

570
□中国高等职业技术教育研究会推荐

高 职 系 列 教 材

C 语言程序设计实例教程

丁爱萍 郝小会 孙宏莉 编著

西安电子科技大学出版社

2002

内 容 简 介

本书按照全国计算机等级（二级）考试大纲内容编写。书中通过大量实例详细讲述了目前国内广泛使用的结构化程序设计语言——C语言的使用方法。主要包括：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。

本书的讲解由浅入深、通俗易懂，且其内容条理清晰、示例丰富。每一章均提供了相当数量的习题供读者练习；为帮助读者掌握自己的学习程度，附录中还提供了全部习题（共209题）的答案。

本书可作为高职高专院校计算机程序设计的学习教材，也适用于程序设计爱好者自学使用。

图书在版编目（CIP）数据

C语言程序设计实例教程 / 丁爱萍等编著. —西安：西安电子科技大学出版社，2002.2
高职系列教材

ISBN 7-5606-1108-7

I. C… II. 丁… 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 098893 号

策 划 马乐惠

责任编辑 马晓娟

出版发行 西安电子科技大学出版社（西安市太白南路2号）

电 话 (029)8227828 邮 编 710071

<http://www.xduph.com>

E-mail: xdupfbx@pub.xaonline.com

经 销 新华书店

印 刷 陕西画报社印刷厂

版 次 2002年2月第1版 2002年2月第1次印刷

开 本 787毫米×1092毫米 1/16 印张 17.625

字 数 414千字

印 数 1~4 000册

定 价 18.00元

ISBN 7-5606-1108-7 / TP·0557 (课)

XDUP 1379001-1

*** 如有印装问题可调换 ***

本书封面贴有西安电子科技大学出版社的激光防伪标志，无标志者不得销售。

序

在即将跨入 21 世纪的前夕,中共中央、国务院召开了第三次全国教育工作会议,并颁发了《中共中央、国务院关于深化教育改革全面推进素质教育的决定》,进一步明确了高等职业教育的重要地位,指出“高等职业教育是高等教育的重要组成部分。要大力发展高等职业教育。”在这一方针的指引下,我国高等职业教育取得了空前规模的发展。至 1999 年,从事高等职业教育的高等职业学校、高等专科学校和独立设置的成人高校已达 1345 所,占全国高校总数的 69.2%;专科层次的在校生占全国高校在校生的 55.37%,毕业生占高校毕业生总数的 68.5%。这些数字表明,高等职业教育在我国高等教育事业中占有极其重要的地位,在我国社会主义现代化建设事业中发挥着极其重要的作用。随着社会的发展、科技的进步,以及我国高等教育逐步走向大众化,我国的高等职业教育必将进一步发展壮大。

在高等职业教育大发展的同时,也有着许多亟待解决的问题。其中最主要的是按照高等职业教育培养目标的要求,培养一批“双师型”的中青年骨干教师;编写出一批有特色的基础课和专业主干课教材;创建一批教学工作优秀学校。

为解决当前高职教材严重匮乏的问题,西安电子科技大学出版社与中国高等职业技术教育研究会联合策划、组织编写了计算机及应用电子技术两个专业的教材,现已出版。本系列教材,从策划到主编、主审的遴选,从成立专家组反复讨论大纲,研讨职业教材特色到书稿的字斟句酌,每走一步都比较扎实、十分精心。作者在编写中紧密联系实际,尽可能地吸收新理论、新技术、新工艺,并按照案例引入、改造拓宽、课题综合(通过一个大型的课题,综合运用所学内容)的思路,进行编写,努力突出高职教材的特点。本系列教材内容取材新颖、实用;层次清楚,结构合理;文笔流畅,装帧上乘。这套教材比较适合高等职业学校、高等专科学校和成人高校等高等职业教育的需要。

教材建设是高等职业院校基本建设的主要工作之一,是教学内容改革的重要基础。为此,有关高职院校都十分重视教材建设,组织教师积极参加教材编写,为高职教材从无到有,从有到优而辛勤工作。但高职教材的建设还刚刚起步,还需要做艰苦的工作,我们殷切地希望广大从事高等职业教育的教师,在教书育人的同时,组织起来,共同努力,编写出一批高职教材的精品,为推出一批有特色的、高质量的高职教材作出积极的贡献。

中国高等职业技术教育研究会会长

李宗尧

高等职业技术教育“计算机及应用电子技术专业” 教材编审专家委员会

主任：闵光太(中国高等职业技术教育研究会副会长，
金陵职业大学校长，教授)

副主任：俞克新(中国高等职业技术教育研究会秘书长，研究员)

孙建京(北京联合大学教务长，副教授)

余苏宁(深圳职业技术学院计算机应用工程系副主任，副教授)

李荣才(西安电子科技大学出版社总编辑，教授)

计算机组

组长：余苏宁(兼)

成员：(按姓氏笔画排列)

丁桂芝(天津职业大学计算机工程系主任，副教授)

朱振元(长沙大学高级工程师)

张 燕(金陵职业大学计算机系讲师)

唐连章(广州大学副教授)

韩伟忠(金陵职业大学计算机系主任，副教授)

樊月华(北京联合大学应用技术学院副教授)

颜 彬(江汉大学副教授)

应用电子技术组

组长：孙建京(兼)

成员：(按姓氏笔画排列)

付植桐(天津职业大学副教授)

刘守义(深圳职业技术学院电子通信工程系副主任，高工)

李建民(江汉大学应用物理系副主任，副教授)

高泽涵(广州大学机电工程系副主任，高级实验师)

鲁宇红(金陵职业大学副校长，副教授)

熊幸明(长沙大学工程系主任，副教授)

总策划：梁家新

策 划：马乐惠 徐德源 云立实

前 言

C 语言是目前广泛用于软件开发的一种编译型程序设计语言。与其它高级语言相比，C 语言具有以下优点：处理功能丰富，表达能力强，使用灵活方便，执行程序效率高，可移植性强；具有丰富的数据类型和运算符，语句非常简单，源程序简洁清晰；具有汇编语言的能力，可以直接处理硬件系统和外围设备接口的控制。C 语言是一种结构化的程序设计语言，支持自顶向下、逐步求精的结构化程序设计技术。另外，C 语言程序的函数式结构也为实现程序的模块化设计提供了强有力的保障。因此，C 语言被广泛地应用于系统软件的开发设计中，是大中型软件设计的常用工具。

本书按照全国计算机等级（二级）考试大纲内容编写。全书通过大量实例详细讲述了目前国内广泛使用的结构化程序设计语言——C 语言的使用方法，主要内容有：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、文件操作等。全书共分 13 章：

第 1 章介绍了程序设计的基本概念、算法分析、程序设计的步骤以及 C 语言的特点、基本结构、上机步骤等。

第 2 章讲解了 C 语言程序设计的初步知识，包括标识符、常量、变量的概念及其基本运算表达式等。

第 3 章是简单的程序设计，包括赋值语句、数据输出、数据输入、复合语句和空语句等。

第 4 章讲解选择结构的程序设计，介绍关系运算符、逻辑运算符及其表达式，以及条件运算符、if 语句、switch 语句的使用方法。

第 5 章讲述循环结构的程序设计，介绍当型循环——while 语句、直到型循环——do ... while 语句、次数型循环——for 语句等。

第 6 章讲述一维数组、二维数组和字符数组的定义、引用、初始化等。

第 7 章介绍函数的定义、调用及程序设计方法。

第 8 章介绍指针的概念，指针变量的定义、赋值和引用，以及数组指针和函数指针的使用方法。

第 9 章讲述编译预处理的方法，包括宏定义、文件包含和条件编译的使用方式等。

第 10 章介绍结构体和共用体的特点、定义和使用及程序设计方法。

第 11 章讲述位运算符及其运算功能。

第 12 章讲述 C 语言中文件的概念、文件指针及打开和关闭文件的方法，常用文件操作函数等。

第 13 章讲解 C 语言的程序调试方法，指出用户在使用 C 语言中经常出现的错误。

本书由浅入深，通俗易懂，结构清晰，层次分明，示例丰富。书中每一章均提供了相当数量的习题供读者练习。为帮助读者掌握自己的学习程度，还在附录中提供了该教材全部习题（共 209 题）的答案。

本书可作为高职高专院校的计算机程序设计学习教材，也适用于程序设计爱好者自学使用。

本书由丁爱萍、郝小会、孙宏莉主编；王文陵主审。参加编写的人员还有：赵全利、张辉、曹建春、朱一、苏锡锋、关六三、张二峰、马国俊、郭新勇、翟红生、杨东艳、祁迎华、张勇等。由于作者水平有限，书中错误疏漏之处在所难免，恳请广大读者批评指正。

作者联系邮箱：E-mail: dingap@371.net

作 者

2001 年 12 月

目 录

第1章 程序设计与C语言1	3.2 赋值语句.....36
1.1 程序设计的基本概念.....1	3.3 数据输出.....37
1.1.1 程序.....1	3.3.1 printf 函数.....37
1.1.2 计算机语言.....1	3.3.2 putchar 函数.....39
1.1.3 程序设计.....2	3.4 数据输入.....40
1.2 算法.....2	3.4.1 scanf 函数.....40
1.2.1 算法设计.....3	3.4.2 getchar 函数.....41
1.2.2 算法的描述.....3	3.5 复合语句和空语句.....42
1.3 结构化程序及设计步骤.....4	3.5.1 复合语句.....42
1.3.1 结构化程序设计的概念.....4	3.5.2 空语句.....42
1.3.2 结构化程序设计步骤.....7	习题3.....43
1.4 C语言的特点.....9	第4章 选择结构程序设计46
1.5 C语言程序的基本结构.....10	4.1 关系运算符及其表达式.....46
1.6 C程序的上机操作.....12	4.2 逻辑运算符及其表达式.....47
1.6.1 Turbo C的安装.....12	4.3 条件运算符.....49
1.6.2 Turbo C的使用.....12	4.4 单条件选择if语句.....49
习题1.....18	4.4.1 if语句.....49
第2章 C语言程序设计初步知识19	4.4.2 if语句的嵌套.....50
2.1 标识符、常量和变量.....19	4.5 开关分支switch语句.....53
2.1.1 标识符.....19	习题4.....57
2.1.2 常量和变量.....20	第5章 循环结构程序设计61
2.2 简单数据类型.....20	5.1 循环结构程序设计的概念.....61
2.2.1 整型数据.....21	5.2 while语句.....61
2.2.2 实型数据.....24	5.2.1 while语句的一般格式.....61
2.2.3 字符型数据.....25	5.2.2 while语句的执行过程.....62
2.3 基本运算表达式.....26	5.3 do...while语句.....67
2.3.1 算术运算符及其表达式.....26	5.3.1 do...while语句的一般格式.....67
2.3.2 赋值运算符及其表达式.....27	5.3.2 do...while语句的执行过程.....67
2.3.3 不同类型数据间的转换.....29	5.4 for语句.....68
2.3.4 自增、自减运算符及其表达式.....33	5.4.1 for语句的一般格式.....68
2.3.5 逗号运算符及其表达式.....33	5.4.2 for语句的执行过程.....69
习题2.....34	5.5 循环结构嵌套.....72
第3章 简单程序设计36	5.6 break语句和continue语句.....73
3.1 顺序结构程序设计的概念.....36	5.6.1 break语句.....73

5.6.2 continue 语句	73	7.10 内部函数和外部函数	129
5.7 语句标号和 goto 语句	74	7.10.1 内部函数	129
习题 5	75	7.10.2 外部函数	130
第 6 章 数组	82	7.11 程序举例	130
6.1 数组和数组单元的基本概念	82	习题 7	133
6.2 一维数组	83	第 8 章 指针	138
6.2.1 一维数组的定义、引用和初始化	83	8.1 指针的基本概念	138
6.2.2 一维数组应用示例	86	8.2 指针和指针变量	139
6.3 二维数组	88	8.2.1 指针变量的定义	139
6.3.1 二维数组的定义、引用和初始化	88	8.2.2 指针变量的赋值	140
6.3.2 二维数组应用示例	90	8.2.3 指针变量的引用	140
6.4 字符数组	91	8.2.4 指针变量作为函数参数	142
6.4.1 字符数组的定义、引用和初始化	91	8.3 数组和指针	144
6.4.2 字符串与字符型数组	92	8.3.1 一维数组与指针	144
6.4.3 字符数组的输入、输出	94	8.3.2 二维数组与指针	147
6.4.4 字符串处理函数	95	8.3.3 指针与字符串	149
6.4.5 字符串数组应用示例	97	8.3.4 指向数组的指针作为函数参数	152
习题 6	99	8.4 函数的指针	154
第 7 章 函数	106	8.5 数组、函数、指针综合应用示例	155
7.1 模块化程序设计的概念	106	习题 8	155
7.2 库函数	107	第 9 章 编译预处理	163
7.2.1 C 语言常用库函数	107	9.1 宏定义	163
7.2.2 标准库函数的调用	107	9.1.1 不带参数的宏定义	163
7.3 函数的定义和调用	108	9.1.2 带参数的宏定义	167
7.3.1 函数的定义	108	9.2 文件包含	171
7.3.2 函数的调用	109	9.3 条件编译	172
7.4 函数的返回值及其类型	113	习题 9	174
7.5 函数调用时参数间的传递	117	第 10 章 结构体和共用体	175
7.5.1 变量、常量、数组元素作为 函数参数	117	10.1 结构体类型	175
7.5.2 数组名作为函数参数	118	10.1.1 结构体类型定义	175
7.6 函数的嵌套调用	121	10.1.2 结构体变量的定义、引用和 初始化	176
7.7 函数的递归调用	123	10.1.3 结构体数组	179
7.8 局部变量和全局变量	125	10.1.4 结构体的指针	181
7.8.1 局部变量	125	10.1.5 利用结构体变量构成链表	186
7.8.2 全局变量	126	10.2 共用体	189
7.9 静态存储变量和动态存储变量	127	10.2.1 共用体类型说明	190
7.9.1 静态存储变量	127	10.2.2 共用体变量的定义和引用	190
7.9.2 动态存储变量	128	习题 10	191

第 11 章 位运算	192	12.4.1 文件的输入、输出	202
11.1 位运算符	192	12.4.2 文件的定位	211
11.1.1 按位取反	193	12.4.3 出错的检测	217
11.1.2 移位	193	习题 12	218
11.1.3 按位“与”	193	第 13 章 程序调试与常见错误分析	219
11.1.4 按位“或”	194	13.1 程序调试	219
11.1.5 按位“异或”	195	13.1.1 程序调试的步骤	219
11.2 位运算符的运算功能	195	13.1.2 检查和分析错误原因	220
习题 11	198	13.2 常见错误分析	221
第 12 章 文件	199	附录	233
12.1 文件概念	199	附录 A 各章习题参考答案	233
12.2 文件指针	200	附录 B C 语言的标准库函数	260
12.3 打开和关闭文件	200	附录 C Turbo C 编译、链接时的错误 和警告信息	269
12.3.1 文件的打开 (fopen 函数)	200	附录 D ASCII 码表	272
12.3.2 文件的关闭 (fclose 函数)	202		
12.4 常用文件操作函数	202		

第 1 章 程序设计与 C 语言

C 语言是目前国际上广泛流行的一种结构化程序设计语言。它不仅是开发系统软件的理想工具，也是开发应用软件的理想程序设计语言，因此，深受广大程序设计者的欢迎。

1.1 程序设计的基本概念

1.1.1 程序

计算机科学技术的发展在改变着世界，计算机在人类社会生活中已被广泛应用并产生着重大影响。但对有些人来说，计算机仍然是十分神秘的，其实，计算机只是一种以二进制数据形式在内部存储信息、以程序存储为基础、由程序自动控制的电子设备。

人们需要计算机所做的任何工作，都必须以计算机所能识别的指令形式送入计算机内存中。一条条有序指令的集合称为程序。计算机的工作过程也就是执行程序的过程，计算机所做的任何工作都是执行程序的结果。可以说，程序是计算机赖以生存的食粮，离开程序，计算机寸步难行。

1.1.2 计算机语言

程序设计语言是实现程序设计，以便于人与计算机进行信息交流的必备工具，又称其为计算机语言。目前，社会上广泛使用的是高级语言，它是一种接近人们习惯的程序设计语言。高级语言使用人们所熟悉的文字、符号及数学表达式来编写程序，使程序的编写和操作都显得十分方便。由高级语言编写的程序称为“源程序”。在计算机内部，源程序必须翻译为机器能够接受的二进制代码所表示的“目标程序”（也称机器语言）。其中，具有这种翻译功能的程序称为“编译程序”，如图 1.1 所示。

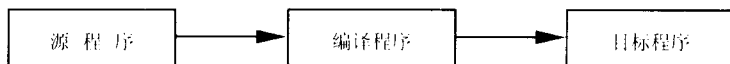


图 1.1 源程序编译过程

每一种高级语言都有与其相应的编译程序。在计算机内运行编译程序，才能运行相应的高级语言所编写的源程序。

本书所介绍的 C 语言是目前国内外广泛流行的高级程序设计语言，是面向过程的较好的结构化程序设计语言。它不仅可以用来编写系统软件，也可以用来编写应用软件，同时也是面向对象程序设计技术的主要工具。

1.1.3 程序设计

下面给出计算机在处理简单问题时，程序设计的一般步骤。

1. 确定数据结构

依据任务提出的要求，规划输入数据和输出的结果，确定存放数据的数据结构。

2. 确定算法

针对所确定的数据结构确定解决问题的步骤。

3. 编程

根据算法和数据结构，用程序设计语言编写程序，存入计算机中。

4. 调试

在编译程序环境下，编译、调试源程序，修改语法错误和逻辑错误，直至程序运行成功。

5. 整理源程序并总结资料

下面给出用 C 语言进行简单程序设计的一个实例。

【例 1-1】 某竞赛小组有 3 个成员，其年龄分别为 17 岁、20 岁、23 岁，求小组的平均年龄。

(1) 分析任务要求：年龄应为整数，不能为小数，故选用整型数据表示年龄。

(2) 算法设计：

- a. 把年龄的值分别赋给 a、b、c 三个变量；
- b. 对 a、b、c 三个变量求和；
- c. 将其和除以 3；
- d. 结果输出显示。

(3) 根据算法设计编程。

在 C 编译环境下可直接进行编辑，设文件名为 ave.c。

编辑程序如下：

```
main( )                /* 主函数 */
{
    int a, b, c, s, ave; /* 定义 4 个整型变量 */
    a=17; b=20; c=23;   /* 给各变量赋值 */
    s=a+b+c;            /* 求和赋给 s */
    ave=s/3;             /* 平均值赋给 ave */
    printf("ave=%d", ave); /* 输出 ave 的值 */
}
```

(4) 运行 C 编译程序，对源程序 ave.c 进行编译、链接、调试运行。

1.2 算 法

所谓算法，是指为解决某一特定的问题，所给出的一系列确切的、有限的操作步骤。

1.2.1 算法设计

程序设计的主要工作是算法设计，有了一个好的算法，就会产生质量较好的程序。程序实际上是用计算机语言所描述的算法。也就是说，依据算法所给定的步骤，用计算机语言所规定的表达形式去实现这些步骤，即为源程序。

在算法设计中应遵循下面几个准则：

- (1) 可执行性。算法是编写程序代码的主要依据，算法设计中的每一步骤，都必须是由所使用的高级语言能够描述的操作。
- (2) 确定性。算法中每一操作步骤必须有确切的含义。也就是说，该操作对于相同的输入必能得出相同的结果。
- (3) 有穷性。一个算法必须在有限的操作步骤完成后，得出正确结果，能够使算法结束。
- (4) 输入。一个算法，可以有零个、一个或多个特定对象的输入。
- (5) 输出。一个算法，其主要目的是求解问题，可以有一个或多个与输入相关的输出。

1.2.2 算法的描述

目前，对算法的描述，一般采用自然语言、一般流程图或N-S结构流程图。

常用一般流程图符号如图1.2所示

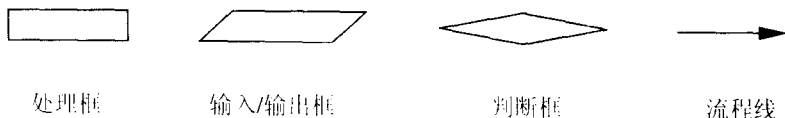


图 1.2 常用流程图符号

【例 1-2】用不同的方式描述求 $s = 1 + 2 + 3 + \dots + 99 + 100$ 的值的算法。

(1) 用自然语言描述。

设一整型变量 i ，并令 $i=1$ （这里的“=”不同于数学里的等号，它表示赋值，这里把 1 赋给 i ，以下类同）。

设一整型变量 s ，用其存放累加和

每次将 i 与 s 相加后存入 s ；

使 i 值增 1，取得下次的加数。

重复执行上步，直到 i 的值大于 100 时，执行下一步。

将累加和 s 的值输出。

(2) 用一般流程图描述，如图 1.3 所示。

(3) N-S 结构流程图（简称 N-S 流程图）是将算法的每一步骤，按序连接成一个大的矩形框来表示，从而完整地描述一个算法。N-S 流程图更适合对结构化程序设计的描述，其描述方法将在下一节介绍。

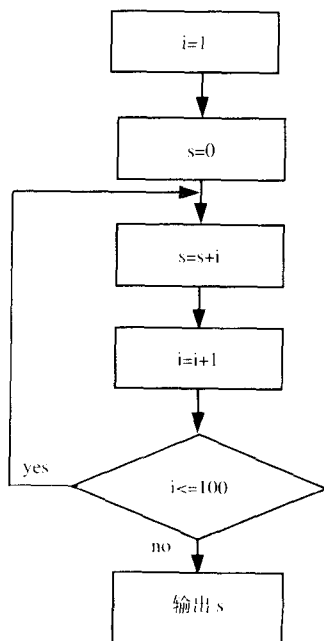


图 1.3 一般流程图

1.3 结构化程序及设计步骤

1.3.1 结构化程序设计的概念

对同一个需要求解的问题，不同的算法会编出不同的程序。

结构化程序要求程序设计者不能随心所欲地编写程序，而要按一定的结构形式来设计、编写程序。在程序设计时，大家都共同遵守这一规定，使程序清晰、易读、易修改。

结构化程序由 3 种基本结构：顺序结构、选择结构、循环结构组成。已经证明，这 3 种基本结构组成的算法可以解决任何复杂的问题。使用这 3 种基本结构编写的程序，称为结构化程序。

下面通过简单的实例说明结构化程序设计的 3 种基本结构。

1. 顺序结构

程序中的语句按先后顺序逐条执行。

【例 1-3】顺序结构示例。求 $s=a+b$ 的值。

其流程图如图 1.4 所示，图(a)是一般流程图，图(b)是 N-S 结构流程图。

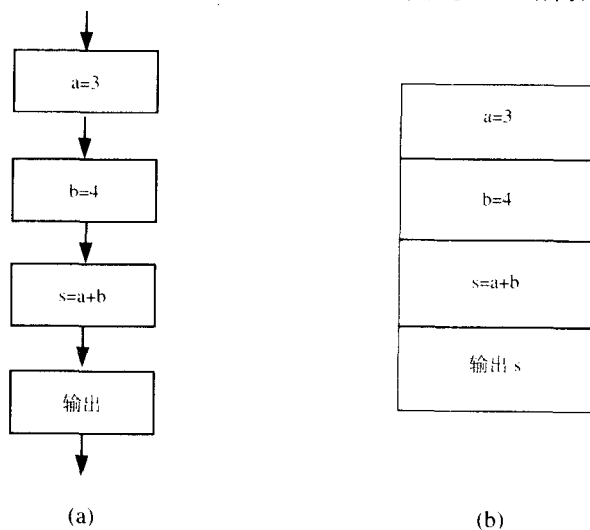


图 1.4 顺序结构流程图

(a) 一般流程图; (b) N-S 结构流程图

源程序如下：

```

main( )                /* 主函数 */
{
    int a, b, s;        /* 定义 a, b, s 为整型变量 */
    a=6; b=8;
    s=a+b;
    printf("s=%d\n", s); /* 输出 s 的值 */
}

```

2. 选择结构

在执行程序中的选择结构语句时，该语句的执行将根据不同的条件执行不同分支的语句。

【例 1-4】选择结构示例。求 a、b 两数中的较大数。

其流程图如图 1.5 所示，图(a)是一般流程图，图(b)是 N-S 结构流程图。

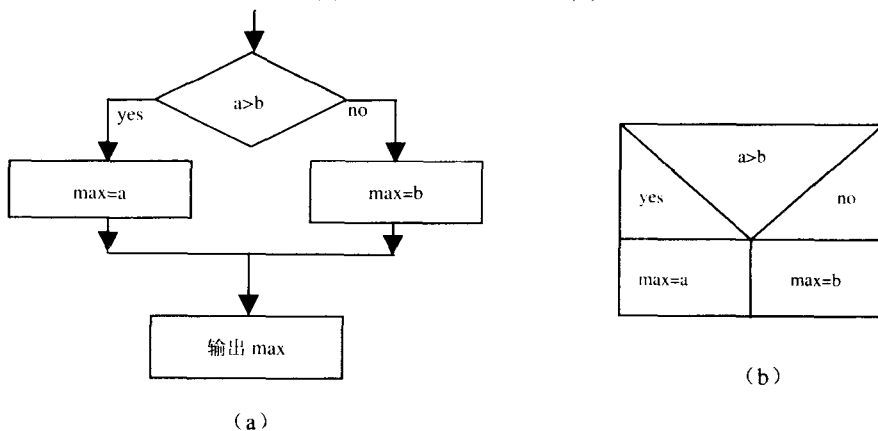


图 1.5 选择结构流程图

(a) 一般流程图; (b) N-S 结构流程图

程序如下：

```
main( )
{
    int  a=6, b=8, max ;
    if(a>b)  max=a ;
    else  max=b ;
    printf("max=%d" , max) ;
}
```

3. 循环结构

在执行程序中的循环结构语句时，该语句将根据各自的条件，对循环结构所限定的语句（即循环体）重复执行多次或零次。

循环结构又分为当型循环和直到型循环两种情况。

1) 当型循环

该循环的特点是：当条件满足时，就执行循环体，否则就退出循环结构。

【例 1-5】循环结构示例。计算 n!。

其流程图如图 1.6 所示，图(a)是一般流程图，图(b)是 N-S 结构流程图。

程序如下：

```
main( )
{
    int  i, n, t;
```

```

i=1 ; t=1 ;
scanf("%d" , &n);
while (i<=n)
    { t=t*i ; i=i+1 ; }
printf("n=%d" , t);
}

```

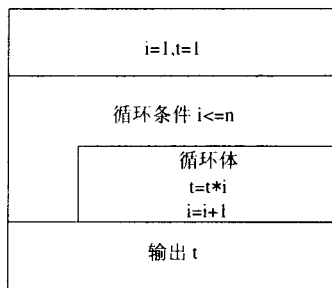
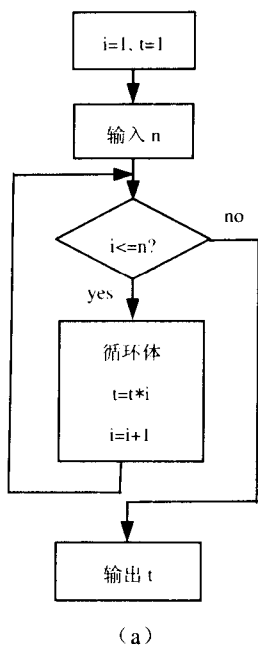


图 1.6 当型循环流程图

(a) 一般流程图; (b) N-S 结构流程图

2) 直到型循环

该循环的特点是：先执行循环体，再判断循环条件，条件不满足时，就退出循环结构。

下面给出用直到型循环求解例 1-5 的流程图（见图 1.7）和程序段。

程序如下：

```

main()
{
    int t=1, i=1;
    do
    { t=t*i;
      i=i+1; }
    while (i<=5);
    printf("t=%d\n", t);
}

```

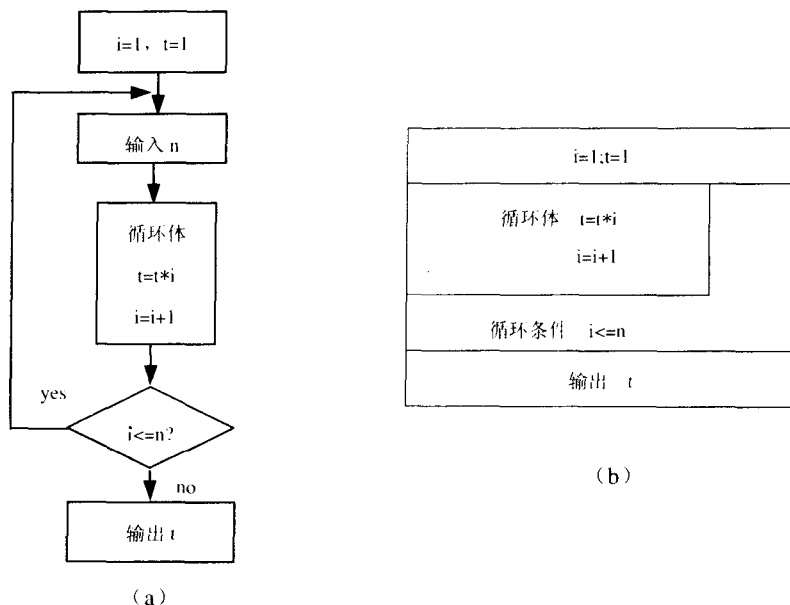


图 1.7 直到型循环流程图

(a) 一般流程图; (b) N-S 结构流程图

1.3.2 结构化程序设计步骤

结构化程序设计步骤与前面一般程序设计步骤大体相同，其步骤如图 1.8 所示。

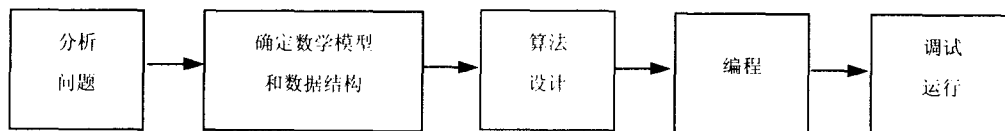


图 1.8 结构化程序设计步骤框图

1. 分析问题

首先正确地分析所要解决的问题，整体上得出程序设计要实现的功能。

2. 建立数学模型，确定数据结构

对于一般问题，可根据所使用的高级语言，确定数据结构。在处理较复杂的有关专业方面问题时，需要把实际问题抽象成数学模型的形式，以便确立程序中使用何种数据结构，为下一步的算法设计提供依据。

3. 算法设计

结构化算法设计的方法如下：

(1) 自顶向下，逐步求精。所谓自顶向下，就是首先从全局出发进行整体设计，然后，依据整体设计向下逐层分解。所谓逐步求精，就是对上层任务逐层进行细化。一般来说，一个大的任务可以分解为若干个子任务，而每个子任务又可以继续分解为若干个更小的子任务，这样向下逐层细化直至每个子任务仅处理一个简单容易的问题。

(2) 模块化设计。所谓模块化,就是在程序设计时,由自顶向下,逐步求精所得出的一个个子任务的处理程序,称为“功能模块”。一个大的程序,就是由若干个这样的功能模块组成。在整体设计部署下,编程实际上成为若干个小问题的处理。每一块模块可以分配给不同的程序设计者去完成,这样,编程不再是一件十分复杂、十分困难的事情。

由结构化算法得出的功能模块应具备下述特点:

- (1) 一个模块处理一个特定的小问题。
- (2) 每一个模块仅用顺序、选择、循环这 3 种基本结构描述。
- (3) 每一个模块可以独立地进行编程、调试。
- (4) 除最上层外,每层功能模块可接受上层调用。

4. 编程

编程即编写“源程序”,也就是依据算法设计,用高级语言对算法进行描述。

5. 程序编译、调试及运行

编译:在高级语言环境下,对源程序进行编译以检查是否存在语法错误,以便修改。

链接:编译过后,C 语言还要经过链接才能生成可执行文件。

运行、调试该程序,以检查程序是否存在逻辑错误,直至程序运行成功。

下面举例说明结构化程序设计步骤。

【例 1-6】对 50 名学生的成绩进行由高分到低分排序,并打印出来。

- (1) 分析问题:程序的功能是将学生成绩输入计算机,然后降序排列后输出。
- (2) 确定数据类型:学生成绩所使用数据范围为 0~100,其类型为整型数据。

学生的成绩可以存在 50 个整型变量中,也可以存在具有 50 个元素的整型数组中,也可以不设变量按输入成绩的序号直接进行处理。用户可根据所使用的高级语言,本着简单、高效、清晰的原则,选取相应的数据类型。这里选用具有 50 个元素的整型数组来存放每个学生的成绩。

(3) 算法设计:按照自顶向下,逐步求精的处理原则。从整体上看,这一问题可分为 3 部分:

- a. 输入 50 名学生的成绩;
- b. 对 50 名学生成绩进行排序;
- c. 将排序结果输出。

学生成绩模块图如图 1.9 所示。

A	将学生成绩输入 a[1]~a[50]中
B	对学生成绩降序排序
C	输出学生成绩

图 1.9 学生成绩模块图

考虑到每部分解决的问题已经简单且已独立,可将 A、B、C 三部分用功能模块的方法去实现。接着将每一模块逐步细化直至用高级语言中的结构化语句能够描述为止,如图 1.10 中 A 块、B 块、C 块所示(C 语言规定,数组下标从 0 开始,这里定义数组元素为 51 个, a[0]不用)。