

C 语言

程序设计教程

(修订版)

张松波 赵惠新 主编
吴 敏 张俊峰 主审

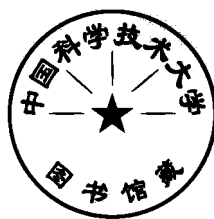
register
putchar
fclose
continue
acos
else
switch
short
const
limit.h
if
for
long
typedef
modf

哈尔滨工业大学出版社

C 语言程序设计教程

(修订版)

张松波 赵惠新 主 编
吴亚春 孔庆彦 副主编
吴 敏 张俊峰 主 审



哈尔滨工业大学出版社
哈 尔 滨

内 容 提 要

本书是按照教育部关于高等工程类专科学校教学基本要求编写的。全书共分十章，分别介绍了程序设计语言与程序设计概念、计算机算法与算法的图形表示法、C语言概述，C语言的数据类型、表达式和基本操作，C流程控制结构，函数和模块化程序设计，C语言的几种组合数据类型：数组、指针、结构体与共用体，C语言的文件操作，数据结构与程序设计关系，面向对象程序设计。

作为C语言程序设计课程教材，本书是直接基于计算机基本操作与使用知识的、面向初学程序设计语言的高等专科学校学生的入门与提高教材。本书也可以作为普通高校学生、各类计算机培训班学员的参考书和参加全国等级考试的辅助教材。

C 语言程序设计教程

Xiyuyan Chengxushejijiaocheng

张松波 赵惠新 主 编

*

哈尔滨工业大学出版社出版发行

哈尔滨工大节能印刷厂印刷

*

开本 787×1092 1/16 印张 16 字数 362 千字

2000 年 5 月第 2 版 2000 年 5 月第 2 次印刷

印数 3 501—4 500

ISBN 7-5603-1378-7/TP·122 定价 23.00 元

序

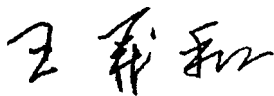
1968 年, George Forsythe 就说过:“在科学或技术教育中最有价值的收获是在一生中都有用的通用智慧工具。我认为自然语言和数学是这些工具中最重要, 而计算机科学排第三。” Forsythe 的话现在已经实现了, 如同数学一样, 计算机科学已被看作普通教育的基础学科。如同数学和其他科学一样, 计算机科学将继续含糊地分为两个范围, 即称之为“理论的”和“应用的”两部分。计算机科学大部分是偏重于实际的, 与数学相比较更像工程。在当今的信息时代, 不会使用计算机读写, 不会用计算机进行思维、工作和学习, 将成为下一个世纪的功能性文盲。

为了适应 21 世纪人才的需要, 我们的教育工作需为此作好准备。对此, 教育部提出要使中国的高校计算机基础教学跃上一个新台阶, 以改善学生的知识结构, 增强学生的适应能力, 具体提出了三个层次的教学目标。其中, 第二个层次的核心内容是, 在完成计算机文化基础教学内容的前提下, 进行计算机高级语言程序设计的教学, 在教学内容和教学方式上强调精讲多练。

专科学校在三个层次的教学过程中应有自己的特色, 要实用够用, 在相对较少的教学时数内, 切实贯彻精讲多练的原则。在计算机高级语言程序设计的教学中, C 语言是首选, 即把 C 语言作为首门高级语言在第二个层次进行讲授。

本书按照教育部关于高等工程类专科学校教学基本要求编写, 是直接基于计算机基本操作与使用知识的、面向初学程序设计语言的高等专科学校学生的教材。它突出了算法设计, 强调了结构化程序设计, 同时解决了 C 语言入门的问题, 给出了数据结构与程序设计的关系, 并进一步给出面向对象程序设计的基本知识。本教材具有实用性、先进性, 突出了专科特色和面向 C 语言初学者的特点, 做到了少而精。

本书编者都是多年从事高校计算机高级语言教学的副教授、讲师, 书中包含了编者多年的教学经验。本书对于专科学校 C 语言教学是一本很好的教材和很有价值的参考书, 对专科学校 C 语言教学提前迈上一个新台阶, 做出了成功的尝试。



1999 年 3 月 于哈尔滨

前 言

随着 21 世纪的即将来临,人类已经进入了一个飞速发展的信息化社会,计算机的普及和应用使学习使用计算机已成为一种文化时尚。

计算机的特点是高速、准确、自动地处理信息,它的运行是由人通过程序控制的。在学习了一定的计算机操作及使用知识后,学习用计算机程序设计语言编写程序,使计算机完成用户指定的任务,已成为学习者进一步掌握计算机的必然方向,这也正是现代科学技术人员、理工科大专院校学生学习计算机的宗旨。

C 语言是一种功能极强、效率极高的计算机程序设计语言。它原本是被高级软件人员用于编写操作系统及系统软件的,随着计算机的普及,全社会计算机文化教育水平的普遍提高及 C 语言功能和运用深度广度的提高, C 语言不仅可能,而且已经成为各领域技术人员学习程序设计语言的首选, C 语言的普及已成必然趋势。

本书作为 C 语言程序设计教材,正是基于这一趋势,按照教育部关于高等工程类专科学校教学基本要求编写的,是直接基于计算机基本操作与使用知识的、面向初学程序设计语言的高等专科学校学生的入门与提高教材。

本书内容共分十章。其中:第一章概述程序设计语言与程序设计概念、计算机算法、C 语言的发展、特点和 C 程序结构,是学习 C 语言的预备知识。第二章讲述 C 语言的数据类型、表达式和基本操作及位操作,这是构成 C 语言的基础。第三章讲述基本 C 语句、C 流程控制结构及结构化设计。第四章介绍函数和模块化设计及变量存储属性和编译预处理。第五、六、七章较详尽地描述了 C 语言的几种组合数据类型:数组、指针、结构体与共用体及相关的各部分内容。第八章介绍 C 语言的文件操作,包括文件的概念、文件系统、文件类型指针、文件的打开与关闭、文件的读写、文件的定位与出错检测。第九章讨论数据结构与程序设计关系、常用线性数据结构,给出几个典型 C 语言程序设计例子。第十章简单介绍面向对象程序设计,包括面向对象方法和 C++语言的基本概念、面向对象方法比结构化方法的优越性。附录: Turbo C 集成环境简介、C 语言常见错误分析、ASCII 字符编码表、运算符及其优先级、C 语言关键字、C 常用库函数等。其中第九、十章为选学内容,是考虑到计算机语言程序设计的发展和为学习者的进一步学习要求而设置的。

本书还参考了全国计算机等级考试二级 C 语言大纲,使广大在校生在完成课内学习内容的同时,兼顾等级考试的要求。

本书按照教育部关于计算机基础教育要面向 21 世纪、完成三个层次教学目标的要求,把计算机语言程序设计安排在第二层次,最好在计算机文化基础课程之后,使用本教材。

本书的例题和习题充分考虑了初学者和参加等级考试者,强调精讲多练,重点突出。读者应充分练习并上机实践,以期获得最好的学习效果。

本书也可以作为各类普通高等院校学生、各类计算机培训班学员的学习参考书。

参加本书编写的单位有黑龙江水利高等专科学校、哈尔滨工程高等专科学校、黑龙江

交通高等专科学校、黑龙江财政高等专科学校、哈尔滨理工大学工业技术学院、哈尔滨师范专科学校和黑龙江省社会科学院等。

本书由张松波、赵惠新主编，吴敏、张俊峰主审，吴亚春、孔庆彦副主编，韩玉敏、牛健、葛丽、王鑫、洪岩、刘永等参加编写。东北林业大学计算中心常健斌教授、哈尔滨医科大学计算中心安齐国教授、黑龙江财政高等专科学校张洪瀚教授对全书的编写给予很大帮助，并提出了宝贵意见。

本书的编写工作得到黑龙江高校计算机基础教育研究会各理事单位的大力帮助，在此表示深深的谢意。

由于编者水平有限，不足之处敬请读者批评指教。

编 者

2000年5月 于哈尔滨

目 录

第一章 概 述.....	1
1.1 计算机程序设计语言	1
1.2 程序设计的概念	3
1.3 计算机算法与算法的图形表示法	4
1.4 C 语言基本概念	14
习 题.....	19
第二章 数据类型、表达式和基本操作.....	20
2.1 数据的类型及其表示方法	20
2.2 常量与变量	25
2.3 运算符和表达式	28
2.4 数据类型的转换	33
2.5 位运算	34
2.6 输入、输出函数	39
习 题.....	42
第三章 基本 C 语句和流程控制	45
3.1 基本 C 语句	45
3.2 C 流程控制结构	47
3.3 选择型流程控制语句	48
3.4 循环型流程控制语句	55
习 题.....	66
第四章 函数和模块化程序设计	69
4.1 函数	69
4.2 变量的存储属性	75
4.3 编译预处理	86
习 题.....	89
第五章 数 组.....	92
5.1 一维数组	92
5.2 二维数组和多维数组	95
5.3 字符数组和字符串	103
习 题.....	108
第六章 指 针	109

6.2 指针与数组	115
6.3 指针作为函数的参数	121
6.4 多级指针	129
6.5 指针数组	130
6.6 指针与函数	133
习 题	137
第七章 结构体与共用体	139
7.1 结构体类型数据	139
7.2 共用体类型数据	143
7.3 枚举类型数据	148
7.4 用户定义类型	150
7.5 结构体数组	151
7.6 结构体与指针	156
7.7 函数之间结构体变量的数据传递	158
习 题	162
第八章 文件操作	164
8.1 文件概述	164
8.2 文件类型指针	166
8.3 文件的打开与关闭	166
8.4 文件的读写	168
8.5 文件的定位	174
8.6 出错检测	176
习 题	177
第九章 数据结构与程序设计	178
9.1 数据结构与程序设计关系	178
9.2 常用线性数据结构	179
9.3 图形编程	184
9.4 菜单程序设计	188
习 题	191
第十章 面向对象程序设计	192
10.1 面向对象程序设计和 C++语言的基本概念	192
10.2 从结构化编程到面向对象编程	194
10.3 C++在传统过程语言方面对 C 的扩展	195
10.4 类与对象	202
习 题	211
附录 I Turbo C 集成环境简介	212

附录Ⅱ C语言常见错误分析	219
附录Ⅲ ASCII 字符编码一览表	231
附录Ⅳ 关键字及其用途	232
附录Ⅴ 运算符的优先级别和结合方向	233
附录Ⅵ C 常用库函数	234
参 考 文 献	243

第一章 概 述

计算机已广泛应用于当今社会的各个领域，成为大众化的现代工具，计算机知识已成为逐步普及的一种社会文化。计算机是一种具有内部存储能力（记忆能力）、由程序自动控制（分析判断能力）的电子设备（电脑）。人们将需要计算机做的工作写成一定形式的指令，并把它们存储在计算机的内部存储器中，当人们给出命令之后，它就按指令操作顺序自动进行。人们把这种可以连续执行的一条条指令的集合称为“程序”。可以说，程序就是人与机器进行“对话”的“语言”序列，这里的语言就是我们常说的“计算机程序设计语言”，即用于编写计算机程序的语言。

本章即要介绍计算机程序设计语言、程序设计、算法的概念和 C 语言的初步知识。

1.1 计算机程序设计语言

1.1.1 程序设计语言的发展

电子计算机作为高速、精确、自动处理信息的工具，已经广泛用于科学计算、数据处理、自动控制、人工智能、计算机辅助设计、计算机辅助教育、办公自动化等各个领域。要使计算机懂得人的意图，按人的意图去工作，就需要人与计算机交流信息的工具——计算机语言。计算机语言已经历了由机器语言→汇编语言→高级语言的发展变化过程。

1. 机器语言

最早的机器语言是和计算机硬件设计同时产生的。计算机硬件能直接识别和处理由 0 和 1 组成的代码，例如，在 IBM PC 机上

00000011 11000001

是一条指令，作用是让计算机的两个寄存器 AX 与 CX 的值相加，其结果放在 AX 中。指令是用来告诉计算机执行某一特定的基本操作，例如，加、减、读、写、控制转移等。基本操作指令的集合就是机器语言，即指令系统。对于不同的机型，机器语言是不同的，因此机器语言程序可移植性差。机器语言是一组二进制编码的指令，难学难用，易错难改。

2. 汇编语言

为了克服机器语言的缺点，人们设计了汇编语言。在汇编语言中采用一些助记符代替机器码，助记符通常是英语单词的头几个字母，数值用十进制数或十六进制数，例如上述指令：00000011 11000001，汇编语言的形式为

ADD AX, CX

显然用汇编语言写程序比用机器语言容易得多，程序的效率及质量都有所提高。由于计算

机只能直接执行机器语言的程序，为此，人们设计出一种把汇编语言程序翻译成机器语言程序的专用程序——汇编程序，让计算机自动地完成翻译工作。汇编语言中的助记符指令与机器语言中的二进制代码指令基本上是一一对应的。因此，它们都是为适应机器要求而设计的，属于“低级”语言，与人类自然语言差别很大。程序员用它们编程时，不仅要考虑解题思路，还要熟悉机器的内部结构。这种编程的劳动强度仍然很大，非计算机专业人员学起来还是很困难。

3. 面向过程的高级语言

从第一台计算机 ENIAC 问世，又经过 10 年的努力，到了 1954 年出现了第一个高级语言——FORTRAN I。高级语言是由自然语言（英语）和数学公式构成的，用语句表述操作，一个语句对应着若干条指令，大大地提高了效率。用高级语言编写的程序不受计算机类型的约束，具有通用性。

用高级语言编写的程序要经过翻译才能被机器识别执行，翻译工作由专用程序完成，有编译和解释两种方式。编译方式：一次性把高级语言程序全部翻译成机器语言程序，让机器执行。解释方式：在执行源程序时，把源程序调入内存，然后翻译一句，执行一句。

高级语言不再是面向具体的机器，而是面向解题的过程。即用人们易于理解的形式来表示解题的过程。使用高级语言编写程序时人们不必熟悉计算机内部的具体构造和熟记机器指令，而把主要精力放在算法描述上面。因此，面向过程的高级语言又称为算法语言。使用高级语言编程大大地降低编程过程的劳动强度，提高了编程的效率。

面向过程的高级语言的出现是计算机发展道路上的一个里程碑。它使计算机开始成为大众化工具。据统计，全世界已有成千种高级语言，其中使用较多的有近百种。在这些语言中，C 语言以其高效、灵活、功能丰富、表达力强、移植性好而受青睐，被称为近二十年来在计算机程序设计实践中做出了重大贡献的一种语言。

4. 面向对象的高级语言

使用过程化的语言解题，人们首先理解要求我们“做什么”，然后去构造“怎么做”的解题过程。80 年代初，有人提出了“对象”和“面向对象的程序设计”，其基本思想是：通过建立对象，并通过对这些对象的组合来创建具体的应用。所谓对象是指在传统的数据结构中加入处理这些数据的过程和函数所构成的实体。面向对象的高级语言是非过程化的高级语言，用它编程序不但无须考虑机器的构造，而且无须去构造“怎么做”的解题过程，用户只需要告诉计算机“做什么”，机器便会替你完成解题过程。面向对象的高级语言解题效率更高，使用更为简便，更适合于非计算机专业人员使用。但是，它的运行效率和灵活性都不如过程化的语言，而且，目前所使用的大多数程序都是过程化语言编写的。因此，学习用过程化的高级语言进行程序设计仍是计算机应用人员的一项基本任务。

本书将在学习 C 语言数据和数据结构、语句和函数、文件概念基础上进一步介绍面向对象的 C++ 的程序设计思想。

1.1.1.2 程序设计语言的支持环境

每一种计算机都配备操作系统。它是各种软件的核心与基础。所有其它程序的运行都要在操作系统的控制下进行。操作系统的作用是有效地组织和利用计算机的软、硬件资源，

使各种程序能在操作系统的管理下协调工作。操作系统的类型有许多种，例如有分时操作系统、多道作业操作系统等。图 1.1 表示高级语言源程序必须经过编译系统处理，在操作系统控制下才能执行。

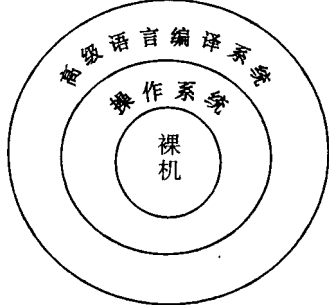


图 1.1

操作系统的功能主要包括：CPU 管理、存储管理、文件管理、设备管理、作业管理等。目前流行的操作系统有 UNIX、MS-DOS、Windows 等。

使用高级语言编写源程序时应当注意程序的运行环境，要充分运用所用的计算机系统提供的软件支持（例如操作系统的功能），并了解源程序在所用的操作系统管理下是如何工作的。例如在不同的操作系统下编译和运用 C 语言程序的方式和命令是不同的。

1.1.1.3 源程序的编辑、编译、连接、运行

采用编译方式的高级语言，从编写好一个程序到完成运行一般经过以下几个步骤。

1. 编辑

编辑的对象是源程序，它是以 ASCII 代码的形式存在的，计算机不能执行。编辑包括：源程序输入、修改、存盘。MS-DOS 下的 EDIT 是最常用的编辑软件。

2. 编译

编译就是将已编辑好的源程序翻译成二进制的目标代码。编译后得到的二进制代码在 MS-DOS 下是后缀为“.obj”文件。

3. 连接

经编译得到的二进制代码还不能直接执行，必须把各个模块的目标代码与系统提供的标准模块（如 C 语言的标准函数库）连接处理后，得到具有绝对地址的可执行文件，才能运行。可执行文件在 MS-DOS 下以“.exe”为后缀。

4. 执行

经过编译和连接的可执行的目标文件，只有在操作系统的支持和管理下才能执行。

现在，一般使用“集成化”开发环境，将编辑、编译、连接、调试工具集于一身（例如 Turbo C）。本书附录中较详细地介绍了 Turbo C “集成化”开发环境。

1.2 程序设计的概念

通俗地讲，程序设计就是用计算机语言编写程序。而程序在简单情况下也可以用下面的公式来概括：

算法 + 数据结构 = 程序

就是说，要编写程序必须要研究如何组织数据，如何进行操作。数据是程序操作的对象。因此首先应确定要对其施加操作的是什么样的数据，然后设计出操作的具体步骤，操作步

骤就是算法。数据结构和算法是程序的两大要素。就像加工零件一样，数据结构相当于原料，算法就是操作步骤。不同的数据结构，算法是不同的。例如，对 10 个数排序，用数组处理和不用数组处理，算法差别很大，选择合适的数据结构可以降低算法的复杂程度。

对于规模较大的程序，还要考虑如何组织程序设计，即要研究用某种程序设计方法编写程序。例如把一个大的任务先分解为若干较小的子任务（即模块），再分别逐一进行程序设计，从而降低了程序设计的复杂性。这是当前流行的结构化程序设计方法，目的是使程序设计工作规范化，提高程序的可读性。结构化程序设计方法的要点是：自顶向下；逐步细化；模块化。每一模块都由具有良好结构特征的基本结构组成。

此外，一个程序必须用一种计算机语言来编写的，要有必要的环境支持。例如，要用操作系统提供的一些命令和系统软件。因此，我们认为：

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境。

即程序设计人员应当具备以上四个方面的知识。本书就是按照这四个方面组织的。

在四个方面中，数据结构包含丰富的内容，本书在第九章讲述数据结构与程序设计的关系及常用数据结构举例。本章重点是讨论算法，根据算法编写程序，在本书中所有的程序都是按结构化程序设计方法编写的。

程序设计的全过程应该包括：分析问题、划分模块、建立模型（物理模型和数学模型）、选择数据结构和算法、描述算法（例如画出流程图）、编程序、上机调试程序、分析运行结果、整理结果，最后，写出程序说明书等文档。

1.3 计算机算法与算法的图形表示法

简单地说，算法就是解题的方法和步骤。程序设计就是用计算机语言把一个问题的算法描述出来。从上节程序设计的全过程可知程序设计的两个重要因素是算法设计能力和学会一门计算机语言，如 C 语言。算法设计训练对提高解题能力有重要意义。

描述算法的工具具有图形和伪代码两种表示法，它们是算法的逻辑表达，是由算法到程序的桥梁。学画算法图比学习计算机语言容易，图形直观而且包含信息量大。所以，本书先介绍计算机算法的表示法，然后再讲 C 语言程序设计。

1.3.1 计算机算法与特征

1. 算法——计算机算法

广义地说，为解决一个问题而采取的方法和步骤称为“算法”（Algorithm）。包括设计算法和实现算法两个部分。本书所关心的算法是用计算机能实现的算法，所以称之为计算机算法，简称算法。下面举例说明。

例 1.1 30 名学生参加一次考试，要求把不及格学生的学号和成绩打印出来。

算法分析：设 N 为学号，G 为成绩，I 为计数器，算法可表示如下：

第 1 步：I ← 1

第 2 步：输入第 I 个学生的 N，G

第 3 步：如果 G < 60 则打印 N 和 G，否则不打印

第4步: $I \leftarrow I + 1$

第5步: 直到 $I > 30$ 转至第6步, 否则转至第2步

第6步: 结束

这个算法的第2步到第5步是一个重复操作, 也叫循环, 在循环中又包含了一个条件打印输出, 计算机高级语言都有实现循环的语句和实现条件打印的条件语句。因此, 上述算法也是计算机算法。“ $\leftarrow$ ”表示把右边表达式的值赋给左边的变量, 称为赋值号。

2. 计算机算法的特征

一个计算机能实现的算法具有以下特征:

(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。例1.1虽包含重复操作, 但是有条件的。当然, “有穷性”应根据需要规定在合理范围之内。

(2)确定性。算法中的每一步骤含义都应当是确定的、唯一的, 而不应当是模棱两可的。例如, 例1.1中第3步, 改为“如果 G 在60左右则打印 N 和 G , 否则不打印。”因为没有给出准确的 G 的范围, 因而计算机就无法执行。

(3)有零个或多个输入。所谓输入是指在执行算法过程中需要从外界取得必要的信息。例如, 例1.1的第2步要执行30多次输入。

(4)有一个或多个输出。实现算法的目的是求解, 因此必须把答案输出, 至少一次输出。例1.1要输出所有不及格学生的学号和成绩, 从逻辑上说, 它有一个或多个输出。

(5)有效性。算法中每一步都应是有效地操作。例如, 若 $A \neq 0$, $B = 0$, 执行 A/B 就是无效的操作, 这样的操作出现在程序中, 计算机输出一个错误信息, 并中断程序的执行。

1.3.2 算法的图形表示法

图形表示法是二维的, 它能容纳更多的信息, 更直观、更便于交流。

1. 传统流程图

1940年, H.H.Goldstone等提出了流程图表示法。例1.1的流程图如图1.2所示。

这种流程图传播得比较广, 时间也早, 又称传统流程图。传统流程图常用的图形符号如图1.3所示。下面对其作简要说明:

输入输出框: 平行四边形表示数据输入输出, 其中可注明数据名称、来源、用途或其他的文字说明。

处理框: 矩形表示各种处理功能。例如, 执行一个或一组特定的操作, 从而使信息的值、信息形式或所在位置发生变化。矩形内注明处理名称或其简要功能。

判断框: 菱形表示判断。菱形内可注明判断的条件。它只有一个入口, 但可以有若干个可供选择的出口, 在对定义的判断条件求值后, 有一个且仅有一个出口被选择。求值结果可在表示出口路径的流线附近写出, T 表示真, F 表示假。

流向线: 直线表示执行的流程。当流程自上向下时, 流向线可不带箭头, 其它情况应加箭头表示流程。

连接点: 圆形表示转向外部环境或从外部环境转入的端点符。

起止框: 扁圆形表示程序流程的起止点。

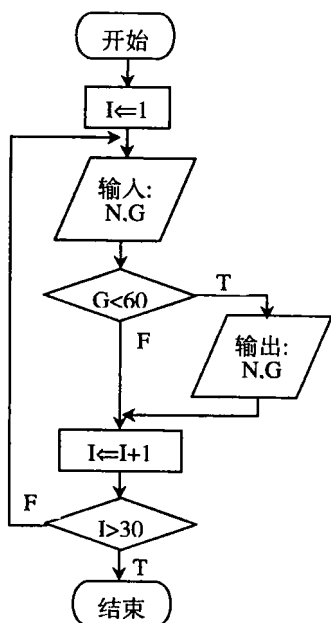


图 1.2 例 1.1 的流程图

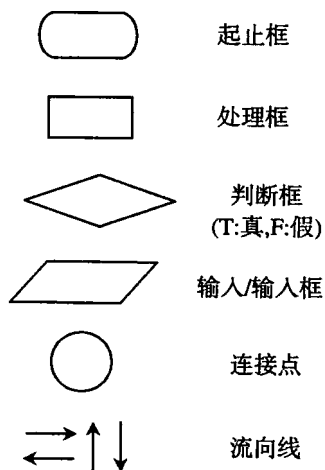


图 1.3 流程图常用符号

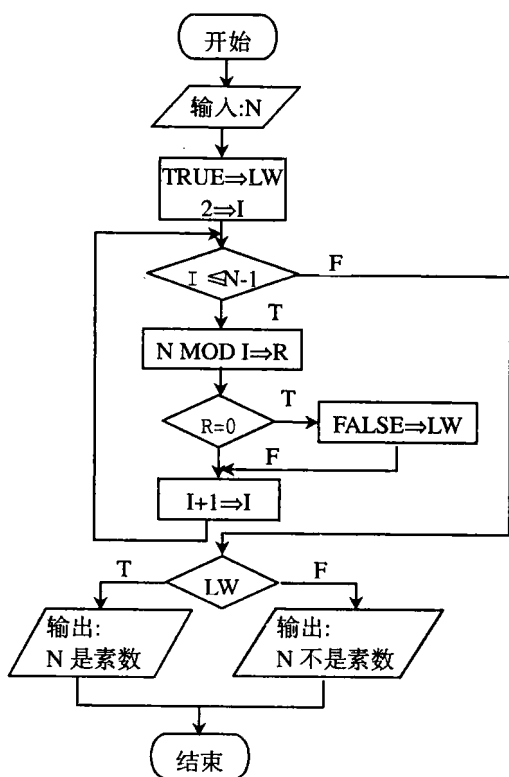


图 1.4 例 1.2 的结构化流程图

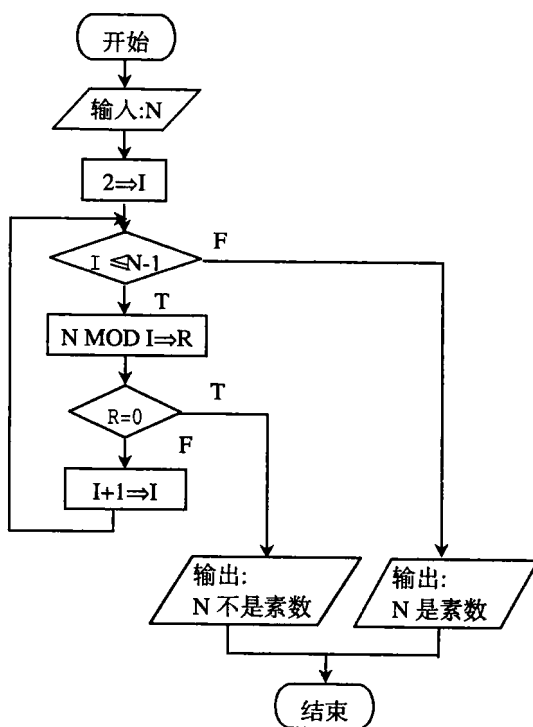


图 1.5 例 1.2 的非结构化流程图

传统流程图有一个流向线(→)(高级语言中常用 GOTO 语句表示它),它使得图形绘制有很大的任意性,请看如下例题。

例 1.2 任意给定一个自然数 $N(N>1)$, 判断 N 是否是素数。

算法分析: 素数的定义是: 如果一个自然数 N 只能被 1 和 N 整除, 不能被其它整数整除时, 称 N 为素数。由此设计出两种算法, 如图 1.4 和图 1.5 所示。前者结构清晰些。

1966 年 G.Jacopini 等证明, 任何(单入口, 单出口)程序均可用三种基本控制结构(顺序、选择、循环)复合、叠加组成。这一证明为结构化程序设计奠定了理论基础。三种基本控制结构如图 1.6 所示。其中, 块可以是一个不引起控制转移的操作, 也可以是一个空块, 或者是三个基本控制结构之一。后来进一步规定, 结构化程序应满足以下条件:

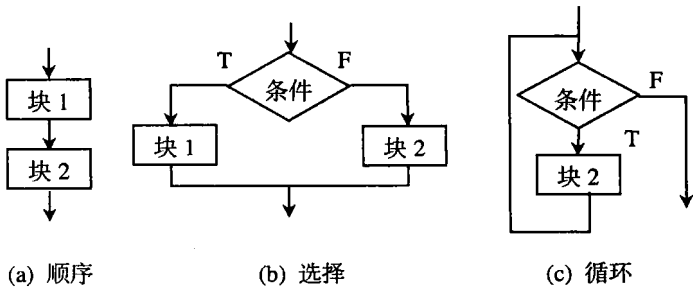


图 1.6 三种基本控制结构

(1)单入口, 单出口 (对于每个基本结构)。

(2)对于每一个结点 (语句或块), 都应有一个从入口到出口的通路通过它, 即不允许有永远执行不到的语句。

(3)不应有死循环 (无休止的循环)。

按照上述条件, 图 1.4 从上到下: 输入→赋值→循环→选择→结束是顺序结构, 循环内嵌套了赋值→选择→赋值的顺序结构, 为结构化流程图。图 1.5 的循环有两个出口, 为非结构化流程图。

上述(2)和(3)两个条件对非结构化程序也是需要满足的条件。所以, 结构化程序的特点是: 组成它的基本结构都满足“单入口、单出口”的条件。符合这样条件的算法叫结构化算法, 符合这样条件的流程图叫结构化流程图。把能够严格地表述三种基本结构的计算机语言叫结构化语言, 如 C 语言的结构清晰、易于阅读、易于修改。

2. N-S 图

针对结构化程序设计, I.Nassi 等于 1973 年推出了 N-S 图, 也称结构化盒图。按 N-S 图规定画出的算法图都是结构化的。包括 6 个基本图形, 如图 1.7 所示。

N-S 图上边框表示入口, 下边框表示出口, 它的每个基本图形都符合“单入口, 单出口”的要求。而且和结构化程序设计的基本控制结构的约定一样, 其中“块”或是一个不引起控制转移的操作, 或是一个空块, 或是 6 个基本图形之一。图 1.7a、b、d 是基本控制结构, 而 c、e、f 是基本控制结构组合图的简化图 具有更高的集成度。本书将用 N-S 图表示算法, 下面作进一步说明。

(1)顺序结构 用图 1.7 (a) 形式表示。它 对应。块 1 和块 2 两个框组成一个顺序结构。

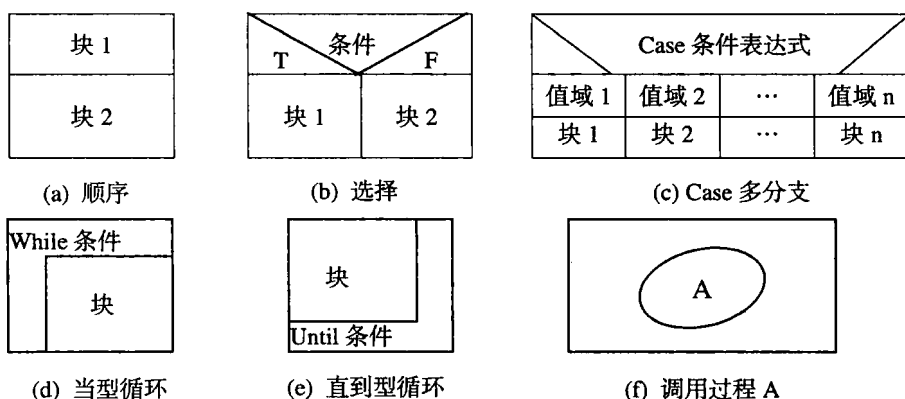


图 1.7 N-S 图的基本图形

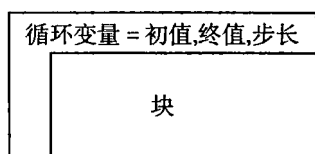
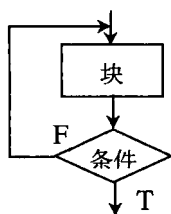
(2)选择结构 用图 1.7 (b) 形式表示。它与图 1.6 (b) 对应。当“条件”成立时执行块 1 操作，否则执行块 2 操作。在执行一次选择结构时，块 1 和块 2 只有一个块被执行。

(3)当型循环结构 用图 1.7 (d) 形式表示。它与图 1.6 (c) 对应。当“条件”成立时，重复执行块操作，“条件”不成立时止。特点是：先判断，后执行块。

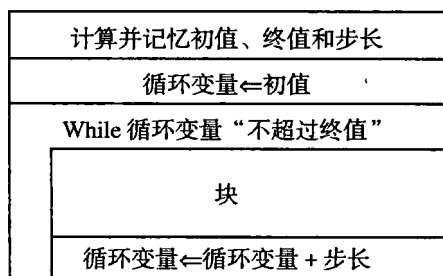
(4)直到型循环结构 用图 1.7 (e) 形式表示。图 1.8 表示此结构的执行过程，先执行块操作，再判断“条件”，若不成立，则重复执行块操作，直到“条件”成立为止。特点是：先执行块，后判断。

(5)Case 多分支结构 用图 1.7 (c) 形式表示。它的执行过程是：①计算“条件表达式”的值；②检查该值是否在值域 1 内，如果该值在值域 1 内，则执行块 1，结束；否则检查是否在值域 2 内，如果在值域 2 内，则执行块 2，结束；否则再检查是否在值域 3 内……。

(6)调用过程 用图 1.7 (f) 形式表示。它主要用于处理复杂算法。当算法用一个 N-S 图无法表示时，可以用它表示逐步细化的步骤。



(a) 计数型循环



(b) 执行过程

图 1.8 直到型循环执行过程

图 1.9 计数型循环和执行过程

此外，还有计数型循环结构，即循环次数在进入循环之前是已知的，表示形式和执行过程如图 1.9 所示。计数循环是当型循环的特殊形式。例 1.1 的 N-S 图如图 1.10 所示。

1.3.3 基础算法

计算机程序的核心和基础是

算法设计不同于数学上的公理系统思维方法，