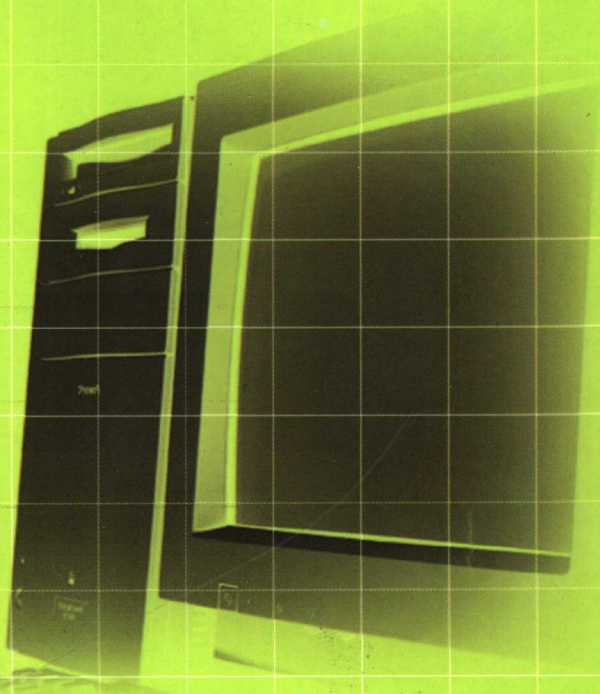


C 全国职业教育计算机类系列教材

C语言 程序设计

YUYAN 邢素萍 主编

CHENGXUSHEJI



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

全国职业教育计算机类系列教材

C 语言程序设计

主 编	邢素萍	左晓英
副主编	李新春	邵 菊
参 编	王 梅	王继红

东南大学出版社

内 容 提 要

本书详细地介绍了C语言的数据类型及其运算、输入输出、选择结构、循环结构、数组、函数、指针、编译预处理、结构体和共用体、位运算及文件操作等知识。

本书在内容安排与编写方面,力求条理清晰、层次分明、通俗易懂、方便教学。学生通过学习,能够掌握C语言的基本知识,为进一步学习C++语言,同时也为参加全国计算机等级考试中的二级C语言考试奠定扎实的基础。

本书可作为高职高专或中职业院校的计算机程序设计教材,也可选择作为全国计算机等级考试的教材,以及供程序设计爱好者自学使用。

图书在版编目(CIP)数据

C语言程序设计/邢素萍,左晓英主编. — 南京:东南大学出版社,2005.1

ISBN 7-81089-827-2

I. C... II. 邢... III. C语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆CIP数据核字(2004)第131134号

东南大学出版社出版发行
(南京四牌楼2号 邮编210096)

出版人:宋增民

江苏省新华书店经销 扬州市文丰印刷制品有限公司印刷

开本:787mm×1092mm 1/16 印张:14 字数:350千字

2005年1月第1版 2005年1月第1次印刷

印数:1—4 000 册 定价:21.00元

(凡有印装质量问题,可直接向发行部调换。电话:025-83795801)

出版说明

全国职业教育计算机专业建设研讨会于2004年7月18日在湖北三峡职业技术学院召开,来自上海、江苏、山西、辽宁、贵州、黑龙江等40多位职业技术学院的代表参加了会议。

在本次会议上,与会专家学者对目前职业教育的现状进行了深刻地分析,特别对计算机专业建设提出了独到的见解。他们一致认为:计算机专业建设要与教学改革相结合,以市场需求为导向,以教材建设为基础。因此,会议决定为配合计算机专业建设,编写一套适合职业教育的计算机系列教材,要求突出职业特点,有创新思想,以“考证”为切入点,加强实践环节。

根据各校计算机专业建设和课程设置情况,本次会议由全国职业教育计算机类教材建设委员会秘书长孔繁华组织各院校计算机专业教师确定了首批教材建设的选题,以后还将随着专业建设的深入及计算机技术的发展,逐步形成一套完善的、切合实际的计算机职业教育系列教材。

全国职业教育电子信息类教材编委会总要求:坚决贯彻职业教育的要求,即基础适度够用、加强实践环节、突出职能教育,把握职业教育电子信息类专业课程建设的特点;立足当前学生现状,面向用人单位(市场),打破条条框框,少一些理论,多一些技能教育;采取逆向思维的方式编写,即从市场需要什么技能来决定学生需要什么知识结构,并由此决定编写什么教材。

全国职业教育电子信息类教材编委会会员单位:

南京信息职业技术学院

本溪电子工业学校

扬州电子信息学校

河南信息工程学校

大连电子工业学校

黑龙江信息技术职业学院

本溪财贸学校

山西工程职业技术学院

四川省电子工业学校

锦州铁路运输学校

内蒙古电子信息职业技术学院

江苏海事职业技术学院

黑龙江农业经济职业学院

南通纺织职业技术学院

湖北三峡职业技术学院

长沙市电子工业学校

山西综合职业技术学院

北京信息职业技术学院

福建省电子工业学校

山西省邮电学校

新疆机械电子职业技术学院

山东信息职业技术学院

哈尔滨机电工程学校

上海机电工业学校

贵州省电子工业学校

南京交通职业技术学院

扬州职业大学

南通航运职业技术学院

全国职业教育电子信息类教材编委会

2005年1月

前 言

本书是全国职业教育电子信息类教材编委会组织编写的《全国职业教育计算机类系列教材》之一,经过编委会组织全国 30 多所高职、中职院校的计算机专业教师认真研讨,并根据当前我国职业院校计算机专业教学的实际编写而成。

C 语言是目前世界上最优秀的计算机程序设计语言之一。C 语言以它丰富的数据类型、模块化的数据结构和流程控制、高效的语言效率、良好的人机界面和强大的接口操作能力等特点,博得了计算机程序设计人员的青睐,被称为“兼备高级语言和汇编语言优点的中级语言”或“工程设计语言”。由于它在开发 Unix 系统中的功绩,人们又称其为“具有系统开发能力的语言”。

目前,C 语言是我国职业院校计算机类专业的专业课之一,也是国家计算机等级考试二级考试内容。

本书在内容安排与编写方面,力求条理清晰、层次分明、通俗易懂、方便教学。学生通过学习,能够掌握 C 语言的基本知识,为进一步学习 C++ 语言,同时也为参加全国计算机等级考试中的二级 C 语言考试奠定扎实的基础。

本书参考教学时数为 70 学时,其中理论教学为 42 学时,实验实习为 28 学时。

本书由黑龙江农业经济职业学院邢素萍副教授、黑龙江信息技术职业学院左晓英副教授任主编;由南京交通职业技术学院李新春、湖北三峡职业技术学院邵菊任副主编,江苏海事职业技术学院王梅、黑龙江信息技术职业学院王继红参编。其中第 1、8、9 章由邢素萍编写,第 2、3 章由王梅编写,第 4、5 章由李新春编写,第 6 章由王继红编写,第 7、12 章由左晓英编写,第 10、11 章由邵菊编写。全书由邢素萍统稿。

在本书编写过程中,得到了全国职业教育电子信息类教材编委会、黑龙江农业经济职业学院、黑龙江信息技术职业学院、湖北三峡职业技术学院、南京交通职业技术学院、江苏海事职业技术学院等学院领导的关怀指导以及同行教师的大力帮助,在此一并表示衷心的感谢。

由于时间仓促及编者水平有限,书中错误之处在所难免,恳请读者批评指正。

编 者

2005 年 1 月

目 录

1 C 语言程序设计概述	(1)
1.1 程序设计的基本概念	(1)
1.1.1 计算机语言	(1)
1.1.2 程序概述	(2)
1.1.3 算法及其描述	(3)
1.1.4 结构化程序设计	(6)
1.1.5 C 语言的特点	(7)
1.2 C 语言程序的结构和书写格式	(8)
1.2.1 C 语言程序的结构	(8)
1.2.2 C 语言程序的书写格式	(9)
1.3 C 语言程序运行环境简介	(9)
1.3.1 Turbo C 2.0 的安装	(9)
1.3.2 Turbo C 2.0 的启动、退出与设置	(10)
1.3.3 Turbo C 2.0 的使用	(11)
习 题	(12)
实 验	(13)
2 数据类型和表达式	(14)
2.1 标识符、常量、变量	(14)
2.1.1 标识符	(14)
2.1.2 常量	(14)
2.1.3 变量	(15)
2.2 基本数据类型	(15)
2.2.1 整型数据	(15)
2.2.2 实型数据	(16)
2.2.3 字符型数据	(17)
2.2.4 变量初始化	(18)
2.3 运算符和表达式	(18)
2.3.1 算术运算符和算术表达式	(18)
2.3.2 赋值运算符和赋值表达式	(19)
2.3.3 关系运算符和关系表达式	(19)
2.3.4 逻辑运算符和逻辑表达式	(20)
2.3.5 自增运算符、自减运算符及表达式	(21)
2.3.6 逗号运算符和逗号表达式	(22)
2.3.7 不同类型数据间的转换	(22)
习 题	(23)

实 验	(24)
3 简单 C 语言程序设计	(26)
3.1 顺序结构程序执行流程.....	(26)
3.2 C 语言的基本语句	(26)
3.2.1 赋值语句	(26)
3.2.2 表达式语句	(27)
3.2.3 复合语句	(27)
3.2.4 空语句	(27)
3.3 数据输出函数.....	(27)
3.3.1 printf()函数	(28)
3.3.2 putchar()函数	(29)
3.3.3 puts()函数	(30)
3.4 数据输入函数.....	(30)
3.4.1 scanf()函数	(30)
3.4.2 getchar()函数	(31)
3.4.3 gets()函数	(32)
习 题	(32)
实 验	(33)
4 选择结构程序设计	(35)
4.1 if 语句	(35)
4.1.1 if 语句的格式	(35)
4.1.2 if 语句的嵌套	(37)
4.1.3 条件运算符	(38)
4.2 switch 语句	(40)
4.2.1 switch 语句的一般格式	(40)
4.2.2 switch 语句的执行过程	(40)
4.3 程序举例.....	(41)
习 题	(44)
5 循环结构程序设计	(47)
5.1 while 语句	(47)
5.2 do-while 语句	(48)
5.3 for 语句	(49)
5.4 循环的嵌套.....	(51)
5.5 break 语句和 continue 语句	(53)
5.5.1 break 语句.....	(53)
5.5.2 continue 语句.....	(54)
5.6 goto 语句	(55)
5.7 程序举例.....	(55)
习 题	(59)

6 数组	(61)
6.1 一维数组	(61)
6.1.1 一维数组的定义	(61)
6.1.2 数组元素的引用与一维数组的初始化	(61)
6.1.3 程序应用举例	(62)
6.2 二维数组	(65)
6.2.1 二维数组的定义	(65)
6.2.2 二维数组的初始化与数组元素的引用	(66)
6.2.3 二维数组应用举例	(66)
6.3 字符数组	(68)
6.3.1 字符数组的定义	(68)
6.3.2 字符数组的初始化	(68)
6.3.3 字符数组的输入输出	(69)
6.3.4 字符串处理函数	(69)
6.3.5 字符数组应用举例	(72)
6.4 数组应用举例	(75)
习 题	(79)
实 验	(82)
7 函数	(84)
7.1 库函数	(84)
7.2 函数的定义和调用	(87)
7.2.1 函数的定义	(87)
7.2.2 函数的调用	(88)
7.3 函数的返回值及其类型	(89)
7.4 函数调用时参数间的传递	(90)
7.4.1 将变量、常量、数组元素作为参数的传递	(90)
7.4.2 将数组名作为参数的传递	(91)
7.5 函数说明	(91)
7.6 函数的嵌套调用	(93)
7.7 函数的递归调用	(93)
7.8 作用域和存储类型	(95)
7.8.1 变量的作用域	(95)
7.8.2 变量的存储类型	(98)
7.9 内部函数和外部函数	(101)
7.9.1 内部函数	(101)
7.9.2 外部函数	(101)
7.10 程序举例	(102)
习 题	(106)
实 验	(110)

8 编译预处理	(112)
8.1 宏定义	(112)
8.1.1 无参宏定义	(112)
8.1.2 带参宏定义	(115)
8.2 文件包含	(119)
习 题	(120)
9 指针和指针变量	(123)
9.1 指针和指针变量的概念	(123)
9.2 指针变量的定义和对指针变量的操作	(124)
9.2.1 指针变量的定义与指针变量的赋值	(124)
9.2.2 对指针变量的操作	(125)
9.2.3 指针变量作为函数参数	(127)
9.3 数组的指针及利用指针数组进行的操作	(129)
9.3.1 通过指针引用数组元素	(129)
9.3.2 二维数组和指针	(133)
9.3.3 二维数组名和指针数组作为实参	(138)
9.4 利用字符串的指针对字符串进行操作	(139)
9.4.1 字符串的表示和引用	(139)
9.4.2 字符串指针作函数参数	(142)
9.5 利用指向函数的指针变量调用函数	(143)
9.6 返回指针值的函数	(144)
9.7 主函数 main() 的形参	(146)
9.8 指向指针的指针变量	(147)
9.9 函数的指针和指向函数的指针变量	(148)
9.10 指针内容程序举例	(150)
习 题	(155)
10 结构体和共用体	(164)
10.1 结构体	(164)
10.1.1 结构体类型的定义	(164)
10.1.2 结构体变量的定义、引用和初始化	(165)
10.1.3 结构体数组	(168)
10.1.4 结构体类型的指针变量	(169)
10.1.5 结构体的指针作为函数的参数	(171)
10.1.6 用指针处理链表	(172)
10.2 共用体	(181)
10.2.1 共用体的定义	(182)
10.2.2 共用体变量的定义和引用	(182)
10.2.3 共用体类型数据的特点	(184)
10.3 枚举类型	(185)
10.3.1 枚举类型的定义和枚举变量的说明	(185)

10.3.2 枚举变量的赋值和使用	(186)
习 题	(187)
实 验	(188)
11 位运算	(189)
11.1 位运算符	(189)
11.1.1 按位取反	(189)
11.1.2 移位	(190)
11.1.3 按位与	(191)
11.1.4 按位或	(191)
11.1.5 按位异或	(192)
11.2 位段	(192)
11.2.1 位段的定义和位段变量的说明	(192)
11.2.2 位段的使用	(194)
习 题	(195)
实 验	(195)
12 文件操作	(196)
12.1 文件概述	(196)
12.1.1 文件与文件名	(196)
12.1.2 文件的分类	(196)
12.1.3 读文件与写文件	(197)
12.1.4 构成文件的基本单元与流式文件	(197)
12.1.5 文件类型 FILE	(197)
12.1.6 ANSI C 语言的缓冲文件系统	(197)
12.2 文件指针	(197)
12.3 打开和关闭文件	(198)
12.3.1 文件的打开——fopen() 函数	(198)
12.3.2 文件的关闭——fclose() 函数	(200)
12.4 常用文件操作函数	(200)
12.4.1 文件的输入输出	(200)
12.4.2 缓冲文件的定位函数	(206)
12.4.3 文件检测函数	(208)
习 题	(208)
实 验	(210)
参考文献	(212)

1 C 语言程序设计概述

1.1 程序设计的基本概念

1.1.1 计算机语言

人们控制计算机进行科学计算、信息处理主要依靠“计算机语言”。这样的计算机语言是由计算机专家设计的。最初的计算机语言是由类似二进制符号组成的机器语言,不同型号计算机的机器语言有所不同。这样的“语言”只是由少数专家设计和使用。在 20 世纪 50 年代,随着计算机数量的增多,专家们设计出了一种利用其他人容易理解的符号(英文、数字等)就能够编写的“汇编语言”。它的出现是计算机语言的重大进步。至今,汇编语言在工业控制和计算机硬件驱动程序设计方面仍然发挥着独到的作用。到了 60 年代,计算机专家又设计出了“计算机高级语言”,它是借用人类普遍使用的语言(英语)来编写计算机运行程序的语言规范。第一种高级语言是 BASIC 语言(具有简单易学的长处),后来又有了 FORTRAN(用于科学计算)、ALGOL(具有与硬件无关的特点)、COBOL(用于商务数据处理)、PASCAL(结构严谨的教学语言)、C 语言(运行效率高、速度快的计算机语言)等等。高级语言的出现具有划时代的意义,为普通人学习和掌握计算机程序设计奠定了基础,使计算机应用的普及成为可能。

随着计算机硬件技术的飞速发展,到了 20 世纪 90 年代,IBM 公司和 Microsoft 公司推出了可视化的操作系统 OS/2、Windows95/98/2000/XP。伴随出现了 Visual Basic、Delphi、Visual C++、C++ Builder、Java 等几十种面向对象的程序设计语言。总而言之,计算机语言是多种多样的,其功能及应用领域也不尽相同,同一种语言有多种版本,版本越新功能越强,同时可能对计算机硬件的要求更高一些。目前,在计算机语言的大家庭里,各种不同的语言在不同的应用领域发挥着各自的独特作用,成为了人类文化花园的一片奇葩。

在诸多的计算机语言中,C 语言具有特殊性。在 C 语言出现之前,对计算机操作系统的编写主要使用汇编语言。但是,汇编语言的可读性和可移植性很差,那时已有的高级语言又难以对计算机硬件访问和进行位操作,即不能用于设计开发计算机操作系统,而当时人们迫切需要一种能够开发操作系统、又具备高级语言特性的新的计算机语言。在这样的背景下,C 语言应运而生。

1963 年,英国剑桥大学的计算机专家们在 ALGOL 60 语言基础上进行了改进,推出了 CPL(Combined Programming Language)语言。1967 年,剑桥大学的 Martin Richards 对 CPL 进行了简化处理,形成了 BCPL(Basic CPL)语言。1970 年,美国贝尔实验室的 Ken Thompson 对 BCPL 语言进行了进一步的简化,开发了对硬件访问能力较强且应用较简单的 B 语言,并用它设计了第一个 Unix 操作系统。到了 1972 年,贝尔实验室的 D. M. Ritchie 进一步改进了 B 语言的一些不足,推出了 C 语言的第一个版本。C 语言的出现,对计算机操作系统和应

用软件的设计开发,起到了空前的推动作用。正因为如此,一些著名的计算机软件公司,如 Microsoft 和 Borland 纷纷推出了自己的 C 语言版本,不断地丰富 C 语言的功能。

国内外较为流行的 C 语言版本有:Microsoft C、Turbo C、Borland C 等等。后来在 C 语言的基础之上,又有了 C++ 语言以及面向对象的 C++ 语言。目前,计算机程序设计业内流行着“真正的程序员使用 VC++”这样一句话。

虽然面向对象的 C++ 语言已成为计算机软件设计开发的主流工具,但是,非面向对象的 C 语言仍然可用于科学计算、工业控制,而且它本身是 C++ 语言的基础,所以它仍然是目前具有很大应用价值的计算机语言。

1.1.2 程序概述

1) 程序的概念

如同告诉别人做一件事情要用语言一样,控制计算机按照人们的意图去工作,完成一系列的运算也要使用计算机语言。所谓“程序”就是人们用计算机语言编写的、计算机能够识别的操作命令序列。计算机运行不同的程序会产生不同的结果,要想使计算机完成某种特定的工作,就要设计相应的计算机程序。对于初学者,先要学习某种计算机语言,掌握了计算机语言(如 C 语言)的编写规范之后就可以由简单到复杂地设计计算机程序。

就任何一个程序而言,它通常由一个或一个以上的语句构成。这些语句必须是计算机能够识别并执行的,不能有丝毫差错,其先后执行的逻辑顺序是清楚的。

程序是一个计算机软件的主体或核心部分。

2) 程序设计的步骤

程序设计是编写某一特定程序的过程,C 语言程序的设计通常有下列步骤:

- (1) 确定待解决问题的计算过程或处理方法。
- (2) 建立数学模型,即将实际的问题转变为一个数学问题。
- (3) 编制程序框图,确定程序结构。
- (4) 编写程序,人工编写的程序称为“源程序代码”。
- (5) 上机编辑、调试(包括编译和连接),消除语法错误。
- (6) 如果调试正确则生成目标程序和最终的可执行文件。C 语言的目标程序是可执行的 EXE 文件,通常在文本化的 DOS 环境下运行。

对于一个大型的程序设计工作,还要应用软件工程的方法来进行。由系统分析员根据问题的任务情况确定项目的实施计划;再确定各阶段的任务和费用情况,明确参与设计人员的工作职责;提出项目的需求分析报告;确定项目的软、硬件环境,各个题目的设计要求、数据结构;然后进行项目的概要设计,对项目的大致结构和数据传递,即数据流进行设计,并完成项目中的各模块任务及其关系的确立;在概要设计的基础上对各个模块的数据输入输出和流程进行详细的设计;有了详细设计方案,程序员即可编写源程序代码,源程序代码编写完成即可进行上机调试。为保证程序的正确性和可靠性,需要制定一个完整的软件测试计划并按计划进行测试,以保证软件的各项功能模块功能正确。通过功能测试后要完成测试报告。至此,软件的程序设计工作基本完成。随后,还要编写软件项目开发总结和软件使用说明书。

3) 程序的结构

只有一条语句的应用程序几乎是没的,绝大多数程序含有多条语句。人们为了简化程

序代码,提高程序编写和运行效率,要尽可能用少量的语句来完成程序的编写。而在构造程序设计语言时,计算机专家就已经设置好了不同程序结构的语句架构:顺序结构、选择(分支)结构和循环结构。

(1) 顺序结构 总体来看,任何程序都是按照源程序代码从前往后的顺序来运行的,这就是所谓的顺序结构。

(2) 选择(分支)结构 所谓选择(分支)结构是指:当程序运行到某一个条件判断语句时,判断某一个条件是否成立,若条件成立则执行一个程序段;若条件不成立则执行另一个程序段,因此,称其为“分支”。

(3) 循环结构 所谓“循环”是指重复执行某一程序段。循环语句的使用是程序设计的基本技巧。可以避免大量的程序代码的重复编写。

在本教材的第3章,将对这三种基本的程序结构作较为详细的讨论。

4) 程序的质量评价原则

解决某个问题的计算机程序设计方法不止一种,对于一个特定的问题,在理论上存在着一个最好的程序。评价一个程序的质量一般有下列原则:

(1) 功能齐全,运行稳定可靠,运行结果正确。

(2) 在计算机硬件条件一定的条件下,程序运行速度快。

(3) 程序代码精炼、效率高,占用存储器的容量相对较低。

(4) 可移植性好,即程序在另外型号计算机上能够正常运行。这取决于程序设计所采用的语言。

(5) 具有易维护性和可扩展性,即能够方便地进行软件内容的调整及软件的升级换代。

1.1.3 算法及其描述

1) 算法的概念和特性

无论程序规模的大小,总是要涉及“算法”。“算法”是指设计一个程序的具体方法步骤。算法要遵循一个计算问题的数学原理和科学的逻辑。

例如:计算一个自然数 N 的阶乘,平常人们用计算 $1 \times 2 \times \cdots \times N$ 的方法来直接求得结果。而在程序设计的场合,像上面用笔算的方法就显得太没效率了。在这里,人们通常用一个变量值的不断变化来确定参与计算的自变量 $x(x=1, 2, \cdots, N)$,用另一个变量计算并存储阶乘的值。在其中要应用不断重复计算的“循环结构”。

一个具体问题的算法有以下特性:

(1) 有穷性 即一个算法包括有限的运算步骤,而不能有永无止境的运算。通常,要根据常识和运算的需要将运算次数限制在合理的范围之内。

(2) 确定性 算法中的每一个步骤都应当是明确的,而不是含糊不清、模棱两可的。否则,程序设计无法进行。

(3) 有零个或多个输入 “输入”是指在程序运行时需要手工输入数字、文字以及其他符号信息。对于某个程序而言,需要的信息已经包括在程序里面,在运行时可能不再需要输入;有的程序在运行后需要输入参与计算的数据。究竟如何,取决于程序设计时采取的数据处理方式。

(4) 有一个或多个输出 “输出”是指程序运行的结果要在屏幕或打印机上显示出来。程序运行结果不显示,人们就无从知晓。

(5) 有效性 “有效性”是指程序中的每一步都应当能有效地执行,并得到明确的计算结果。如果使用了不合法的语句会导致程序运行的失效。

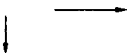
总之,算法对程序设计是重要的。程序设计的好坏在一定程度上取决于算法。因此,计算机科学家 Writhe 提出了“程序就是数据结构加上算法”这一著名的程序设计理念。

2) 算法的描述

虽然最终的程序是人的大脑思考出来的,可在程序设计过程中,由于可能涉及许多变量、函数、表达式以及多种程序结构及语句,人们不能把复杂的问题一下子都想好。这里就需要借助描述算法的工具来循序渐进地进行程序设计。程序设计中描述算法的工具是流程图。流程图有传统流程图和结构化流程图两种形式。

(1) 传统流程图 传统流程图是计算机程序设计人员普遍采用的,用一些特殊流程图符号表示程序运行过程的图形,如表 1.1 所示。

表 1.1 常用的流程图符号

名 称	符号样式	功 能
起止框		表示程序的开始和结束位置
输入输出框		表示程序中的数据输入、输出
判断框		程序中确定分支结构
处理框		表示某种计算处理
流程线		表示程序运行的走向
连接点		在一张纸画不下流程图时,进行标注,下一张续接
注释框		用于程序行(段)意义的注释

【例 1.1】判定 2000—2500 年中的哪些年是闰年。

闰年的条件是:① 年份数能被 4 整除,但不能被 100 整除;② 年份数能被 100 整除,又能被 400 整除。不符合这两个条件的年份不是闰年。

处理这个问题的思路如下:设 x 表示 2000—2500 年间的年份数。先将 2000 赋值给 x ,判断 x 能否被 4 整除,若不能被 4 整除则显示输出“ x 不是闰年”;若能被 4 整除,再判断 x 能否被 100 整除,若不能被 100 整除则显示输出“ x 是闰年”;若能被 100 整除,再判断 x 能否被 400 整除,若能被 400 整除,则显示输出“ x 是闰年”,若不能被 400 整除,则显示输出“ x 不是闰年”。这样就能够判断某一个年份数(如现在的 2000 年)是否符合闰年的条件,从而确定该年份是否为闰年。

经过全面的分析比较后,再将 x 加 1,进行第二轮的比较判断,然后再将 x 加 1,进行第三轮比较判断,依此类推。这里的 x 实际上构成了一个循环变量。从 2000 到 2500 共进行了 501 次的分析比较。这样,可以将所有的闰年挑选出来。

这样的计算过程可用流程图来表示,图 1.1 表示求闰年的流程。

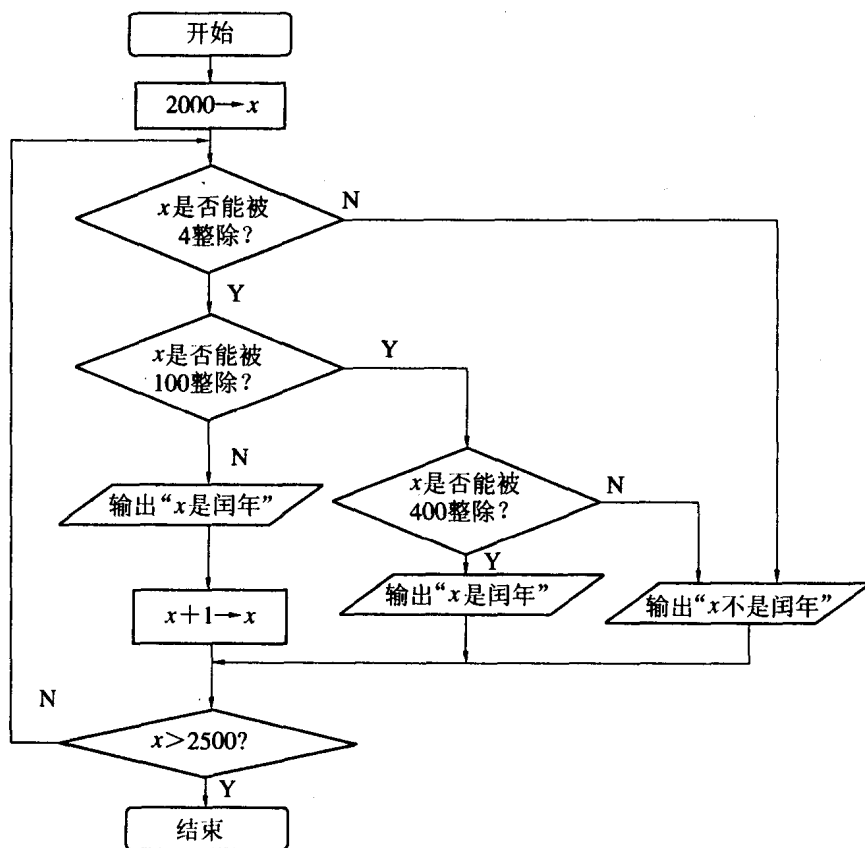


图 1.1 求 2000—2500 年之间的闰年

在上面的例子中,兼有顺序结构、选择结构和循环结构。通常,在一个程序中究竟使用哪一种结构,完全根据需要来决定。

(2) 结构化流程图(N-S流程图) 传统流程图具有简明扼要的特点,但是,在较大型的程序设计中,判断框和流程线的过多使用会使流程图结构过于复杂、条理不清、结构零乱,难于为他人所理解。为此,美国人 I. Nassi 和 B. Shneiderman 提出了另外一种流程图形式,这就是后来人们简称的“N-S流程图”。这种流程图的特点是:用一个矩形框说明一个程序的全部计算步骤及算法,矩形框中再设置从属框用来表示算法的细节。

结构化流程图表示不同程序结构的基本图示如图 1.2 所示。

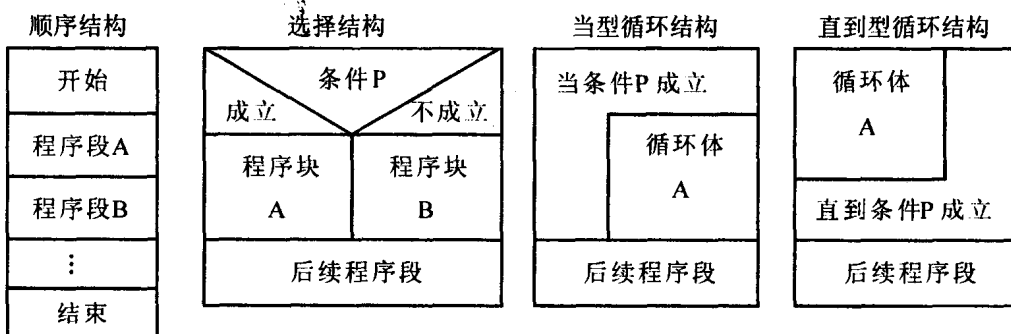


图 1.2 三种结构的 N-S 流程图

【例 1.2】 用 N-S 流程图来描述判定 2000 至 2500 年中的哪一年是闰年程序的流程,如图 1.3 所示。

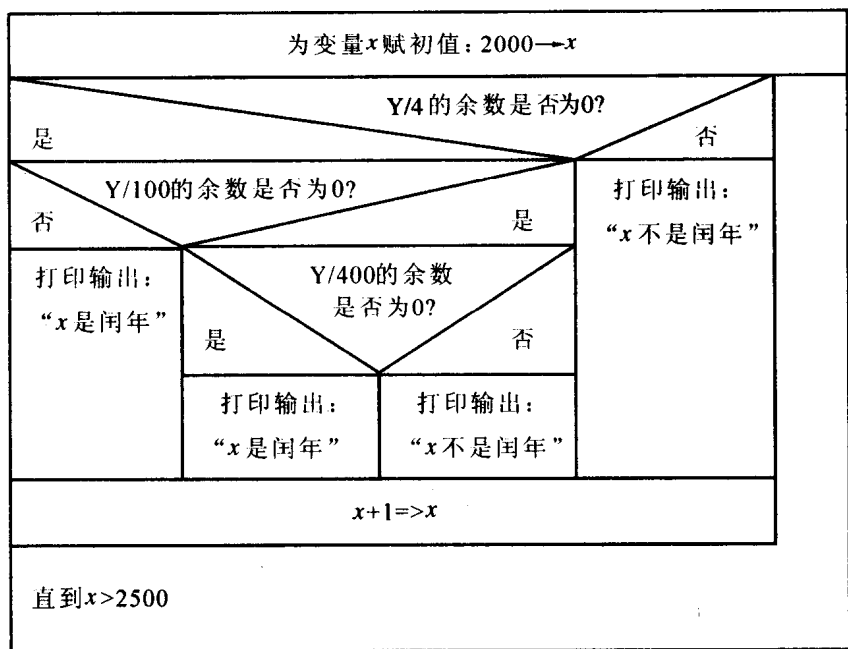


图 1.3 用于计算闰年的 N-S 流程图

1.1.4 结构化程序设计

C 语言是一种面向过程的结构化程序设计语言,不同于具有所见即所得设计特点的 Visual Basic、Delphi、Visual C++ 等面向对象的程序设计语言。C 语言这种面向过程的程序设计语言在程序设计方面具有下列特点:① 整个程序完全用手工编写各种语句;② 程序的总体结构是顺序结构,程序内部的不同功能部分可以是选择(分支)结构或循环结构,即在一个程序中往往涉及几种不同的程序结构问题;③ 解决同一程序设计问题,应用 C 语言比应用其他高级程序设计语言(BASIC、PASCAL)的难度要更大一些。

在处理较大的程序设计问题时,人们提出了“结构化程序设计”的概念。所谓“结构化程序设计”是指:将一个复杂的程序设计问题分解成若干个阶段,分别从较小型的阶段性问题入手进行阶段性设计,就容易完成整个程序设计工作。专家们指出:进行结构化程序设计要遵循以下的原则:自顶向下、逐步细化、模块化设计、结构化编码。其涵义是:要综观程序设计工作的全局,将整个问题分解为若干个较小型的阶段性问题,即将整个程序分成若干个功能模块,再通过各种结构化的程序设计语句编写各功能模块的代码。

这种程序设计思想或原则类似于建造楼房,首先要进行建筑的整体规划,确定整个房屋的建造方案;然后对房屋的各个细节部分进行设计,形成设计图纸;在施工时,从打地基开始,由低层到高层、一间间地建造,这样逐步分阶段地最终完成整个房屋的建造工程。

C 语言为用户提供了丰富的结构化程序设计语句,更重要的是提供了大量的、可直接使用的函数(后面将陆续介绍),以方便程序设计人员进行有条理的、层次分明的结构化程序设计。

1.1.5 C语言的特点

1) C语言的优点

C语言与其他16位面向过程的程序设计语言相比,具有以下优点:

(1) C语言是大小写敏感的语言,它将同一字母的大小写看成是不同的两个字符。通常用小写字母作为标识符,用大写字母表示内部定义的宏、外部变量和常量。在书写程序代码时,要特别注意这个特点。

(2) 数据类型丰富,自定义类型简便、灵活。C语言除了具有常规的整型(int)、浮点型(float、double)、字符型(char)之外,还有结构(struct)、指针(*)、枚举(enum)、链表、位段等。对于同一数据类型还可以根据数据在内存中的存放方式不同进一步区分为不同的数据类型。

(3) 具有低级语言处理地址的能力。C语言用表达式&x表示变量x的地址,*p表示地址单元中p的内容。

(4) 能进行位操作,如按位与(&)、按位或(|)、移位(<<,>>)等。

(5) 语言简捷,如:a=a*3可写成a*=3(当标识符a较长或较复杂时,其优越性更为明显),复合语句(语句组)的标识符为一对大括号{ }。

(6) 良好的模块化结构,使用灵活。一个C语言程序由若干个函数组成,每个函数是完成某一特定任务的模块;而一个大模块又可由若干个小模块组成,只要函数的输入输出接口一致,便可灵活地修改其中的内容,并且可由多个程序员按函数接口要求进行编程,提高工作效率。

(7) 生成的目标代码效率高。C语言程序通常只比汇编语言的代码效率低10%~20%。在高级语言中,C语言程序编译后形成的可执行文件的体积相对较小。

(8) 书写格式自由,对每条语句的起始位置无任何特殊要求,不像其他高级语言那样对每一位置均有严格规定。但这样的自由并不意味着程序的编写不必注意格式。为了增强程序的可读性,必须对程序的格式进行规范,贯彻“层次清楚”的原则。如同一列上开始的语句属同一层次,复合语句中的语句组缩进到分隔符{ }之间,并且各功能块间要有空行并加入适当的注释。此外,在函数的开始处还应对函数的功能、输入输出参数、库文件使用情况进行注释。

(9) 可移植性好。C语言程序能够运行在DOS、Unix、Windows98/2000/XP、Netware等不同操作系统上。C语言是16位面向过程程序设计语言中最具跨平台性质的。

(10) 配置有结构化的语句。C语言具有标准的控制语句以代替随意的转移语句;具有表示层次结构的语句。

2) C语言的缺点

正是由于C语言具有上述优点,又引出了它的若干“缺点”:

(1) 运算符较多,优先级不易记住,个别运算符的含意多,且有双重意义。如:“*”既是乘法运算符,又是指针运算符;地址操作符“&”也是按位与的运算符;小括号“()”既是函数的传递通道,又是表达式的优先运算符,还是数据类型强制转换符等等。

(2) 有些约定与常规不同,如按位与(&)、按位或(|)的优先级较低,在应用时要注意。

(3) 类型转换灵活,容易出错。

(4) 多重复合语句中各层次的区分符,即大括号{ }不醒目。

应该说,这些“缺点”并不是C语言的不足,对于初学者可能是一种学习上的障碍,但对于熟练的程序设计员来说却是程序的高效率设计方式。学习并掌握了C语言之后,就能够体会C语言的独到之处。