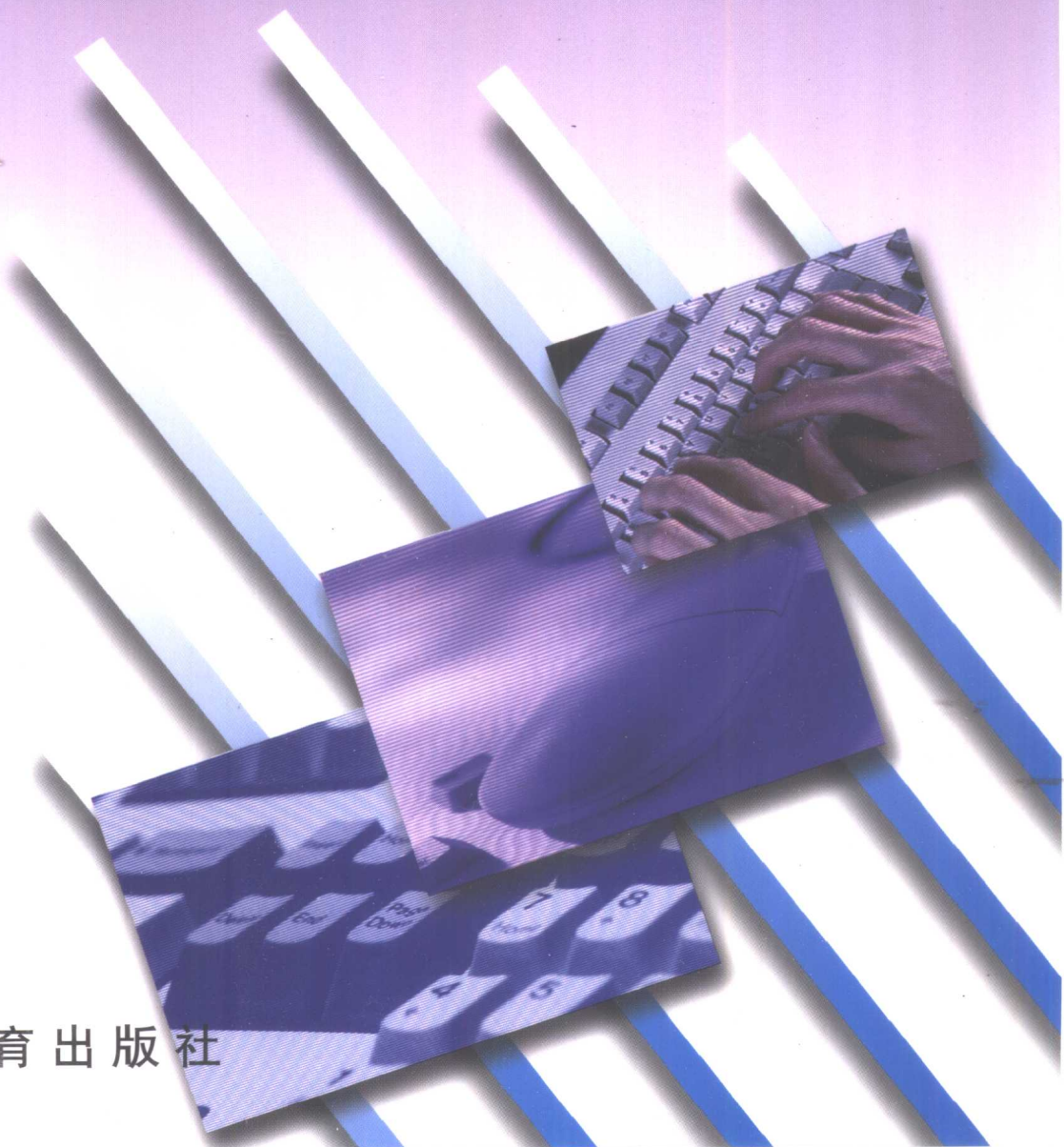


全国计算机等级考试

# 二级教程

## — C 语言程序设计

教育部考试中心



高等教育出版社

# 全国计算机等级考试

## 二级教程

——C 语言程序设计

教育部考试中心

主编 田淑清

编者 田淑清 周海燕 赵重敏

杨明福 王卫兵

高等教育出版社

## 内 容 提 要

本书根据教育部考试中心制定的《全国计算机等级考试二级考试大纲》中有关C语言程序设计的要求而编写,是教育部考试中心组织编写的计算机等级考试系列教程之一。本书主要内容包括:C语言的各种数据类型和运算符,各种表达式,语句结构,函数及库函数,指针,数组,字符串,变量的作用域及存储类,结构体及共用体,文件等。本书内容精炼,结构合理,便于自学,对读者可能遇到的难点做了十分系统、清楚和详细的阐述,极大地减轻了读者学习C语言的困难,是应试人员考前必备的教材。

### 图书在版编目(CIP)数据

全国计算机等级考试二级教程:C语言程序设计/教育部  
考试中心编. —北京:高等教育出版社,1998(2001重印)  
ISBN 7-04-006898-2

I. 全… I. 教… II. ①电子计算机-工程技术人员-水平  
考试-教材②C语言-程序设计 N. TP3

中国版本图书馆CIP数据核字(98)第21303号

---

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街55号

电 话 010-64054588

网 址 <http://www.hep.edu.cn>

邮政编码 100009

传 真 010-64014048

经 销 新华书店北京发行所

印 刷 北京印刷集团有限责任公司印刷二厂

开 本 787×1092 1/16

印 张 17.25

字 数 430 000

版 次 1998年9月第1版

印 次 2001年9月第9次印刷

定 价 24.50元

---

凡购买高等教育出版社图书,如有缺页、倒页、脱页等  
质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

# 第二届全国计算机等级 考试委员会名单

主任委员：杨芙清

副主任委员：（以姓氏笔画为序）

朱三元 杨学为 应书增 罗晓沛 谭浩强

委 员：（以姓氏笔画为序）

王义和	王申康	边冀英	古天祥
齐治昌	仲萃豪	刘澄	刘瑞挺
李克洪	吴文虎	吴功宜	沈钧毅
杨洪	杨明福	林卓然	施伯乐
钟津立	侯炳辉	俞瑞钊	张福炎
袁开榜	席先觉	唐兆亮	徐沪生
钱维民	潘桂明	鞠九滨	瞿坦

秘 书 长：徐沪生（兼）

# 全国计算机等级考试 系列用书编审委员会名单

**主任委员：** 杨芙清

**副主任委员：** 应书增 罗晓沛 谭浩强

**委 员：** (以姓氏笔画为序)

王申康 孙显福 刘瑞挺 吴文虎  
钟津立 唐兆亮 徐沪生 温 波

# 大力推行全国计算机等级考试 为发展知识经济、信息产业和培养 计算机专门人才做出贡献

中国科学院院士

北京大学计算机科学技术系主任

全国计算机等级考试委员会主任委员

杨芙清

当今，人类正在进入一个以智力资源的占有和配置，知识生产、分配和使用为最重要因素的知识经济时代，也就是小平同志提出的“科学技术是第一生产力”的时代。科教是经济发展的基础、知识是人类创新的源泉，基础研究的科学发现，应用研究的原理探索和开发研究的技术发明，三者之间的联系愈来愈紧密，转换周期日趋缩短。世界各国的竞争已成为以经济为基础，以科技特别是以高科技为先导的综合国力的竞争。

在高科技中，信息科学技术是知识高度密集、学科高度综合、具有科学与技术融合特征的学科，它直接渗透到经济、文化和社会的各个领域，迅速改变着人们的观念、生活和社会的结构，是当代发展知识经济的支撑之一。

在信息科学技术中，微电子是基础，计算机硬件及通信设施是载体，计算机软件是核心。软件是计算机的灵魂，没有软件就没有计算机的应用。软件产业已成为信息产业的核心和支柱。信息产业的发展，会大大提高我国的总体实力，增强我国在全球的竞争地位。

为了适应知识经济发展的需要，大力推动信息产业的发展，就需要在全民中普及计算机的基本知识，广开渠道，培养和造就一批又一批能熟练运用计算机和软件技术的各行各业的专门人才。

1994年，国家教委推出了全国计算机等级考试，它是一种重视应试人员对计算机和软件的实际掌握能力的考试。它不限制报考人员的学历背景，任何年龄阶段的人员都可以报考。这就为培养各行业计算机的应用人才，开辟了一条广阔的道路。

1994年是推出计算机等级考试的第一年，当年参加考试的有1万余人；到了1998年上半年，报考人员已达38万余人。截止至1998年上半年，等级考试共开考7次，考生人数累计共达115万人。其中，有49万4千人获得了各级计算机等级证书。

事实说明，鼓励社会各阶层的人士通过多种途径掌握计算机应用技术，并运用等级考试对他们的才干予以认真的、有权威性的认证，是一种较好的人才培养的有效途径，是比较符合我国具体情况的。等级考试也为用人单位录用和考核人员提供了一种评测手段。从有关公司对等级考试所作的社会抽样调查结果来看，不论是管理人员还是应试人员，对该项考试的内容和形式都给予了充分肯定的评价。

计算机等级考试所取得的良好效果，也同全国各有关单位专家们在等级考试的大纲编写、试题设计、阅卷评分及效果分析等等多项工作中，付出的大量心血和辛勤的劳动密切相关，他们为这项工作的顺利开展作出了重要的贡献。

计算机与软件技术是一项日新月异的高新技术。计算机等级考试大纲有必要根据计算机与软件技术在近年的新发展，进行适当的修正，从而使等级考试更能反映当前计算机与软件技术的应用实际，使培养计算机应用人才的基础工作更健康地向前发展。

从面临知识经济的机遇与挑战这样一个社会大环境的背景下，考察全国计算机等级考试，就会看到，这一举措是符合知识经济和发展信息产业的方向的，是值得大力推行的。

我们相信，在 21 世纪知识经济和加快发展信息产业的形势下，在教育部考试中心的精心组织领导下，在全国各有关专家们的大力配合下，全国计算机等级考试一定会以更新的面貌出现，从而为我国培养计算机应用专门人才的宏大事业做出更多的贡献。

# 前 言

随着我国计算机应用的进一步普及和深入，人们已经达成了一个共识：计算机知识是当代人类文化的重要组成部分；计算机应用能力是跨世纪人才不可缺少的素质。因此，许多单位把计算机知识和应用能力作为考核、录用工作人员的重要条件；许多人也在努力证实自身在这方面的实力。人们都在寻求一个统一、客观、公正的衡量标准，各种考试也就应运而生，其中最受欢迎和信赖的是教育部考试中心组织的“全国计算机等级考试”。这一考试自1994年举办以来，应试人数逐年递增。为适应考试的需要，我们根据教育部考试中心制定的《全国计算机等级考试（二级）C程序设计考试大纲》编写了这本等级考试教程。本书紧扣考试大纲，内容取舍得当，是一本系统的应试教材。

本书共分十七章，内容包括：程序设计基本概念、C语言程序设计的初步知识、顺序结构、选择结构、循环结构、字符型数据、函数、指针、数组、字符串、对函数的进一步讨论、C语言中用户标识符的作用域和存储类、编译预处理和动态存储分配、结构体、共用体和用户定义类型、位运算、文件和上机操作。本书的编写力求在体系结构上安排合理、重点突出、难点分散、便于掌握；在语言叙述上注重概念清晰、逻辑性强、通俗易懂、便于自学。根据等级考试要求，考试分为笔试和上机两部分。为便于读者自我检查，书中各章后均配有与笔试题型一致的习题和供读者上机练习用的编程题。

本书由教育部考试中心组织编写并审定。前十六章由田淑清、周海燕、赵重敏合作编写，第十七章由杨明福、王卫兵编写。全书由田淑清统稿。谭浩强教授对本书进行了全面的审阅，并提出了许多宝贵意见；在本书的编写和出版过程中，教育部考试中心和高等教育出版社给予了大力支持，在此，一并表示衷心的感谢。

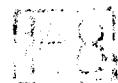
由于我们水平有限，加之时间仓促，书中难免会有许多不足之处。我们真诚希望得到广大读者的批评指正。

作 者

1998.7



责任编辑 鲍 涌  
封面设计 王凌波  
版式设计 焦东立  
责任校对 王 巍  
责任印制 宋克学



# 目 录

## 第一章 程序设计基本概念

1.1 程序和程序设计	1	1.3 结构化程序设计和模块化结构	3
1.1.1 C 程序	1	1.3.1 结构化程序	3
1.1.2 程序设计	2	1.3.2 模块化结构	6
1.2 算法	2	习题	6

## 第二章 C 程序设计的初步知识

2.1 简单 C 语言程序的构成和格式	7	2.4.2 实型变量	13
2.2 常量、变量和标识符	8	2.5 算术表达式	14
2.2.1 标识符	8	2.5.1 基本的算术运算符	14
2.2.2 常量	9	2.5.2 运算符的优先级、结合性和算术表达式	15
2.2.3 用定义一个符号名的方法来代表一个常量	9	2.5.3 强制类型转换表达式	16
2.2.4 变量	10	2.6 赋值表达式	16
2.3 整型数据	10	2.6.1 赋值运算符和赋值表达式	16
2.3.1 整型常量	10	2.6.2 复合的赋值表达式	17
2.3.2 整型变量	11	2.6.3 赋值运算中的类型转换	17
2.3.3 整型数据的分类	11	2.7 自加、自减运算符和逗号运算符	18
2.3.4 整数在内存中的存储形式	12	2.7.1 自加运算符(++ )和自减运算符(-- )	18
2.4 实型数据	13	2.7.2 逗号运算符和逗号表达式	19
2.4.1 实型常量	13	习题	19

## 第三章 顺序结构

3.1 赋值语句	22	3.3.2 scanf 函数中常用的格式说明	27
3.2 数据输出	22	3.3.3 通过 scanf 函数从键盘输入数据	28
3.2.1 printf 函数的一般调用形式	22	3.4 复合语句和空语句	29
3.2.2 printf 函数中常用的格式说明	23	3.4.1 复合语句	29
3.2.3 调用 printf 函数时的注意事项	26	3.4.2 空语句	29
3.3 数据输入	26	3.5 程序举例	30
3.3.1 scanf 函数的一般调用形式	26	习题	31

## 第四章 选择结构

4.1 关系运算和逻辑运算.....	35	break 语句构成的选择结构.....	44
4.1.1 C 语言中的逻辑值.....	35	4.4.1 switch 语句.....	44
4.1.2 关系运算符和关系表达式.....	35	4.4.2 switch 语句的执行过程.....	44
4.1.3 逻辑运算符和逻辑表达式.....	36	4.4.3 在 switch 语句体中使用 break 语句.....	45
4.2 if 语句和用 if 语句构成的选择结构.....	37	4.5 语句标号和 goto 语句.....	46
4.2.1 if 语句.....	37	4.5.1 语句标号.....	46
4.2.2 嵌套的 if 语句.....	40	4.5.2 goto 语句.....	46
4.3 条件表达式构成的选择结构.....	43	习题.....	46
4.4 switch 语句以及用 switch 语句和			

## 第五章 循环结构

5.1 while 语句和用 while 语句构成的循环结构.....	51	5.3.2 for 循环的执行过程.....	55
5.1.1 while 循环的一般形式.....	51	5.3.3 有关 for 语句的说明.....	55
5.1.2 while 循环的执行过程.....	51	5.4 循环结构的嵌套.....	56
5.2 do-while 语句和用 do-while 语句构成的循环结构.....	53	5.5 break 和 continue 语句在循环体中的作用.....	58
5.2.1 do-while 语句构成的循环结构.....	53	5.5.1 break 语句.....	58
5.2.2 do-while 循环的执行过程.....	54	5.5.2 continue 语句.....	59
5.3 for 语句和用 for 语句构成的循环结构.....	54	5.6 程序举例.....	60
5.3.1 for 语句构成的循环结构.....	54	习题.....	63

## 第六章 字符型数据

6.1 字符型常量.....	68	6.3.1 调用 printf 和 scanf 函数输出和输入字符.....	70
6.1.1 字符常量.....	68	6.3.2 调用 putchar 和 getchar 函数输出和输入字符.....	71
6.1.2 转义字符常量.....	68	6.4 程序举例.....	72
6.1.3 字符串常量.....	69	习题.....	74
6.1.4 可对字符量进行的运算.....	69		
6.2 字符变量.....	70		
6.3 字符的输入和输出.....	70		

## 第七章 函 数

7.1 库函数.....	77	7.3 函数的调用.....	80
7.2 函数的定义和返回值.....	78	7.3.1 函数的两种调用方式.....	80
7.2.1 函数定义的语法.....	78	7.3.2 函数调用时的语法要求.....	80
7.2.2 函数的返回值.....	79	7.4 函数的说明.....	81

7.4.1 函数说明的形式	81	7.6 程序举例	84
7.4.2 函数说明的位置	82	习题	89
7.5 调用函数和被调用函数之间的数据传递	82		

## 第八章 指 针

8.1 变量的地址和指针	94	8.4.3 指针比较	99
8.2 指针变量的定义和指针变量的基类型	95	8.5 函数之间地址值的传递	100
8.3 给指针变量赋值	96	8.5.1 形参为指针变量时, 实参和形参之间的数据传递	100
8.3.1 给指针变量赋地址值	96	8.5.2 通过传送地址值, 在被调用函数中直接改变调用函数中的变量的值	101
8.3.2 给指针变量赋“空”值	97	8.5.3 函数返回地址值	102
8.4 对指针变量的操作	97	习题	102
8.4.1 通过指针来引用一个存储单元	97		
8.4.2 移动指针	98		

## 第九章 数 组

9.1 一维数组的定义和一维数组元素的引用	107	引用	124
9.1.1 一维数组的定义	107	9.5.1 二维数组的定义	124
9.1.2 一维数组元素的引用	108	9.5.2 二维数组元素的引用	125
9.1.3 一维数组的初始化	108	9.5.3 二维数组的初始化	125
9.1.4 通过赋初值定义数组的大小	109	9.5.4 通过赋初值定义二维数组的大小	126
9.1.5 一维数组的定义和数组元素引用 举例	109	9.5.5 二维数组的定义和数组元素引用 举例	127
9.2 一维数组和指针	110	9.6 二维数组和指针	128
9.2.1 一维数组和数组元素的地址	110	9.6.1 二维数组和数组元素的地址	128
9.2.2 通过数组的首地址引用数组元素	111	9.6.2 通过地址来引用二维数组元素	129
9.2.3 通过指针来引用一维数组元素	111	9.6.3 通过建立一个指针数组来引用二维数组元素	129
9.2.4 用带下标的指针变量引用一维数组元素	111	9.6.4 通过建立一个行指针来引用二维数组元素	130
9.3 函数之间对一维数组和数组元素的引用	112	9.7 二维数组名和指针数组作为实参	130
9.3.1 数组元素作实参	112	9.7.1 二维数组名作为实参时, 实参和形参之间的数据传递	130
9.3.2 数组名作实参	112	9.7.2 指针数组作为实参时, 实参和形参之间的数据传递	131
9.3.3 数组元素地址作为实参	114	9.8 二维数组程序举例	131
9.3.4 函数的指针形参和函数体中数组的区别	115	习题	135
9.4 一维数组应用举例	115		
9.5 二维数组的定义和二维数组元素的			

## 第十章 字 符 串

10.1 用一个一维字符数组来存放字符串 .....	143	10.3 字符串的输入和输出 .....	147
10.1.1 通过赋初值的方式给一维字符数组赋字符串 .....	144	10.3.1 输入和输出字符串时的必要条件 .....	147
10.1.2 在 C 语言程序执行过程中给一维字符数组赋字符串 .....	145	10.3.2 用格式说明符%s 进行整串输入和输出 .....	147
10.2 使指针指向一个字符串 .....	146	10.3.3 调用 gets、puts 函数, 从终端进行字符串行的输入和输出 .....	148
10.2.1 通过赋初值的方式使指针指向一个字符串 .....	146	10.4 字符串数组 .....	148
10.2.2 通过赋值运算使指针指向一个字符串 .....	146	10.5 用于字符串处理的函数 .....	150
10.2.3 用字符数组作为字符串和用指针指向的一个字符串之间的区别 .....	146	10.6 程序举例 .....	150
		习题 .....	154

## 第十一章 对函数的进一步讨论

11.1 传给 main 函数的参数 .....	159	11.3 函数的递归调用 .....	161
11.2 通过实参向函数传递函数名或指向函数的指针变量 .....	160	习题 .....	164

## 第十二章 C 语言中用户标识符的作用域和存储类

12.1 局部变量、全局变量和存储分类 .....	169	12.3.2 在同一编译单位内用 extern 说明符来扩展全局变量的作用域 .....	173
12.1.1 用户标识符的作用域 .....	169	12.3.3 在不同编译单位内用 extern 说明符来扩展全局变量的作用域 .....	173
12.1.2 局部变量、全局变量和存储分类 .....	169	12.3.4 静态全局变量 .....	174
12.2 局部变量及其作用域和生存期 .....	170	12.4 函数的存储分类 .....	175
12.2.1 auto 变量 .....	170	12.4.1 用 extern 说明函数 .....	175
12.2.2 register 变量 .....	171	12.4.2 用 static 说明函数 .....	175
12.2.3 静态存储类的局部变量 .....	171	习题 .....	175
12.3 全局变量及其作用域和生存期 .....	172		
12.3.1 全局变量的作用域和生存期 .....	172		

## 第十三章 编译预处理和动态存储分配

13.1 编译预处理 .....	179	13.2.1 malloc 函数和 free 函数 .....	182
13.1.1 宏替换 .....	179	13.2.2 calloc 函数 .....	183
13.1.2 文件包含 .....	181	习题 .....	183
13.2 动态存储分配 .....	182		

## 第十四章 结构体、共用体和用户定义类型

14.1 用 typedef 说明一种新类型名	187	14.2.5 函数之间结构体变量的 数据传递	194
14.2 结构体类型	188	14.2.6 利用结构体变量构成链表	197
14.2.1 结构体类型的说明	188	14.3 共用体	203
14.2.2 结构体类型的变量、数组和指针 变量的定义	189	14.3.1 共用体类型的说明和变量定义	203
14.2.3 给结构体变量、数组赋初值	191	14.3.2 共用体变量的引用	204
14.2.4 引用结构体变量中的数据	192	习题	206

## 第十五章 位 运 算

15.1 位运算符	211	习题	214
15.2 位运算符的运算功能	211		

## 第十六章 文 件

16.1 C 语言文件的概念	216	16.8 fgets 函数和 fputs 函数	223
16.2 文件指针	217	16.9 fread 函数和 fwrite 函数	223
16.3 打开文件	217	16.10 文件定位函数	224
16.4 关闭文件	219	16.10.1 fseek 函数	225
16.5 调用 getc ( fgetc ) 和 putc ( fputc ) 函数进行输入和输出	219	16.10.2 ftell 函数	225
16.6 判文件结束函数 feof	221	16.10.3 rewind 函数	226
16.7 fscanf 函数和 fprintf 函数	222	习题	226

## 第十七章 上 机 指 导

17.1 上机考试系统使用说明	229	17.1.7 文件名的说明	236
17.1.1 上机考试环境	229	17.2 上机考试内容	236
17.1.2 上机考试时间	230	17.2.1 DOS 常用命令操作	236
17.1.3 上机考试题型及分值	230	17.2.2 程序修改调试运行	238
17.1.4 上机考试登录	230	17.2.3 程序编制调试运行	242
17.1.5 试题内容查阅工具的使用	233	习题	245
17.1.6 考生目录和文件的恢复	235		

## 附 录

附录一 C 语言的关键字	254	附录三 运算符的优先级和结合性	254
附录二 双目算术运算中两边运算量类型 转换规律	254	附录四 常用字符与 ASCII 代码对照表	255
		附录五 库函数	256

# 第一章 程序设计基本概念

## 1.1 程序和程序设计

### 1.1.1 C 程序

当今,计算机已广泛应用于社会生活的各个领域,成为大众化的现代工具。但是,不熟悉计算机的人仍然把它想象得十分神秘。其实,计算机不过是一种具有内部存储能力、由程序自动控制的电子设备。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,也就是我们常说的“程序设计语言”。

目前,在社会上使用的程序设计语言有上百种,它们都被称为计算机的“高级语言”,如 BASIC、PASCAL 以及本书将介绍的 C 语言等。这些语言都是用接近人们习惯的自然语言和数学语言做为语言的表达形式,人们学习和操作起来感到十分方便。

但是,对于计算机本身来说,它并不能直接识别由高级语言编写的程序,它只能接受和处理由 0 和 1 的代码构成的二进制指令或数据。由于这种形式的指令是面向机器的,因此也称为“机器语言”。

我们把由高级语言编写的程序称为“源程序”,把由二进制代码表示的程序称为“目标程序”。如何把源程序转换成机器能够接受的目标程序,软件工作者编制了一系列软件,通过这些软件可以把用户按规定语法写出的语句一一翻译成二进制的机器指令。这种具有翻译功能的软件称为“编译程序”。每一种高级语言都有与它对应的编译程序。例如,C 语言编译程序就是这样的一种软件,C 语言编译程序的功能如图 1.1 所示。

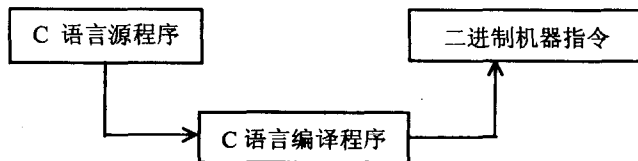


图 1.1

我们所写的每条 C 语句,经过编译(Compile)最终都将转换成二进制的机器指令。由 C 语言构成的指令序列称 C 源程序,按 C 语言的语法编写 C 程序的过程,称 C 语言的代码编写。

C 语言源程序经过 C 语言编译程序编译之后生成一个后缀为 .OBJ 的二进制文件(称为目

标文件)。最后还要由称为“连接程序”(Link)的软件,把此 .OBJ 文件与 C 语言提供的各种库函数连接起来生成一个后缀为 .EXE 的可执行文件。在 DOS 状态下,只须打入此文件的名字(而不必打入后缀 .EXE),该执行文件就可运行。

### 1.1.2 程序设计

简单的程序设计一般包含以下几个部分。

1. 确定数据结构。根据任务书提出的要求、指定的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。针对存放数据的数据结构来确定解决问题、完成任务的一步一步的步骤。有关算法的概念将在下一节中介绍。
3. 编码。根据确定的数据结构和算法,使用选定的计算机语言编写程序代码,输入到计算机并保存在磁盘上,简称编程。
4. 在计算机上调试程序。消除由于疏忽而引起的语法错误或逻辑错误;用各种可能的输入数据对程序进行测试,使之对各种合理的数据都能得到正确的结果,对不合理的数据能进行适当的处理。
5. 整理并写出文档资料。

## 1.2 算 法

学习计算机程序设计语言的目的,是要用语言作为工具,设计出可供计算机运行的程序。

在拿到一个需要求解的问题之后,怎样才能编写出程序呢?除了选定合理的数据结构外,一般来说,十分关键的一步是设计算法,有了一个好的算法,就可以用任何一种计算机高级语言把算法转换为程序(编写程序)。

算法是指为解决某个特定问题而采取的确定且有限的步骤。一个算法应当具有以下五个特性:

1. 有穷性。一个算法应包含有限个操作步骤。也就是说,在执行若干个操作步骤之后,算法将结束,而且每一步都在合理的时间内完成。
2. 确定性。算法中每一条指令必须有确切的含义,不能有二义性,对于相同的输入必能得出相同的执行结果。
3. 可行性。算法中指定的操作,都可以通过已经实现的基本运算执行有限次后实现。
4. 有零个或多个输入。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。
5. 有一个或多个输出。算法的目的是为了求“解”,这些“解”只有通过输出才能得到。算法可以用各种描述方法来进行描述,最常用的是伪代码和流程图。

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,这在英语国家中使用起来更为方便。

流程图也是描述算法的很好的工具,传统的流程图由图 1.2 中所示的几种基本框组成。



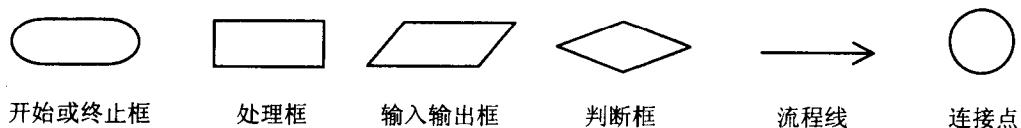


图 1.2

由这些框和流程线组成的流程图来表示算法，形象直观，简单方便，但是这种流程图对于流程线的走向没有任何限制，可以任意转向，在描述复杂的算法时所占篇幅较多，费时费力且不易阅读。

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 流程图。我们将在下一节中结合结构化程序设计中的三种基本结构来介绍这种流程图的基本结构。

本书中的例题，由于比较简单，因此大都采用文字来进行描述，但要求读者能读懂这两种流程图。

## 1.3 结构化程序设计和模块化结构

### 1.3.1 结构化程序

结构化程序由三种基本结构组成。

1. 顺序结构。在本书第三章中将要介绍的如赋值语句，输入、输出语句都可构成顺序结构。当执行由这些语句构成的程序时，将按这些语句在程序中的先后顺序逐条执行，没有分支，没有转移。顺序结构可用图 1.3 所示的流程图表示，(a) 是一般的流程图，(b) 是 N-S 流程图。

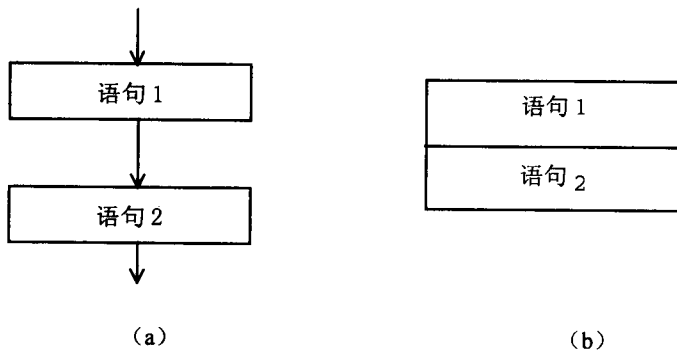


图 1.3

2. 选择结构。在本书第四章中将要介绍的 if 语句、switch 语句都可构成选择结构。当执