点：

1. 有穷性 一个算法必须包含有限个操作步骤，且要在合理的时间范围内由计算机处理完成。

2. 确定性 算法中的每一个步骤必须是确定的操作，不能有歧义。

3. 有零个或多个输入 有些算法不需要从外界输入信息，即可求解，如例 2，而有些算法则需要 1 个或多个输入，如例 3 和例 4 就需要从外界输入数据。

4. 有一个或多个输出 所谓输出指的是算法的解，因为算法的目的是为了求解，所以算法都应该有输出。但要注意这里说的输出不完全指的是计算机屏幕显示输出，只要算法的功能和作用就是算法的输出，如例 1，虽然没有屏幕显示输出，但其作用是交换两个变量的值，这就是它的输出。没有必要设计一个无输出的算法，因为无输出的算法没有任何作用，当然是无意义的。

5. 有效性 算法中的每一个步骤都必须能有效执行，并能得到确定的结果。如把 100 除以 0 这样的操作是无效的。

解决同一问题可能有不同种算法，这些算法有优有劣，那么如何评价算法的质量呢？

算法首先要保证其正确性，即算法必须能正确求解，且要求在任何情况下，算法都能

得到正确的结果，否则就是不完善的算法，如例 3 中若设 max 的初值为零，则当 100 个数全为负数时，就不能正确求解。

其次还要考虑算法的效率，即空间效率和时间效率。一个算法的空间效率高，指的是采用该算法程序的目标代码短，占用内存单元少。一个算法的时间效率高，指的是采用该算法的程序执行时间短、速度快。例如计算多项式 $6x^3+4x^2+2x+1$，可用表达式 6*x*x*x+4*x*x+2*x+1 计算，也可用((6*x+4)*x+2)*x+1 计算，虽然两者的结果一致，但后者只需 3 次乘法和 3 次加法运算，而前者则需要 6 次乘法和 3 次加法运算，所以，虽然后者不明显直观，但执行时间短、速度快。

最后，还要考虑算法的结构化程度，见下节讨论。

1.2.4 三种基本结构和 N-S 结构图

上一小节介绍的流程图可以直观地表示算法，但传统的流程图用流程线指出各框的执行顺序，对流程线的使用没有严格限制。因此，编程者可以随意改变流程，使流程图变得毫无规律，增加程序阅读、编写、修改、调试和维护的难度，这样的算法不是好的算法。

为了提高算法的质量，必须限制流程线的随意使用，即不允许无规律地随意转向。为了解决这个问题，人们设想，规定出几个基本结构，再由这些基本结构按照一定规律组成一个算法结构。这样算法的质量可以得到保证和提高。

1. 三种基本结构　1966 年，Bohra 等提出了以下三种基本结构，用这三种基本结构作为表示一个良好算法的基本单元。

(1) 顺序结构，如图 1-6 所示，其中 A 和 B 两个框是顺序执行的。

(2) 选择结构（或称为分支结构）如图 1-7 所示，选择结构内必须包含一个判断框。根据给定的条件 p 是否成立而选择执行 A 框或 B 框，A 框和 B 框必有一个被执行。A 框或 B 框可以有一个为空，如图 1-8 所示，这是选择结构的一个特例。

(3) 循环结构（又称重复结构），即重复执行某一部分操作的结构。循环结构分两种：当型循环结构和直到型循环结构。

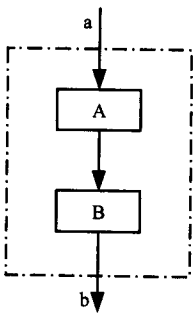


图 1-6　顺序结构

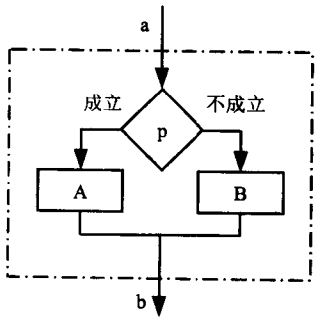


图 1-7　选择结构

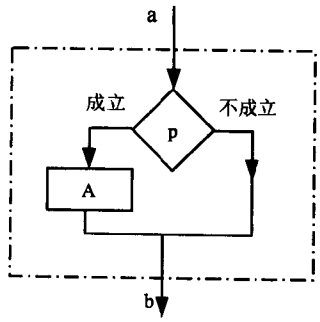


图 1-8　选择结构特例

当型循环结构如图 1-9 所示，首先判断条件 p1 是否成立，若成立则执行 A 框操作，执行完 A 后，再回到判断框判断 p1 是否成立，若仍成立，则再执行 A 框，如此反复循环，直到某一次 p1 不成立为止。

直到型循环结构如图 1-10 所示，先执行 A 框，再判断条件 p2 是否成立，若不成立，则再执行 A 框，然后再对 p2 进行判断，决定是否继续执行 A 框，直到给定的条件 p2 成立

为止。

由图 1-9 和图 1-10 可以看出，当型循环结构是先判断，后执行循环体，可能一次都不执行循环体；直到型循环结构则是先执行循环体，后判断，循环体至少被执行一次。需要用循环结构的一些算法，两种结构都可以使用，有时采用其中的一种结构会使算法更加简洁，用户可以根据实际情况来选用。

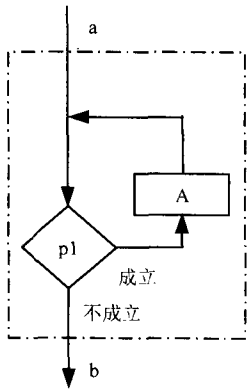


图 1-9　当型循环结构

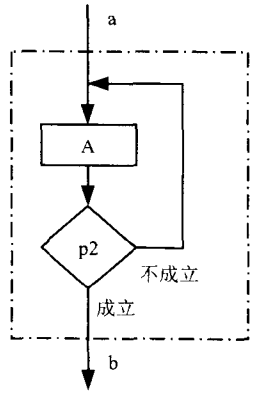


图 1-10　直到型循环结构

上述三种基本结构有以下共同点：只有一个入口，如图 1-6～图 1-10 中的 a 点；也只有一个出口，如图 1-6～图 1-10 中的 b 点；结构中的每一部分都有可能被执行到；基本结构内不存在死循环。

三种基本结构可以分别嵌套其本身，也可以互相嵌套。已经证明，可以利用以上三种基本结构组成的算法结构可以解决任何复杂的问题。

2. N-S 结构图　既然用三种基本结构组成的结构化算法可以表示任何复杂的算法结构，用于解决任何复杂问题，那么就可以不必使用结构之间的流程线了。

1973 年美国学者 I.Nassi 和 B.Shneiderman 提出了一种新的流程图形式，即 N-S 结构图。在 N-S 结构图中，完全不用流程线，而改用矩形框。它适合于结构化程序设计，所以应用广泛。图 1-6、图 1-7、图 1-9、图 1-10 这几种基本结构的 N-S 结构图分别如图 1-11~图 1-14 所示，分别表示顺序结构、选择结构、当型循环结构和直到型循环结构。用这三种基本框，可以组成任意复杂的 N-S 结构图，以表示算法。

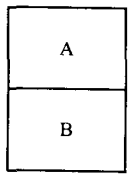


图 1-11　顺序结构的 N-S 结构图

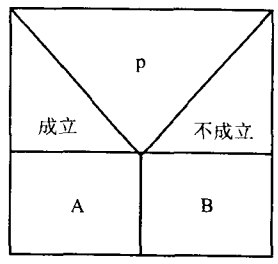


图 1-12　选择结构的 N-S 结构图

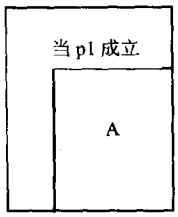


图 1-13　当型循环结构的 N-S 结构图

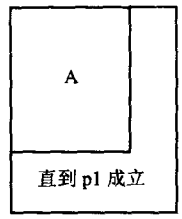


图 1-14　直到型循环结构的 N-S 结构图

例 2 和例 4 的 N-S 结构图分别如图 1-15 和 1-16 所示。

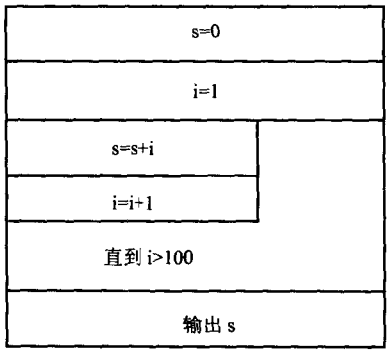


图 1-15　例 2 的 N-S 结构图

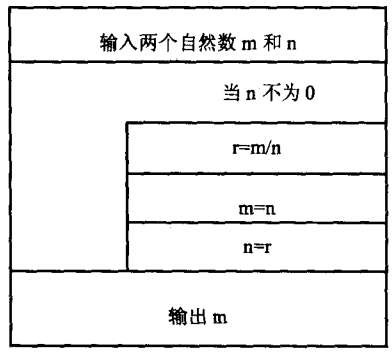


图 1-16　例 4 的 N-S 结构图

比较图 1-5 和图 1-13，图 1-5 的流程图不是基本的循环结构，应该对其适当改造，可以画出图 1-16 所示的包含当型循环结构的 N-S 结构图，注意流程图和 N-S 结构图中循环条件的区别和结果的保存位置（是在变量 m 还是 n 中）。请读者自己改用直到型循环结构实现此算法，画出对应的 N-S 结构图。

1.3　结构化程序设计方法

上节已经简单介绍了结构化的算法和三种基本结构。一个结构化程序就是用高级语言表示或实现的结构化算法。结构化程序设计方法包括以下几个步骤：自顶而下、逐步细化、模块化设计、结构化编码。对于一个大型、复杂的任务，先对其进行详尽的分析，把它分解成若干个相互独立的子任务（模块）；再把每一个子任务分解成若干个更小的子任务（子模块），一直到子任务足够小，可以直接用简单的算法来实现为止；然后对每一个分解后的子任务（子模块）进行程序编码，即模块化程序设计；最后按照刚才分解的相反顺序组合各个模块，最终解决问题。

这种设计时自顶而下、逐步细化而实现时自下而上、逐步组合的结构化程序设计方法全局观念强，有利于保证程序层次分明、结构清晰、算法正确。

1.4　C 语言的发展历史

C 语言是目前世界上最为流行的计算机高级程序设计语言之一。它设计精巧、功能齐全，既适合于编写应用软件，又特别适合于编写系统软件。

C 语言的发展历史最早可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向