

安徽省高等学校省级规划教材

语言程序设计

C YUYAN CHENGXU SHEJI

主 编 钟志水 周鸣争
副主编 强 俊 王 刚
胡为成 黄永青

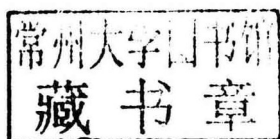


电子科技大学出版社

安徽省高等学校省级规划教材

C 语言程序设计

主 编 钟志水 周鸣争
副主编 强 俊 王 刚
胡为成 黄永青



电子科技大学出版社

内 容 提 要

本书是学习 C 语言程序设计的基础教程,内容包括 C 语言概述、数据类型与运算、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、函数、编译预处理、指针、构造数据类型、文件、班级管理系统的设计与实现。

本书以程序设计思想的掌握为主线,以编程应用为驱动,通过案例和问题引入内容,由浅入深,循序渐进,重点训练编程思想,提高程序设计能力、工程应用能力、解决问题能力和开发创新能力。

本书可作为应用型本科理工类专业的计算机程序设计课程的教材。

图书在版编目(CIP)数据

C 语言程序设计/钟志水,周鸣争主编. —成都:电子科技大学出版社,2015. 5

ISBN 978-7-5647-2972-1

I. ①C… II. ①钟… ②周… III. ①C 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 089047 号

安徽省高等学校省级规划教材

C 语言程序设计

主 编 钟志水 周鸣争

出 版:电子科技大学出版社(成都市一环路东一段 159 号电子信息产业大厦 邮编:610051)

策划编辑:李述娜

责任编辑:李述娜

主 页:www.uestcp.com.cn

电子邮箱:uestcp@uestcp.com.cn

发 行:新华书店经销

印 刷:合肥华星印务有限责任公司

成品尺寸:185mm×260mm 印张 23.5 字数 550 千字

版 次:2015 年 6 月第 1 版

印 次:2015 年 6 月第 1 次印刷

书 号:ISBN 978-7-5647-2972-1

定 价:39.80 元

■ 版权所有 侵权必究 ■

◆ 发行部电话:028-83202463,邮购部电话:028-83208003。

◆ 本书如有缺页、破损、装订错误,请寄回印刷厂调换。

前 言

“C 语言程序设计”是高等院校理工类专业重要的基础课程之一。理论联系实际是该课程的特点,怎样将理论知识应用于实际问题的解决是学好这门课程的重点和难点。本书根据 2013 年版《全国计算机等级考试二级 C 语言考试大纲》,结合安徽省精品课程建设实践和我们多年的教学经验,在分析国内外优秀教材的基础上,针对应用型本科院校人才的培养需求确定的知识结构编写而成。本书于 2013 年 12 月获得安徽省高等学校省级规划教材(皖教高[2013]11 号文件,编号 2013ghjc278)立项。

本书系统地介绍了 C 语言的语法结构、过程化程序设计方法、信息管理系统的开发过程,内容包括 C 语言概述、数据类型与运算、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、函数、编译预处理、指针、构造数据类型、文件、班级管理系统的设计与实现。

本书理论联系实际、语言简练、通俗易懂,以解决问题为主线,重点介绍解决实际问题的思想与方法,不拘泥于 C 语言的语法细节。

本书具有以下特点:

- (1)强调知识的系统性与应用性,注重培养“技能”。
- (2)每章附有阅读材料,旨在拓宽读者的知识面。
- (3)每章配有难易程度不同的练习题,帮助读者复习巩固所学知识。
- (4)实验项目有明确的实验目的、实验内容,方便读者上机实践。
- (5)内容的深度和广度符合最新版全国计算机等级考试二级 C 语言考试大纲要求。

本书共分 12 章,第 1 章由钟志水编写,第 2 章由周鸣争编写,第 3、12 章由黄永青编写,第 4、8 章由强俊编写,第 5、10 章由王刚编写,第 6 章由张步群编写,第 7 章由汪朝霞编写,第 9、11 章由胡为成编写,全书由钟志水统稿和定稿。

本书在编写过程中得到了有关专家的热心指导和无私帮助,电子科技大学出版社为本书的出版做了大量工作,铜陵学院、安徽工程大学和巢湖学院为编者提供了优越的写作环境,我们在此一并表示衷心的感谢!本书在写作过程中参考了大量的文献资料,在此向这些文献资料的作者深表谢意!

本书的编写目标是想让读者在快乐的学习中,迅速提高自己的计算机程序设计能力。由于编者的阅读范围有限,书中还存在很多不理想的地方,也难免有不当和欠妥之处,敬请各位专家、读者多提宝贵意见。

编者
2015 年 3 月

目 录

第 1 章 C 语言概述	1
1.1 程序设计语言	1
1.2 C 语言的发展及特点	2
1.3 程序基本结构	3
1.4 算法及其描述	4
1.5 C 语言字符集、标识符与关键字	11
1.6 C 语言程序开发过程	12
本章小结	19
阅读材料	20
习题一	22
第 2 章 数据类型与运算	24
2.1 数据类型	24
2.2 常量与变量	25
2.3 整型数据	29
2.4 实型数据	31
2.5 字符型数据	32
2.6 运算符和表达式	35
2.7 不同类型数据间的混合运算	43
本章小结	46
阅读材料	47
习题二	48
第 3 章 顺序结构程序设计	52
3.1 数据的输入与输出	52
3.2 C 语言的基本语句	60
3.3 简单顺序结构程序设计	62
3.4 输入输出格式的详细说明	63
本章小结	66
阅读材料	66
习题三	67
实验一 Visual C++6.0 编程环境与顺序结构程序设计	72

第 4 章 选择结构程序设计	75
4.1 if 语句	75
4.2 switch 语句	84
4.3 程序举例	86
本章小结	88
阅读材料	88
习题四	89
实验二 选择结构程序设计	92
第 5 章 循环结构程序设计	94
5.1 while 循环语句	94
5.2 do···while 循环语句	96
5.3 for 循环语句	97
5.4 break 语句和 continue 语句	100
5.5 循环的嵌套	103
5.6 程序举例	104
本章小结	109
阅读材料	109
习题五	112
实验三 循环结构程序设计	117
第 6 章 数组	122
6.1 一维数组的定义与使用	122
6.2 二维数组的定义与使用	127
6.3 字符数组的定义与使用	129
6.4 程序举例	136
本章小结	142
阅读材料	143
习题六	145
实验四 数组应用	152
第 7 章 函数	154
7.1 概述	154
7.2 函数的定义	156
7.3 函数的调用	159
7.4 函数嵌套调用和递归调用	163
7.5 数组作为函数参数	168
7.6 变量的作用域	172
7.7 变量的存储方式	175

7.8 内部函数和外部函数	182
7.9 程序举例	184
本章小结	188
阅读材料	189
习题七	190
实验五 函数应用	194
第 8 章 编译预处理	199
8.1 概念	199
8.2 宏定义	200
8.3 文件包含	205
8.4 条件编译	207
本章小结	210
阅读材料	210
习题八	212
实验六 编译预处理	215
第 9 章 指针	219
9.1 指针的基本概念	219
9.2 指针与数组	227
9.3 指针与函数	241
9.4 程序举例	244
本章小结	248
阅读材料	248
习题九	249
实验七 指针	255
第 10 章 构造数据类型	258
10.1 结构体数据类型	258
10.2 结构体类型数组	262
10.3 结构体指针	265
10.4 结构体应用——链表	270
10.5 共用体数据类型	278
10.6 枚举类型	281
10.7 用户自定义数据类型	283
本章小结	285
阅读材料	285
习题十	289
实验八 构造数据类型	293

第 11 章 文件	299
11.1 文件的概念和文件的种类	299
11.2 文件类型指针	301
11.3 文件的打开和关闭	301
11.4 文件的读写	304
11.5 位置指针与文件定位	311
本章小结	312
阅读材料	313
习题十一	314
实验九 文件	318
第 12 章 班级成绩管理系统的设计与实现	320
12.1 班级成绩管理系统的分析与设计	320
12.2 班级成绩管理系统的实现	325
本章小结	346
阅读材料	347
实验十 班级成绩管理系统	350
附录 I ASCII 码表	352
附录 II 关键字	354
附录 III 运算符的优先级与结合性	356
附录 IV 常用的 C 语言库函数	358
参考文献	365

第 1 章 C 语言概述

C 语言自问世以来一直兴盛不衰,是目前广泛流行的结构化程序设计语言,它既具有高级语言的优点,又具有低级语言的许多特点,适合编写系统软件和应用软件。由于结构化程序设计语言存在一些不足,1980 年后相继出现了一些面向对象程序设计语言,因为一开始学习面向对象程序设计语言难度大,所以 C 语言仍然是众多高校理工科专业首选的第一种程序设计语言。



主要内容

程序设计语言

C 语言的发展及特点

算法的含义、特点和表示方式

Visual C++ 6.0 集成开发环境

C 语言程序开发过程

1.1 程序设计语言

要让计算机了解人的意图,按人的要求进行工作,就需要有与计算机交互的语言。这种人与计算机交互的语言称为程序设计语言(又叫计算机语言)。程序设计语言的发展过程是其功能不断完善、描述问题的方法越来越贴近人类思维方式的过程,程序设计语言发展经历了机器语言→汇编语言(又叫低级语言)→高级语言三个阶段。

1. 机器语言

机器语言是计算机唯一能够识别并能直接执行的语言,实际上机器语言就是计算机中二进制代码指令的集合,用机器语言编写程序十分烦琐,工作量大,且写出的程序可靠性差,只有极少数专业人员使用。

2. 汇编语言

汇编语言是用英文助记符表示的符号语言,克服了机器语言的难读难改的缺点,保持了机器语言的相应优点,它和机器语言都是面向机器的,使用时必须对计算机内部结构有

较为深入的了解,因此仍然存在通用性差、可读性差的缺憾。

3. 高级语言

高级语言接近自然语言,以此编写的程序易学、易读、易改,通用性好,而且不依赖机器,分为面向过程、面向对象的程序设计语言,如:C、C++、C#、Java 语言等。

1.2 C 语言的发展及特点

1.2.1 C 语言发展过程

1978 年美国电话电报公司(AT&T)贝尔实验室正式发布了 C 语言,与此同时,B. W. Kernighan 和 D. M. Ritchie 撰写出版了影响深远的名著《The C Programming Language》,但该书中并没有定义一个完整的标准 C 语言。后来由美国国家标准协会 ANSI (American National Standard Institute)在此基础上制定了一个 C 语言标准,于 1983 年公布,通常称之为 ANSI C。

早期的 C 语言主要用于改写最初用汇编语言开发的 UNIX 操作系统,由于 C 语言的功能强大和优点突出,逐渐为人们所认识,到了 20 世纪 80 年代,C 语言开始移植到其他操作系统,并很快在各类大型、中型、小型、微型机上得到广泛使用,成为现代最优秀的程序设计语言之一。

1987 年,ANSI 又公布了新标准 87 ANSI C。1990 年,国际标准化组织 ISO(International Standard Organization)接受 87 ANSI C 为 ISO C 的标准(ISO 9899—1990)。目前流行的 C 语言编译系统都是以它为基础的,本书的叙述也是以 87 ANSI C 为依据。目前广泛流行的各种版本 C 语言编译系统虽然基本部分是相同的,但也有一些不同。在微机上使用的有 Microsoft C、Turbo C、Quick C、Borland C 等,它们的不同版本又略有差异。因此,读者应了解自己所使用的 C 语言编译系统的特点和规定。

1.2.2 C 语言的特点

C 语言的主要特点有:

(1)语言简洁、紧凑,使用方便、灵活。

(2)运算符丰富。C 语言运算符包含的范围很广,灵活使用各种运算符可以实现其他高级语言难以实现的运算。

(3)数据结构丰富。具有计算机高级语言的各种数据结构,能满足各种复杂问题对数据结构的需求。

(4)具有结构化的控制语句。用函数作为程序的模块单位,便于实现程序的模块化。C 语言是理想的结构化语言,符合现代编程风格的要求。

(5)语法限制不太严格,程序设计自由度大。一般的高级语言对语法检查比较严格,

能检查出几乎所有的语法错误。而C语言自由度较大,放宽了语法检查。“限制”与“灵活”是相互矛盾的。限制严格,失去了灵活性;强调灵活,必然放松限制。

(6)允许直接访问物理地址,能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此,C语言既具有高级语言的功能,又具有低级语言的许多功能。C语言的这种双重性,使它既是系统软件的设计语言,又是应用软件的设计语言。有人把C语言称为“中级语言”,意为兼有高级语言和低级语言的特点。

(7)生成的目标代码质量高,程序执行效率高,一般只比汇编程序生成的目标代码效率低10%~20%。

(8)程序可移植性好(与汇编语言比)。基本上不做修改就能用于各种型号的计算机和操作系统。

以上只是从C语言的整体出发,介绍了C语言与其他语言相比较的一般特点,至于C语言内部的其他特点将结合以后各章的内容做详细介绍。

1.3 程序基本结构

任何一种计算机语言编写出的程序都有三种基本结构,C语言也不例外,这三种基本结构是:顺序结构、选择结构(又称分支结构)、循环结构。

(1)顺序结构是指程序从上往下执行,因此我们在写程序时要把先执行的语句写在上面,后执行的语句写在下面,就像我们平时读书、看报从上往下阅读是一样的。

(2)选择结构是指从两个或多个情况里选择一个,就像走路走到岔路口时,下面选哪一条路,那要根据条件来确定,条件成立时怎么走,条件不成立时又怎么走。

(3)循环结构是指反复执行某一段程序,因为解决问题的需要,一段程序执行完后要返回去再执行一遍,直到任务完成为止。就好像在标准运动场的400米跑道上跑10000米,跑完一圈接着跑下一圈,总共要跑25圈。

以上三种基本结构,有以下共同特点:

(1)只有一个入口。

(2)只有一个出口。

(3)结构内的每一部分都有机会被执行到。

(4)结构内不存在“死循环”(无终止的循环)。

已经证明,有了以上三种基本结构可以实现任何复杂的算法。

1.4 算法及其描述

1.4.1 算法概念

解决任何问题都要有算法,所谓算法就是为解决问题而采取的方法和步骤,即第一步做什么,第二步做什么,第三步做什么,如此下去直至求出问题的解。打个比方,现在我们要请客吃饭,正确的算法是:先买菜,然后择菜洗菜,最后炒菜。如果你把菜买回来后,先炒菜,然后择菜,最后洗菜,那就是算法错了。

数据是算法的处理对象,一个问题中会有各种各样的数据(即数据类型),算法就是按一定顺序对这些数据进行处理以得到期望的结果。

数据类型和数据结构(即数据组织形式)不同,对应的算法也不同。例如,从非数值数据“12345”中找出“3”,与从数值数据 12345 中找出 3 的算法不样。再例如,从有序的队列 1、2、3、4、5 中找出 3,与从无序的集合{2,1,5,3,4}中找出 3 的算法也不一样。

由此可以看出,程序设计时必须要考虑数据结构及算法。著名的计算机科学家沃思(Nikiklaus Wirth)提出了程序定义的式子:

$$\text{程序} = \text{数据结构} + \text{算法}$$

这个式子说明了程序与算法的关系,以及数据结构选择的重要性。数据结构不同算法也有所区别。对初学程序设计的人来讲应该把主要精力放在算法设计上,等到具有一定的程序设计能力后再把主要精力放在数据结构设计上。

同一个问题可以有不同的算法。例如,求 $1+2+3+\cdots+100$ 的值,有的人先求 $1+2$,再加 3,再加 4,一直加到 100,即 $((\cdots((1+2)+3)+\cdots)+100)$ 。有的人采取这样的方法:

$$\begin{array}{cccccccc}
 & 1 & + & 2 & + & 3 & + & \cdots & + & 48 & + & 49 & + & 50 & + \\
 & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & \\
 100 & + & 99 & + & 98 & + & 97 & + & \cdots & + & 52 & + & 51 & \\
 \parallel & & \parallel & & \parallel & & \parallel & & \parallel & & \parallel & & \parallel & \\
 100 & & 100 & & 100 & & 100 & & \cdots & & 100 & & 100 & & 50
 \end{array}$$

50个100

有的人用等差数列前 n 项求和公式来计算:因为数列 $1, 2, 3, \cdots, 100$ 是公差为 1 的等差数列,利用等差数列前 n 项之和公式 $S_n = \frac{n(a_1 + a_n)}{2}$, 可得

$$1+2+3+\cdots+100 = S_{100} = \frac{100(1+100)}{2} = 5050$$

从上面例子的三种求解方法可以看出,同一问题的不同算法有优劣之分。有的算法步骤少、计算简单、不易出错,而有的算法步骤多、计算复杂、容易出错。人们总是喜欢计算简单、步骤少的算法。因此,为了有效地解决问题,不仅要考虑算法的正确性,还要考虑算法的复杂性,选择合适的算法。

计算机算法可分为两大类:数值算法和非数值算法。数值算法是求问题的数值解,例如求代数方程的根,求定积分,求微分方程的解等。非数值算法涉及的范围很广,常见的是事务处理,例如排序、查找等。

1.4.2 算法特点

算法一般具有以下特点:

1. 有穷性

一个算法只能包含有限步的操作,而不能是无限的。这里的“有限”是指“在合理的范围之内有限”。如果让计算机执行一个历时 1000 年才结束的算法,这虽然是有限的,但超过了合理的限度,人们不可能等那么长时间,所以这个算法不能算有效算法。

2. 确定性

算法中的每一步都应当是确定的,而不应当是含糊的、模棱两可的。如“ n 被一个整数除,得余数 r ”这是“不确定”的,它没有说明 n 被哪一个整数除。也就是说,算法的含义应当是唯一的,而不应当产生“歧义性”。所谓“歧义性”是指可以被理解为两种(或多种)的可能含义。

3. 可行性

算法中的每一步必须是计算机语言能实现的。比如,想在计算机屏幕上显示一张九九乘法表,如果我们对计算机说“显示一张九九乘法表”就能实现的话,这个算法当然很好(只需对计算机说一句话),但目前的计算机语言没办法实现,所以这个算法是不可行的。

4. 输入

所谓输入是指在执行算法时需要从外界取得必要的信息。例如,求两个整数 m 和 n 的最大公约数,则需要输入 m 和 n 的值。没有输入的算法缺乏灵活性。一般情况下,一个算法应有一个或多个输入。

5. 输出

算法的目的是为了求解问题,“解”就是输出。但算法的输出不一定是计算机的显示或打印输出,一个算法得到的结果就是算法的输出。没有输出的算法是没有意义的,因此,一个算法要有一个或多个输出。

1.4.3 算法表示

算法的表示方法有很多,常用的有:自然语言、N-S图、伪代码、计算机语言等。

1. 用自然语言表示算法

自然语言就是人们日常使用的语言,如汉语、英语或其他语言。用自然语言表示通俗

易懂,但文字冗长,容易出现“歧义性”。自然语言表示的含义往往不太严格,要根据上下文才能判断其正确含义。假如有这样一句话:“张先生对李先生说他的孩子考上了大学。”请问是张先生的孩子考上大学呢,还是李先生的孩子考上大学呢?光从这句话本身难以判断。此外,用自然语言描述包含分支结构和循环结构的算法不太方便。因此,除了很简单的问题以外,一般不用自然语言描述算法。

【例 1-1】求边长为 a 的正方形外接圆面积。

算法用自然语言描述如下:

第一步:输入 a	(输入正方形边长 a)
第二步: $d=a/\sqrt{2}$	(求外接圆直径 d)
第三步: $r=d/2$	(求外接圆半径 r)
第四步: $S=3.1415926 * r * r$	(求外接圆面积 S)
第五步:输出 S	(输出外接圆面积 S)

2. 用 N-S 图表示算法

1973 年美国学者 I. Nassi 和 B. Shneiderman 提出了结构化流程图(简称 N-S 图,N 和 S 是他们姓名的第一个字母)。这种流程图适合结构化程序设计,因而很受欢迎。

三种基本结构的 N-S 图如下:

(1)顺序结构,用图 1-1 表示。A 和 B 两个矩形框组成一个顺序结构,即先执行 A 操作,然后执行 B 操作。

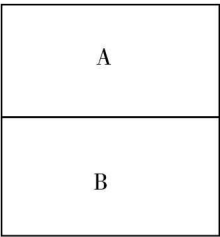


图 1-1 顺序结构 N-S 图

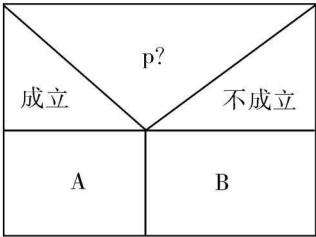


图 1-2 选择结构 N-S 图

(2)选择结构,用图 1-2 表示。当条件 p 成立时,执行 A 操作不执行 B 操作; p 不成立时,执行 B 操作不执行 A 操作。请注意图 1-2 是一个整体,代表一个选择结构。

(3)循环结构。

①当型循环结构,用图 1-3 表示。当条件 p_1 成立时执行 A 操作, p_1 不成立时不执行 A 操作;然后再判断条件 p_1 , p_1 成立时执行 A 操作, p_1 不成立时不执行 A 操作;如此反复下去。

②直到型循环结构,用图 1-4 表示。执行 A 操作后判断条件 p_2 ,当 p_2 成立时执行 A 操作,当 p_2 不成立时不执行 A 操作;然后再判断条件 p_2 , p_2 成立时执行 A 操作, p_2 不成立时不执行 A 操作;如此反复下去。

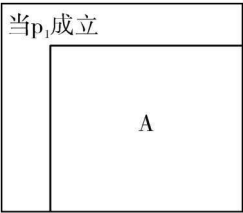


图 1-3 当型循环结构 N-S 图

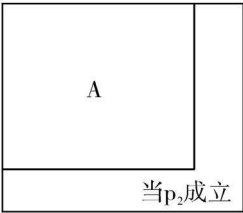


图 1-4 直到型循环结构 N-S 图

当型循环结构与直到型循环结构的区别在于：当型循环结构中的 A 操作可能一次都没有执行(即一开始条件 p_1 就不成立)，直到型循环结构中的 A 操作至少执行一次。两者之间通常情况下可以相互转换。

需要指出的是，图 1-1~图 1-4 中的 A 框或 B 框，可以是一个简单操作(如输入数据或输出结果等)，也可以是 3 个基本结构之一。如图 1-1 所示的顺序结构，其中的 A 框可以是一个选择结构，B 框又可以是一个循环结构。图 1-5 就是这样的 N-S 图。

下面举几个用 N-S 图表示算法的例子。

【例 1-2】求 5!。

如图 1-6 所示的 N-S 图。

【例 1-3】输入 50 名同学的学号和一门课成绩，输出成绩大于等于 80 分同学的学号和成绩。

如图 1-7 所示的 N-S 图。

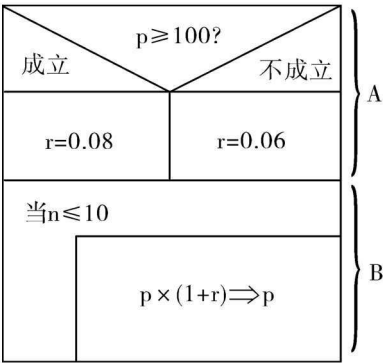


图 1-5 一般的顺序结构 N-S 图

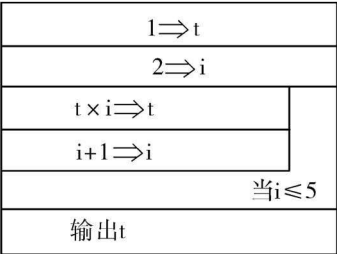


图 1-6 例 1-2 N-S 图

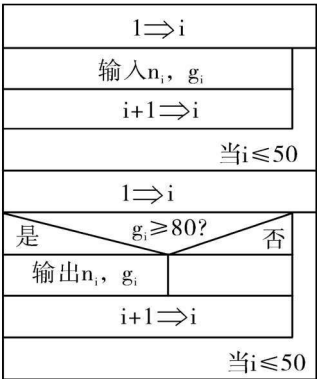


图 1-7 例 1-3 N-S 图

【例 1-4】求 $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{99} - \frac{1}{100}$ 。

如图 1-8 所示的 N-S 图。

由上面几个例子可以看出，用 N-S 图表示算法的优点：它比文字描述直观、形象、易

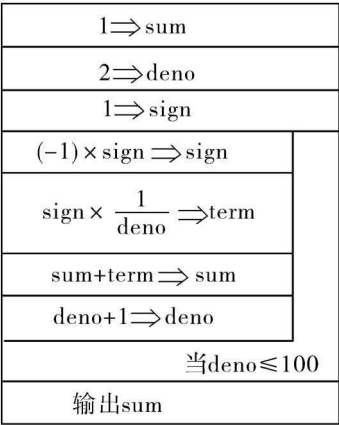


图 1-8 例 1-4N-S 图

于理解,整个流程图是由各个基本结构流程按顺序组成的。N-S 图中的上下顺序就是执行时的顺序,即图中位置在上面的先执行,位置在下面的后执行。写算法和看算法只需从上到下进行就可以了,十分方便。用 N-S 图表示的算法都是结构化的算法。

3. 用伪代码表示算法

用 N-S 图表示算法,直观易懂,但画起来比较费事,在设计一个算法时,可能要反复修改,而修改 N-S 图是比较麻烦的。因此,N-S 图适宜表示一个算法,但在设计算法过程中使用不是很理想(尤其是当算法比较复杂、需要反复修改时)。为了设计算法时方便,常用一种称为伪代码(Pseudo Code)的工具。

伪代码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。它如同一篇文章,自上而下地写出来。每一行(或几行)表示一个基本操作。它不用图形符号,因此书写方便、格式紧凑,也比较易懂,便于用计算机语言表示算法(即程序)过渡。

例如,“输出 x 的绝对值”的算法可以用伪代码表示如下:

```
IF x is positive THEN
    print x
ELSE
    print -x
```

它像一个英语句子一样好懂,也可以用汉字伪代码,如:

```
若 x 为正
    输出 x
否则
    输出 -x
```

也可以中英文混用,如:

```
IF x 为正
    print x
ELSE
    print -x
```

总之,用伪代码表示算法无固定、严格的语法规则,以便于书写和阅读为原则,只要把意思表达清楚,书写成清晰易读的形式就行。

下面举几个用伪代码表示算法的例子。

【例 1-5】求 5!。

```
开始
    置 t 的初值为 1
    置 i 的初值为 2
    当 i ≤ 5, 执行下面操作:
        使 t = t × i
```

```

    使  $i=i+1$ 
    (循环体到此结束)
    输出  $t$  的值
    结束
    也可以写成以下形式:
    BEGIN(算法开始)

```

```

    1 $\Rightarrow$  $t$ 
    2 $\Rightarrow$  $i$ 
    while  $i\leq 5$ 
    {
         $t\times i\Rightarrow t$ 
         $i+1\Rightarrow i$ 
    }
    print  $t$ 

```

END(算法结束)

在本算法中采用当型循环(第 3 行到第 5 行是一当型循环),while 的意思为“当”,表示当 $i\leq 5$ 时执行循环体(大括弧中的两行)的操作。

【例 1-6】输入 50 名同学的学号和一门课成绩,输出成绩大于等于 80 分同学的学号和成绩。

```

    BEGIN(算法开始)
    1 $\Rightarrow$  $i$ 
    while  $i\leq 50$ 
    {   input  $n_i$  and  $g_i$ 
         $i+1\Rightarrow i$ 
    }
    1 $\Rightarrow$  $i$ 
    while  $i\leq 50$ 
    {   if  $g_i\geq 80$  print  $n_i$  and  $g_i$ 
         $i+1\Rightarrow i$ 
    }
    END(算法结束)

```

【例 1-7】求 $1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\cdots+\frac{1}{99}-\frac{1}{100}$ 。

```

    BEGIN(算法开始)
    1 $\Rightarrow$ sum
    2 $\Rightarrow$ deno
    1 $\Rightarrow$ sign
    while deno $\leq$ 100
    {    $(-1)\times\text{sign}\Rightarrow\text{sign}$ 
         $\text{sign}\times 1/\text{deno}\Rightarrow\text{term}$ 
         $\text{sum}+\text{term}\Rightarrow\text{sum}$ 

```