依据教育部考试中心 1998 年
制定考试大纲编写

最新计算机等级考试(二级)用书

C语言程序设计

龚圆明　朱玉文　刘万春　编

国防工业出版社

依据教育部考试中心 1998 年制定考试大纲编写

最新计算机等级考试(二级)用书

C 语言程序设计

龚圆明　朱玉文　刘万春　编

国防工业出版社

·北京·

图书在版编目(CIP)数据

C 语言程序设计/龚圆明等编．—北京：国防工业出版社，1999.6

最新计算机等级考试二级用书

ISBN 7-118-02114-8

Ⅰ.C… Ⅱ.龚… Ⅲ.C 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(1999)第 09230 号

国防工业出版社出版发行

（北京市海淀区紫竹院南路 23 号）

（邮政编码 100044）

北京怀柔新华印刷厂印刷

新华书店经销

*

开本 787×1092 1/16 印张 18½ 424 千字

1999 年 6 月第 1 版 1999 年 6 月北京第 1 次印刷

印数：1—4000 册 定价：25.00 元

出版者的话

掌握和使用电脑已是现代人综合能力的重要组成部分，近几年来，参加计算机等级考试的人数逐年递增。

教育部考试中心 1998 年制定的计算机等级考试大纲，对考试内容及要求重新做了说明。根据这一变化，我们出版了最新计算机等级考试（二级）用书（共 6 册）：

计算机基础知识（DOS、Windows）

QBASIC 程序设计

Pascal 程序设计

C 语言程序设计

FORTRAN 程序设计

FoxBASE 数据库管理系统

同时为帮助读者顺利通过等级考试，每册图书均配有基于 Windows 95/98 环境编写的等级考试软件，具有电脑自动组卷、阅卷、评分及用户自行扩充试题数目、修改试题难易度等诸多功能。

本套书由北京理工大学计算机系教师编写，参加编写人员为教授、副教授，均为相应课程主讲教师，等级考试辅导班主讲教师，有多年教学经验。

本套书内容紧扣新大纲、详实、极具针对性，并附有大量习题和参考答案，是参加 1999 年及以后计算机等级考试人员理想用书。

我们真诚地期望，本套书及配套软件能为您的考试顺利过关助一臂之力。

前　言

C语言是近年来国内外广泛流行并得到迅速推广使用的一种计算机程序设计语言。C语言功能丰富、表达能力强、使用灵活方便，它是计算机等级考试(二级)中近年来参加人数最多的一门科目。

本书重点放在培养读者良好的程序设计风格和习惯，力求做到科学性、实用性和通俗性的统一。根据作者多年C语言程序设计方面的教学经验，对读者易于混淆的概念做了重点说明，采用循序渐进，简明扼要、重点突出的编写方式。

本书共11章：第一章概述，第二章基本数据类型和表达式，第三章简单的程序设计，第四章分支结构，第五章循环控制，第六章函数与变量类型，第七章数组，第八章指针，第九章结构体和共用体，第十章文件及第十一章常用错误分析和程序调试。基本覆盖了教育部考试中心制定的计算机等级考试(二级——C语言程序设计)考试要求。为便于读者自学，各章后附有小结并配有适量习题，并且给出参考答案。

学习计算机语言的目的是利用它来编写程序解决实际中出现的问题，因此，希望读者不应把主要精力花费在死记硬背语言的规则上，而应把重点放在程序设计上，即学会使用C语言去进行程序设计。

在本书编写过程中得到北京理工大学赵鸿德教授的大力帮助，也得到国防工业出版社的大力支持，在此编者表示深切的谢意。参加本书编写工作的还有：刘俐、邢鹏、徐一华、夏军、李斌、郑红霞、谢玉清、龚小舟、顾伟、张飞、谢小叶、李小刚、李东升、赵亮等同志。

由于编者水平有限，不妥之处在所难免，恳切希望读者提出宝贵意见，以便改进。

编　者

内 容 简 介

本书是依据教育部考试中心1998年制定的计算机等级考试大纲(C 语言程序设计)考试要求编写的。

本书系统地介绍了 C 语言的程序设计方法和技巧,对 C 语言各部分内容作了合理安排,由浅入深,循序渐进,概念清晰,通俗易懂。为便于读者自学,在各章后作了小结并附有适量的习题,并给出参考答案供读者参考。

本书是为参加1999年及以后计算机等级考试的读者编写的,也可作为大专院校学生参加高等自学考试或文凭考试的教学用书,还可作为科技人员学习 C 语言的参考。

目　录

绪　论

第一章　C语言概述

第二章　基本数据类型和表达式

第三章 简单的C语言程序设计

第四章 分支结构

第五章　循环控制

第六章　函数与变量类型

第七章　数　组

第八章　指　针

第九章　结构体与共用体

第十章 文 件

第十一章 常见错误分析和程序调试

绪　论

0.1　计算机语言的发展

现代科学的迅猛发展使电子计算机几乎进入了人类生活的各个领域，计算机成为人类必不可少的有力助手。

众所周知，计算机内部采用二进制工作，目前人和计算机还不能像人和人之间那样完全用自然语言进行交流，计算机和人之间的交流需借助于“计算机语言”。

计算机内部完全用二进制，采用0、1进行工作。例如，指令1000101011010000表示把寄存器AL中内容移入到寄存器DL中。由0和1组成的指令序列（程序）称为机器代码，计算机直接用机器代码工作速度快、效率高，但是人们很难记忆和理解这些机器代码，有时不得不依靠八进制和十六进制码来帮助记忆。例如，上面16位二进制数可以用4位十六进制码$(8AD0)_H$来表示，这种完全由0和1组成的二进制信息称为机器语言。这种语言不仅难学难记而且没有通用性，不同型号的计算机机器语言完全不同。

鉴于机器语言的缺陷，人们发展了汇编语言，并采用一些助记符来代替机器代码。例如，上面指令可表示为“MOV DL，AL”，但汇编语言仍保留着机器语言的弊病，没有摆脱具体机器的依赖性，因此称为面向机器的语言。

随着计算机的发展，急需要解决的是计算机硬件的高速度和程序编制的低效率之间的矛盾，在50年代末期出现了“高级程序设计语言”，它较为接近自然语言，具有易学易懂的特点。更重要的是，它是面向用户的语言，当学会了一种高级语言后，在各种类型的计算机上都能使用（但也略有差异）。

自从高级语言问世以来，已出现过上千种程序设计语言，通常可以分为通用型和专用型两大类。具体划分又可分为适合于数值计算的语言（例如ALGOL－60语言、FORTRAN语言）；结构化程序设计语言（例如Pascal语言、C语言）；适合商用和管理领域的语言（例如COBOL、FoxBASE、FOXPRO语言），还有一些交互式的通用语言（例如BASIC，APL语言）。而专用语言更是种类繁多、功能各异。例如适合数控机床工作的数控语言APL，适用于计算机辅助设计的AHPL和DDL语言，适合符号处理的LISP语言，适合于人工智能的PROLOG语言，适合于系统分析的PSL/PSA语言等。另外还有综合各类语言特点、功能强大，适用范围较广的汇集性语言，如ADA和PL/I语言。

面对这么多的高级语言，要全面掌握它是不可能的。事实上在实际计算机应用中常用的语言才十几种，像BASIC、FORTRAN、Pascal、C、FoxBASE、FOXPRO等语言，因此作为初学者应该把精力集中到这些语言上，特别应以一种语言为模板，深入学习和应用，掌握该语言的基础、结构及编程等技术，这样才能为学习其他高级语言打下基础。

0.2 程序的编译和解释

计算机只能执行机器代码(二进制代码),而人是用汇编语言或高级语言来编写程序。因此要在计算机上运行所写的程序,首先就要把这些程序"翻译"为机器代码,在翻译过程中如果发现所编写的程序(称为源程序)有语法或语义错误,就应给出提示,要求程序人员对程序进行必要的修改。

"翻译"程序通常由计算机供应商提供,装载在计算机中,它分为"解释"和"编译"两大类,图 0-1 表示这两种类型的执行过程。

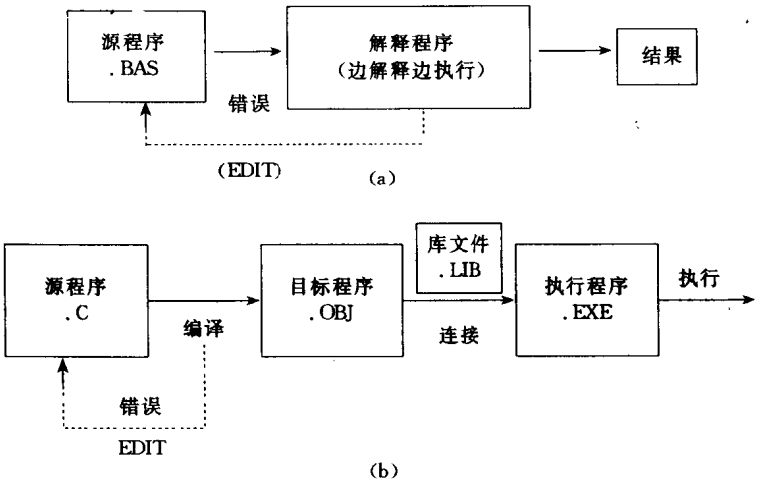

图 0-1 解释过程和编译过程

(a)解释过程; (b)编译过程。

早期的 BASIC 语言都采用"解释"方式,它调用计算机上配备的 BASIC 解释程序,在运行 BASIC 源程序时,逐条对 BASIC 源程序语句进行解释和执行(发现有错误时停止执行,要求用户修改源程序),它不保留机器码的目标程序,不产生.EXE(可执行)文件。这种方式速度慢,每次运行都要对源程序解释一遍,边解释边执行,见图 0-1(a)。

目前绝大多数语言采用"编译"方式,首先调用计算机上配备的编译程序,把源程序变成目标程序(.OBJ 为扩展名),然后再调用连接程序使目标程序与库文件相连形成可执行文件(.EXE 为扩展名)。尽管编译过程复杂一些,但形成.EXE 文件之后,可以反复执行,.EXE 文件运行速度快(只需打入文件名,不需打入扩展名),见图 0-1(b)。

0.3 程序设计的概念

0.3.1 算法及其表达

一般而言,对解题过程准确而完整的描述称为解此问题的算法,因此不知道解题的算法也就不可能编写程序来解题。例如,要求一个圆的周长和面积就必须知道"周长$=2\pi R$"及"面积$=\pi R^2$"两个公式,根据这两个公式就可以列出计算圆的周长和面积的算法。

【例 0-1】表达计算和打印两数之和及平均值的算法。

算法：1)从键盘输入两个数 N1 和 N2

2)计算两数之和　S=N1+N2

3)计算两数平均值　V=S/2

4)打印 S 和 V。

本题涉及 4 个操作，其控制方式是 1)～4)顺序执行的。

【例 0-2】给出职工工作小时数及小时工资率，计算职工的工资及实发工资，若职工工资超过 800 元应扣除超出部分的 20%作为个人所得税。

算法：1)读入职工工作小时数 HOURS 和小时工资率 RATE。

2)计算职工工资 GROSS=HOURS * RATE(* 为乘号)

3)计算实发工资 NET

如果 GROSS>800

则　NET=GROSS-(GROSS-800) * 0.2

否则　NET=GROSS

4)打印出 GROSS 及 NET。

本例涉及的 4 个操作也是顺序的，但第 3)步有一个“选择”控制，即它的执行取决于第 2)步计算出来的 GROSS 值，当 GROSS 超过 800 元要扣除个人所得税，否则不扣除。

【例 0-3】计算 e^x 和 lnx 值，其中 x=1,2,3,…,100。

算法：FOR x=1 TO 100 DO

begin

1)计算 $y=e^x$

2)计算 z=lnx

3)打印 x,y,z 的值

end

这里 FOR 表示一种“循环”，x 依次由 1 增加到 100，从 begin 到 end 之间称为“循环体”。这个循环体重复执行 100 次，每次依次执行 1)、2)、3)三步，这种结构称为“循环控制”，当然，循环次数应该是有限的，否则会出错。

算法表达通常有程序流程图，N-S 图、PAD 图等方式，但目前应用最广泛的是程序流程图。

图 0-2 表示程序流程图的基本符号，图 0-3 表示方程 $ax^2+bx+c=0$ 的求解算法。

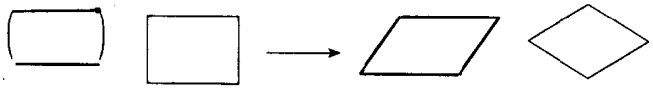

图 0-2　程序流程图基本符号

人们普遍采用程序流程图是因为它简单、直观。把执行的流程表达得一清二楚，易于理解。然而，它也有不少缺陷，程序流程图中的箭头十分灵活，使用不当会影响到程序结构的清晰性。

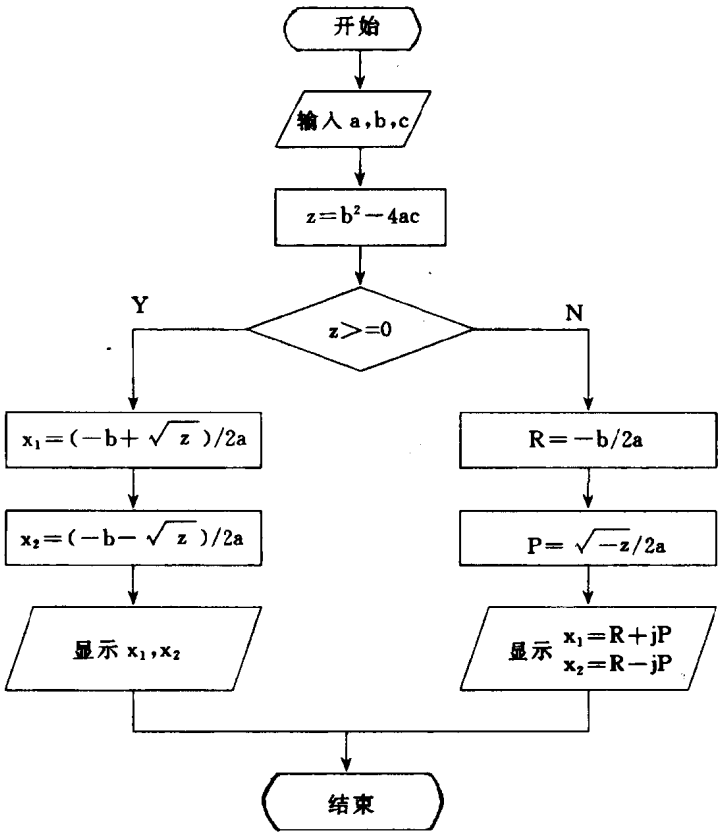

图 0-3　方程 $ax^2+bx+c=0$ 的求解

0.3.2　结构化程序设计

结构化程序设计的提出是从 goto 语句(转向语句)的使用而引起的，goto 语句可以使程序员随心所欲地从程序的一处转到另一处，充分发挥程序员的“技巧”，但过多使用 goto 语句会使程序结构十分零乱，流程一会儿向前，一会儿向后，令人眼花缭乱，顾此失彼，程序可读性差，难以阅读和理解，因此有的学者建议在程序中禁用 goto 语句，从而引起争论。1966 年 C. Bobm 提出任何程序都可以用顺序、选择、循环三种基本结构来组合，这样编写出来的程序易懂易读也易于修改，提高了程序可靠性。这样的程序称为结构化程序，编写这样的程序称为结构化程序设计。目前大多数计算机语言(例如 Pascal、C 语言)都是结构化程序设计语言，这些语言中没有完全禁用 goto 语句，但要求使用时十分谨慎，不要跳得太远，并且只限于一个结构内部使用，不允许从一个结构跳转到另一个结构中。

结构化程序设计的另一个概念是模块化设计，把一个大的复杂的问题逐层分解为一系列小的简单的模块来进行处理，每个小模块只完成单一的具体任务。模块内部联系紧密，而与其他模块之间联系较弱，这样的模块称为独立性高的模块。

独立性高的模块有以下优点：

1)模块之间接口关系简单，每个模块完成独立的工作，易于被人理解，所编写出来的程序可读性和可理解性好；

2)各个模块功能单一，要修改某一个模块时只涉及到要修改的模块而不涉及到其他

模块，不会出现修改程序时牵一发动全身的现象；

3)人们可以单独验证、测试一个个模块，保证其正确性；

4)可以利用现有模块，像搭积木一样进行开发新的程序。

模块之间是一种层次结构，上层模块对下层模块进行调用，图 0-4 所示是报表生成程序的层次结构，以柜形框表示模块，框中的模块名称表明模块的功能，提出顶层的模块“做什么”而不涉及“怎样做”，最下层模块功能十分具体，涉及“怎样做”的精确描述，易于编程。

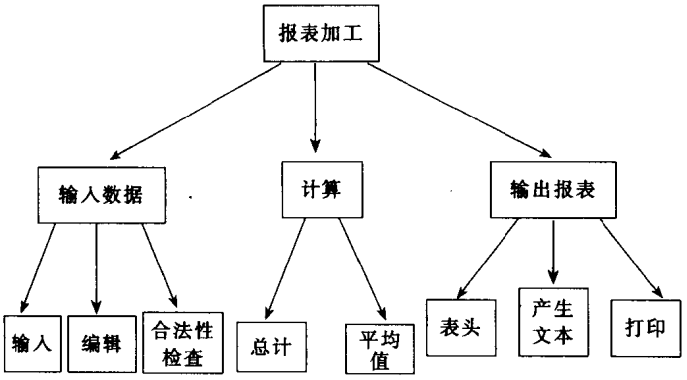

图 0-4　报表生成程序层次结构