工业和信息产业科技与教育专著出版资金
cbij.phei.com.cn

高等学校“十二五”公共课计算机规划教材

C语言程序设计基础

■ 主编　王园宇

COMPUTER TECHNOLOGY

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

高等学校“十二五”公共课计算机规划教材

C 语言程序设计基础

王园宇　主编

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

内 容 简 介

本书详细介绍C语言及其程序设计方法。全书共13章，主要内容包括：引言；数据类型、运算符及表达式；顺序结构程序设计；选择结构程序设计；循环结构程序设计；数组；指针类型；函数和变量的存储类型；结构体、联合体和枚举类型；编译预处理；位运算；文件；常见错误分析。本书展示了C语言灵活、高效的编程方法和在实践中的应用，努力做到理论与实践相结合。为了帮助读者学习，每章设有小结和习题。本书配有PPT、源代码等教学资源。

本书可作为高等院校学生学习C语言程序设计课程的教材，也可作为广大计算机程序设计人员和计算机程序设计爱好者的参考书。

图书在版编目(CIP)数据

C语言程序设计基础 / 王园宇主编. — 北京：电子工业出版社，2013.2
高等学校“十二五”公共课计算机规划教材
ISBN 978-7-121-19294-4

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(2012)第304015号

策划编辑：严永刚
责任编辑：王凌燕
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：787×1 092　1/16　印张：19.5　字数：577千字
印　　次：2013年3月第2次印刷
定　　价：35.00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

C 语言是一种被广泛使用的程序设计语言，其结构简单，数据类型丰富，运算灵活方便，既有高级语言的特性，又具有直接操控计算机硬件的能力，用其编写的程序，具有速度快、效率高、代码紧凑、可移植性好等优点，能够有效地用来编制各种系统软件和应用软件，是当今最为流行的计算机编程语言之一。因此，C 语言成为许多高校开展程序设计课程首选的语言工具。

目前介绍 C 语言的教材很多，但是在多年的教学实践中，我们发现适合高等学校程序设计课程的入门书籍有限。针对学生在学习 C 语言过程中的茫然，本书从程序设计与 C 语言间关系的角度出发，首先树立学生对程序设计的整体认识，然后有目的地开展 C 语言的学习。通过对内容由浅入深的渐进式编排，同时结合大量例题将 C 语言灵活、高效的编程方法和基本的应用展现给广大读者，力求实现 C 语言知识和应用开发能力的融会贯通。本教材是程序设计的基础教材，不但适合于初学者使用，对专业人员也有一定的参考价值。

本教材以美国国家标准 C 语言(ANSIC)为基本内容，以当前广为使用的 Visual C++6.0 编译系统作为集成开发环境，全面系统地介绍了 C 语言及其程序设计方法。本书是工业与信息化部规划教材。全书共 13 章，其中第 1 章介绍了程序设计与程序设计语言的基本知识，使读者建立起程序设计的概念，并且理解 C 语言在程序设计过程中的作用与特点；第 2 章介绍了 C 语言程序中使用的基本数据类型、运算符及表达式；第 3 章对顺序结构程序设计进行了介绍，并结合屏幕输出与键盘输入函数进行简单的 C 语言程序设计；第 4 章对关系运算符和逻辑运算符及它们所形成的条件表达式进行了介绍，在此基础上介绍了选择结构程序设计的思路与 C 语言条件语句；第 5 章介绍了循环结构设计的思想与 C 语言中的 3 种循环结构，以及由此构成的循环嵌套结构；第 6 章结合典型实例介绍了一维数组与多维数组的概念与应用；第 7 章介绍了指针这一具有 C 语言特色的特殊数据类型的基本概念与应用；第 8 章对函数进行了详细的介绍，在 C 语言函数概念的基础上用大量实例详细讲解了使用函数进行模块化编程的方法；第 9 章介绍了结构体、联合体和枚举类型这些用户自定义数据类型的基本概念与应用；第 10 章介绍了编译预处理；第 11 章结合典型例子介绍了 C 语言位运算、位运算符和位域的概念与应用；第 12 章对文件的使用进行了介绍；第 13 章对 C 语言程序设计过程中常见的错误进行了分析与讲解。附录中对 C 语言程序设计中常遇到的 ASCII 码、运算符及其优先级、C 语言常用的库函数等问题进行了汇总，为 C 语言程序设计的入门提供了参考资料。

本教材的参考学时数为 64 学时(含上机 32 学时), 可根据教学实际情况对教材相关内容进行取舍。此外，本书对各章节中的例题都做了问题分析与讲解，代码中添加了详尽的注释，为广大读者的学习提供了方便，同时帮助大家形成良好的编程习惯，提高学习效率和编程能力。

C 语言程序设计是一门实践性很强的课程，课堂教学必须与实践相结合，甚至有时候实践的作用更大一些。因此广大读者必须通过大量的编程训练，在实践中掌握语言知识，培养程序设计的基本能力，并逐步理解和掌握程序设计的思想和方法。

本教材第 1 章和第 11 章及附录由王园宇编写，第 2 章由王星魁编写，第 3 章和第 13 章由王颖编写，第 4 章由李月华编写，第 5 章由王爱莲编写，第 6 章由郭晓红编写，第 7 章由段跃兴编写，第 8 章由杨崇艳编写，第 9 章由杨丽凤编写，第 10 章和第 12 章由王幸民编写。全书由王园宇主编并统稿。本书在编写过程中得到了相洁同志的大力支持与热情帮助。此外，袁永红、侯慧杰、郭明霞参与了本书的校对工作。在此对他们及所有为本书的出版付出了辛勤劳动的同志表示衷心的感谢。

由于编者水平有限，书中难免存在一些缺点和错误，殷切希望广大读者批评指正。

反侵权盗版声明

举报电话：（010）88254396；（010）88258888
传　　真：（010）88254397
E-mail：　dbqq@phei.com.cn
通信地址：北京市万寿路 173 信箱
　　电子工业出版社总编办公室
邮　　编：100036

目　录

第1章 引　言

1.1 计算机程序的基本概念

现代计算机具有计算速度快、计算精度高、自动化程度好及通用性强等特点，在我们的生活中扮演了越来越重要的角色，被广泛应用于数值计算、数据处理、自动控制、计算机辅助设计及人工智能等众多领域。然而计算机并不能自主实现这些任务，其每一步的操作都是在计算机程序的指挥下完成的，可以说计算机程序就是计算机的灵魂。

程序一词来源于生活，通常指完成某项工作的一整套活动过程及活动方式。有了程序这个概念，人们就可以对一系列步骤的执行过程进行详实描述。比如到医院去看病的过程就是一个程序，如图 1-1 所示。按照这个程序大多数病人都能顺利得到医生的诊断与治疗。

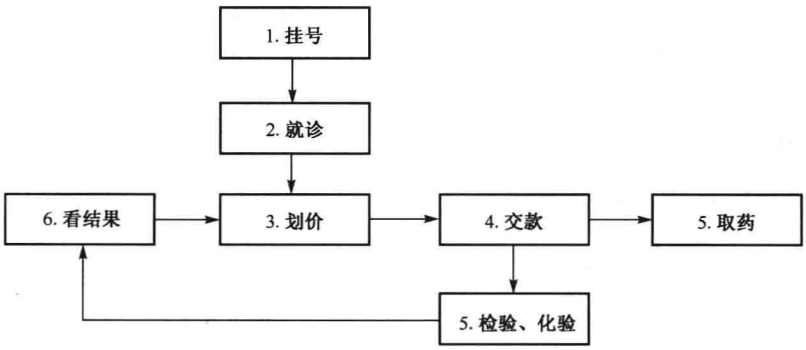


图 1-1　医院就诊流程图

同样的道理，计算机要正确运行并且完成相应的任务，也需要按照计算机程序的安排去执行。在计算机中，程序是指使计算机完成某一特定任务而编写的若干条指令的有序集合。计算机程序往往简称程序。这是计算机系统中最基本的概念。

一个计算机程序具有如下性质：

(1) 目的性。程序有明确的目的，在运行时能正确完成赋予它的功能。

(2) 分步性。程序为完成其复杂的功能，由一系列计算机能执行的步骤组成。

(3) 有限性。程序中所包含的步骤是有限的。永远执行不完的程序没有意义。

(4) 有序性。程序的执行步骤是有序的，不能随意改变这些步骤的执行顺序。

(5) 操作性。程序是对某些对象的操作，完成其有意义的功能。

1.2 计算机程序设计的步骤与方法

1.2.1 计算机程序设计的步骤

因为计算机的本质是程序的机器，所以只有通过对程序设计的学习才能更好地掌握和使用计算机。所谓程序设计简单说就是设计和编制程序的过程，是将实际问题用计算机方法解决的一个转化过

程。尽管不同规模的程序因其复杂程度不同，设计步骤有所差异，但是一些基本步骤是相同的，如下所述。

(1) 问题分析：一般来说程序要解决的对象是一个具体的问题，所以程序设计人员需要具体问题具体分析，在把任务交给计算机处理之前必须对问题做出明确的分析与定义，确定问题的条件及期望的结果，找出解决问题的规律。

(2) 算法设计：算法是解决问题的步骤及其描述，是程序设计的核心内容。算法是根据问题分析中的具体要求而设计的，是对问题处理过程的进一步细化。算法设计过程中，要求采用某种算法描述工具来表示程序运行的具体步骤。

(3) 程序编码：编码就是使用计算机编程语言编写源程序代码的过程。在这个过程中，首先应当选择编程语言，然后用该语言来实现上一步骤中设计好的算法。应当注意，对于相同的算法，采用不同的编程语言会对程序的执行效率产生影响。

(4) 调试运行：指在计算机上调试程序。调试运行程序有两个目的：一是消除由于疏忽而引起的语法错误或逻辑错误；二是用各种可能的输入数据对程序进行测试，验证程序是否对各种合理的数据都能得到正确的结果，而对不合理的数据也能做适当的处理。

(5) 文档编制：指整理并写出文字材料。文档包括程序说明文件和用户操作手册。程序说明文件记录了程序使用的算法、实现过程，用以保证程序的可读性和可维护性。用户操作手册则需要包含程序功能、运行环境、程序的安装与启动、基本参数的输入等内容。对于开发、维护周期较长的程序来说，适时地编制相应的文档显得尤其重要。

1.2.2 计算机结构化程序设计方法

1. 程序设计方法发展的几个阶段

程序设计方法历经了3个阶段，即初级阶段、结构化设计阶段及面向对象的程序设计阶段。

在计算机发展的初期，由于硬件价格比较昂贵，内存容量和运算速度都有一定的限制，所以如何少占内存提高计算机运行效率成为程序设计时要面对的主要问题，为此程序设计人员不惜牺牲程序的可读性而采取一些不规范的、晦涩的技巧来达到这一目的。

然而从20世纪60年代末开始，随着硬件条件的改善，出现了诸如操作系统、数据库等大型软件，同时应用软件需求迅速增长，规模不断扩大。使用以前那种随意而不规范的方法编制的大型软件越来越多地表现出了质量低下、可靠性差、开发周期长、维护费用高等缺陷。由软件错误而引起的系统报废事件屡有发生。人们开始认识到程序设计不只是编写出能够得到正确结果的程序，还应考虑程序的质量及如何得到高质量程序的问题。基于以上思路，1969年迪克斯特拉(E. W. Dijkstra)提出了所谓结构化设计方法，该方法以“结构清晰、容易阅读、容易修改、容易验证”为标准，目标是提高程序的可读性和可维护性。

到了20世纪70年代末，随着软件规模的进一步扩大，应用结构化程序设计方法已经无法降低和适应程序的复杂性和任务的多样性，于是面向对象的程序设计方法孕育而生了。面向对象的程序设计大大提高了软件的开发效率，减少了软件开发的复杂性，提高了软件系统的可维护性、可扩展性，是程序设计方法学的一个里程碑。

2. 学习结构化程序设计方法的目的

虽然出现了面向对象的程序设计方法，但结构化程序设计方法并没有过时，因为在面向对象的程

序设计中仍然需要用到结构化程序设计的知识，学习结构化程序设计对于培养我们的逻辑思维能力是有很大帮助的，可以说结构化程序设计是程序设计技术的基础。

3. 结构化程序设计思想

1966年，C. Bohm和G. Jacopini证明了只用顺序、选择和循环这3种基本结构就能实现任何“单入口、单出口”的程序，这就是结构化程序设计的理论基础。

结构化程序设计的基本思想有以下4点。

(1) 结构化编码

为了使程序具有良好的设计风格，编程时应使用以下3种控制结构进行编程：

顺序结构：顺序结构用来控制一个操作序列，从开始到结束顺序执行。

选择结构：根据条件判定的结果去执行不同分支中的语句。选择结构也称为分支结构，其为程序按不同情况自动选择步骤提供控制手段。

循环结构：根据条件判定的结果决定是否重复多次执行或一次也不执行同一组步骤。循环结构也称为重复结构，其为程序描述重复操作提供了控制手段。

(2) 结构化程序的特点

① 只有一个入口；

② 只有一个出口；

③ 不包含死循环(永远都执行不完的循环)；

④ 不包含死语句(永远也执行不到的语句)。

(3) 结构化程序的分析方法

结构化程序设计对问题采取所谓的“自顶向下，逐步求精”的方法进行分析。这种分析方法是一种由抽象到具体，由复杂到简单的分析过程。在分析问题时，先按组织或功能将一个大的复杂的问题分解成几个子问题，并把这些问题划分为一个层次，如果这一层的问题仍然复杂，就对这些问题再分解，并做相应的层次划分，直到不需要分解为止。每一次分析都是自顶向下的层次划分过程，每一次分解都是对上层问题的细化求精过程，最终会形成一个树形的层次结构分析结果。

例如，开发一个程序用于统计教师所代课程的平均分、最高分数和最低分数。其层次结构如图1-2所示。

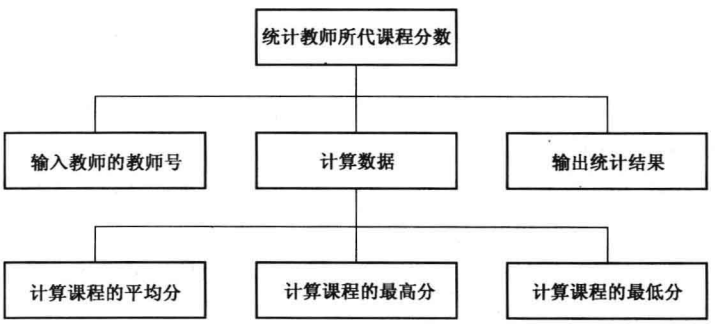


图1-2 教师所代课程统计程序的层次结构图

(4) 模块化设计

在层次结构划分好之后，就开始模块化设计了。模块化设计是在结构化程序设计的概念提出以后，经过逐步完善而形成的设计方法。模块化设计采取分治策略，将一个复杂的任务划分成若干基本模块，然后再对每一个模块进行具体化，逐步细化出一个个功能独立的模块。最下层模块完成最具体的功能。每一个模块都遵循高聚合性、低耦合性的分解原则。高聚合性是指一个模块只完成一项任务，功能相

对独立。低耦合性是指模块间的联系越松散越好，模块间只交换那些必需的信息。依照模块化程序设计方法设计出的程序条理清晰，任务不重复、不丢失，整个程序任务连接逻辑性强，即使程序很长，也能保持程序的易读性、易维护性和正确性。

程序设计时，编程人员根据要求分别完成一个或多个小模块。然后将这些模块分别测试、拼装形成一个软件。

结构化程序设计思想是程序设计的重要内容，在后面的学习中我们会知道：C语言通过函数来实现这种思想，C语言是一种理想的结构化程序设计语言。

1.3 算法及其表示

算法设计在程序设计中的地位如此重要，以至于著名的计算机科学家Nikiklaus Wirth提出了如下的公式：

程序 = 数据结构 + 算法

这个公式中的数据结构是指数据对象及其相互关系和构造方法，程序的数据结构描述了程序中的数据间的组织形式和结构关系。简单地说，数据结构是程序对数据的描述，而算法是程序对操作的描述。数据结构与算法有密切的关系，只有明确了问题的算法，才能设计出好的数据结构，同时，要选择好的算法又常常依赖于合理的数据结构。当然，程序设计除了算法和数据结构，还需要具备程序设计方法和程序设计语言方面的知识。本书由于篇幅所限不可能将这些知识都讲清楚，感兴趣的读者可以深入学习相关的专门知识。然而，由于算法的重要性所以本章还是要对算法的有关知识做初步介绍，以便为后面各章的学习奠定一些基础。

1.3.1 算法的概念与特点

算法是指为解决某个特定问题而采取的方法和步骤。算法是根据具体问题而设计的，问题不同，所设计的算法就不同。当然，即使对于同一个问题，也可以有不同的算法加以解决。

一个算法的设计具备如下特点：

(1) 有穷性。它包含两个方面：一方面是指一个算法应在有限的操作步骤内完成；另一方面是指算法操作应在有限的时间范围内完成。

(2) 确定性。算法中的每一个步骤都是确定的，即不能有二义性，这样才能确保对于同一个算法，相同的输入必然得出相同的执行结果。

(3) 有零个或多个输入。输入是指算法所需要的外部信息。在计算机上实现的算法，是用来处理数据对象的，在大多数情况下这些数据对象需要通过输入来得到。

(4) 有一个或多个输出。算法是有目的的操作，算法的目的是为了求解，这些解只有通过输出才能得到。没有输出的算法是没有意义的。

(5) 有效性。算法中的每一个步骤都应当能有效地执行，并得到确定的结果。

算法还具有顺序结构、分支结构和循环结构这3种基本的控制结构。

【例1-1】 任意输入两个数，按升序(从小到大排列)输出到屏幕上。请给出解决问题的算法。

分析：将任意两个数按升序排列，首先要比较大小，如果前一个数大就交换并输出，否则按原顺序输出。

步骤1：将输入的任意两个数赋给两个变量。

步骤 2：比较这两个变量中的数，如果前一个数大就交换，否则不交换。

步骤 3：将结果输出到屏幕上。

1.3.2　算法的表示

对算法的描述有自然语言、流程图、N-S 图、伪代码等表示方法，下面分别进行介绍。

1．自然语言

自然语言即人们日常生活中使用的语言。使用自然语言描述算法，通俗易懂，初学者容易掌握。【例 1-1】就是采用自然语言进行描述的。然而自然语言的含义有时会模糊，易发生歧义，描述文字显得冗长。这都容易造成理解上的偏差，因而不易直接转化为程序，所以用自然语言描述算法不太合适。通常除了一些很简单的问题外，一般不用自然语言表示算法。

2．流程图

流程图是一种被广泛使用的算法描述方法，它用一些图框和流程线来表示各种类型的操作。常见的流程图符号如图 1-3 所示。流程图方法形象直观，易于理解，便于发现算法出现的错误，可直观地将算法转化为程序。

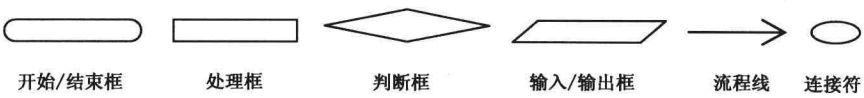


图 1-3　流程图符号

如图 1-4 所示为流程图描述的 3 种基本控制结构。图 1-4(a)为顺序结构，以只有 A、B 两个操作步骤的程序为例。图 1-4(b)为分支结构，其中 P 为一个条件，当 P 成立，则执行操作 A，否则执行操作 B。图 1-4(c)为当型循环结构，当 P 成立则重复执行操作 A，直到 P 不成立，循环结束。图 1-4(d)为直到型循环结构，先执行操作 A，然后判断 P，若 P 不成立则重复执行操作 A，直到 P 成立，循环结束。

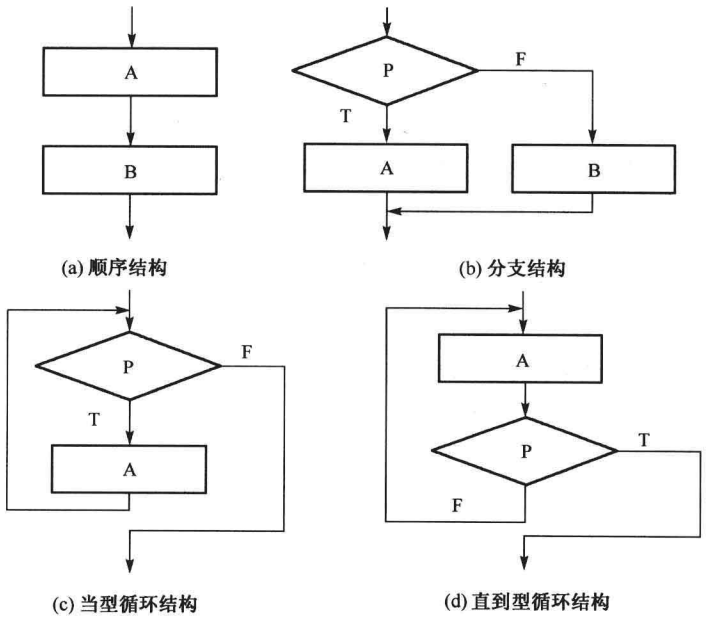


图 1-4　3 种控制结构的流程图

流程图描述法的缺点是占用篇幅较多，画流程图时既费时又不方便，如若有错也不易修改。另外，由于流程线没有约束，可以任意转向，从而造成程序阅读和修改上的困难，不利于结构化程序的设计。

3. N-S 图

随着结构化程序设计方法的出现，1973 年美国学者 I. Nassi 和 B. Shneiderman 提出了一种新的流程图形式。这种流程图中完全去掉了流程线，算法的每一步都用一个矩形框来描述，把一个个矩形框按执行的次序连接起来就是一个完整的算法描述。这种流程图用两位学者名字的第一个英文字母命名，称为 N-S 图。如图 1-5 所示为 N-S 图表示的算法的 3 种基本控制结构。其中图 1-5(a)为顺序结构，图 1-5(b)为分支结构，图 1-5(c)为当型循环结构，图 1-5(d)为直到型循环结构。

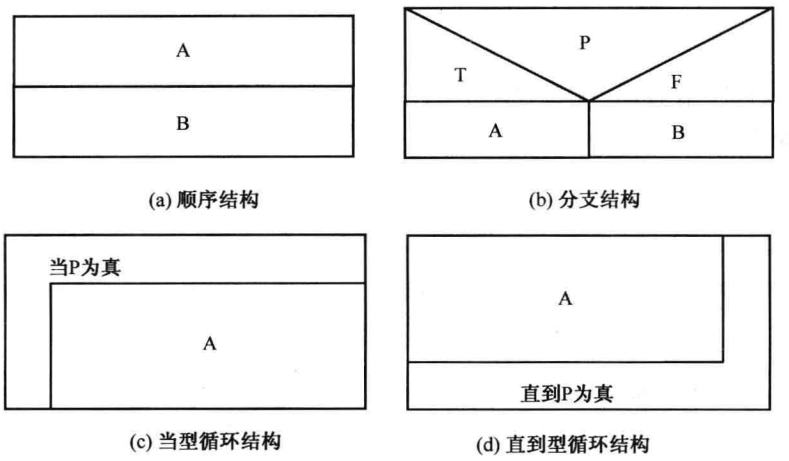


图 1-5　3 种控制结构的 N-S 图

N-S 图描述的算法在执行时只能从上到下顺序执行，从而避免了算法流程的任意转向，保证了程序的质量。N-S 图的另一个优点是形象直观，结构紧凑节省篇幅，非常适合结构化程序的设计。

4. 伪代码描述

流程图与 N-S 图表示算法直观易懂，但在设计时对其进行修改是很麻烦的，特别是所设计的算法反复修改时，为了设计算法方便，常采用伪代码对算法进行描述。伪代码是一种近似高级语言但又不受语法约束的一种语言描述方式，它用一种介于自然语言和程序设计语言之间的文字和符号来描述算法，用一行或几行表示一个基本操作。它书写方便，格式紧凑，便于编码实现。

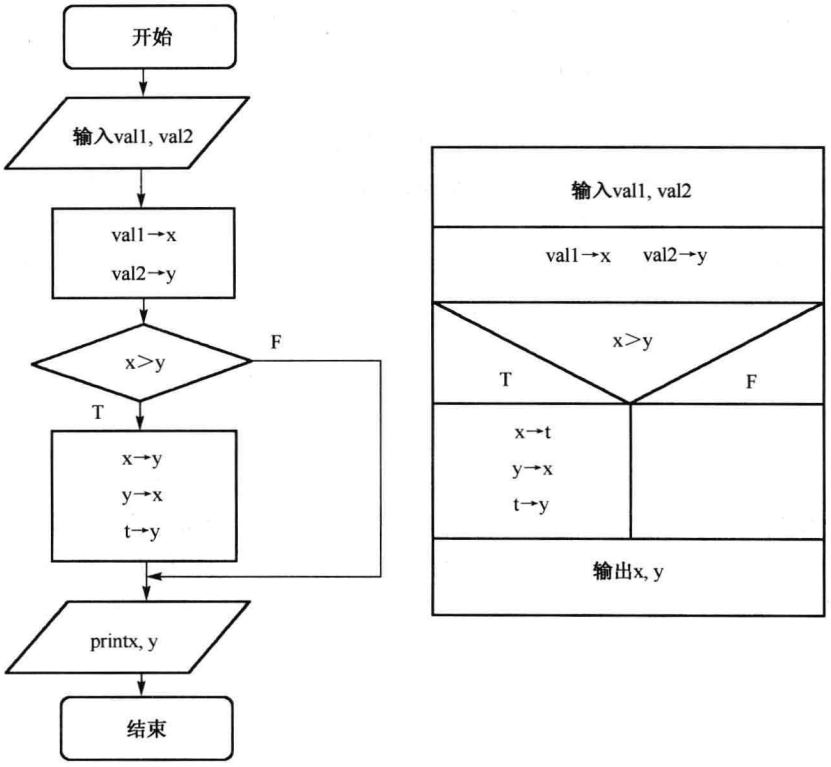
【例 1-2】 将【例 1-1】分别用流程图、N-S 图和伪代码进行描述。

(1) 用流程图描述的算法如图 1-6(a)所示。

(2) 用 N-S 图描述的算法如图 1-6(b)所示。

(3) 伪代码描述如下：

```
输入任意两个数 val1、val2 赋给变量 x 与 y
val1→x
val2→y
if x>y then x→t, y→x, t→y, print x,y
else print x,y
```



(a) 流程图描述 (b) N-S图描述

图 1-6 一个问题的流程图和 N-S 图描述

1.4 计算机程序设计语言

1.4.1 计算机程序设计语言简介

计算机程序最终是要计算机执行的，所以必须用计算机程序设计语言来编写程序。计算机程序设计语言是一个能完整、准确和规则地表达人们的意图，并用以指挥或控制计算机工作的“符号系统”。简单地说，计算机程序设计语言是人与计算机进行信息通信的工具。

1. 计算机程序设计语言分类

按照计算机语言的发展过程，程序设计语言可分为机器语言、汇编语言和高级语言三大类。

(1) 机器语言

机器语言是由 0、1 组成的机器指令的集合。机器语言可以被计算机直接执行。然而不同型号的计算机其机器指令是不能通用的，即按照某种计算机的机器指令编写的程序，不能在另一种计算机上执行，所以机器语言是一种面向机器的语言，也称为低级语言。机器语言程序具有计算机能够直接识别、执行效率高的优点，但其也具有难书写、难记忆、编程困难及可读性差等缺点。目前，除了计算机生产厂家的专业人员外，绝大多数程序员已经不再去学习机器语言了。

【例 1-3】 用机器语言程序实现 2+3 的运算。

10110000 00000010 ：将 2 送入累加器 AL 中

00000100 00000011 ：3与累加器AL中的值相加，并将结果放在AL中

11110100 ：停机结束

(2) 汇编语言

汇编语言克服了机器语言的一些缺点，采用助记符和符号地址来表示机器指令，因此也称为符号语言。在【例1-4】中，用助记符“MOV”表示数据传送，代替了【例1-3】中的机器指令“10110000”；用助记符“ADD”表示加法运算，代替了【例1-3】中的机器指令“00000100”；用助记符“HLT”表示停机结束，代替了【例1-3】中的机器指令“11110100”，这样使程序的可读性有了很大的提高。

【例1-4】 用汇编语言程序实现2+3的运算。

```
MOV  AL, 02
ADD  AL, 03
HLT
```

汇编语言增加了程序的可读性，但其还是低级语言，它也是面向机器的语言。此外，用汇编语言编写的程序不能被计算机直接识别和执行，必须由“汇编程序”，一种能把用汇编语言编写的程序翻译成机器语言程序的软件，将其转换成机器语言之后才能执行，这一过程称为汇编。由于汇编语言比机器语言可读性好，执行效率高，所以，许多系统软件的核心部分仍采用汇编语言编制。

(3) 高级语言

高级语言是一种接近于自然语言的程序设计语言，它按照人们的语言习惯，使用日常用语、数学公式和符号，按照一定的语法规则来编写程序。

【例1-5】 用C语言程序实现2+3的运算。

```
#include<stdio.h>
void main()
{
    int a;                          /*声明整型变量a*/
    a=2+3;
    printf("a=%d", a);              /*显示结果*/
}
```

高级语言与自然语言更接近，而与硬件功能相分离(彻底脱离了具体的指令系统)，编程者不必了解过多的计算机专业知识便可掌握和使用。高级语言的通用性强，兼容性好，便于移植。高级语言的产生，有力地推动了计算机软件产业的发展，进一步扩展了计算机的应用范围。

2. 翻译程序的方式

使用高级语言编写的程序称为高级语言源程序，但计算机不能直接接受和执行源程序，必须通过“翻译程序”将其翻译成机器语言形式后才能执行。这种“翻译”通常有两种方式：编译方式和解释方式。

编译方式：事先编好的一个称为“编译程序”的程序，将其放在计算机中。当高级语言源程序输入到计算机中时，编译程序便把源程序整个翻译成机器指令表示的目标程序，然后执行该目标程序，得到计算结果，如图1-7所示。

解释方式：事先编好的一个称为“解释程序”的程序，将其放在计算机中。当高级语言源程序输入到计算机中时，解释程序将源程序的每一条语句逐句翻译，逐句执行，即边解释边执行，如图1-8所示。所以解释方式并不产生目标程序。