

等级考试
C语言程序设计

最新计算机等级考试教程

【二级】

新大纲，新考点，新法宝
——助您顺利过关，取证



语言 程序设计



黄桂钦
于中华
高元鹏 等 编著

312
7Q/1



机械工业出版社
China Machine Press

TP312
HCTG/1

最新计算机等级考试教程

(二级)

C语言程序设计

黄桂钦 于中华 高元鹏 等编著



机械工业出版社
China Machine Press

052626

本书根据国家教育部制定的全国计算机等级考试二级考试大纲的要求编写,其深度和广度符合考试大纲的要求,可作为全国计算机等级考试(二级)C语言程序设计教材。

本书共分8章,分别介绍了C语言的简单数据类型、运算符和表达式;C语言的语句;C语言的构造数据类型;函数;编译预处理;指针;文件。为了帮助读者熟练掌握各章内容,每章都附有习题。

本书既可供从事计算机研究和应用的人员使用,也适合于广大计算机编程爱好者和初学者自学和参考。更是参加等级考试人员的必备参考书。

本书中文简体字版由机械工业出版社出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

版权所有,翻印必究。

图书在版编目(CIP)数据

JS256/24

C语言程序设计/黄桂钦等编著 - 北京:机械工业出版社,1999.9

最新计算机等级考试教程(二级)

ISBN 7-111 07368-1

I. C... II. 黄... III. C语言 - 程序设计 - 水平考试 - 教材 IV. TP312

中国版本图书馆CIP数据核字(1999)第29336号

出版人:马九荣(北京市百万庄大街22号 邮政编码 100037)

责任编辑:郭东青

北京市密云县印刷厂印刷·新华书店北京发行所发行

1999年9月第1版·1999年10月第2次印刷

787mm × 1092mm 1/16 · 11.5 印张

印数:6 001-9 000 册

定价:19.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

最新计算机等级考试教程编委会

主任委员 史济民

副主任委员 李光琳 周启海

秘书长 白晓毅

副秘书长 王 松

委 员 于中华 冉蜀阳 揭金良 廖 果

黄桂钦 李 薇 董 恒 高元鹏

赵进延 李自力 杨祥茂 丁 庆

陈 康 文 进 张 维 黄晓榕

序 言

计算机技术是近 20 年来发展最迅猛、应用最广泛的现代科学技术之一，是 20 世纪信息技术产业的关键核心技术，21 世纪知识经济时代的重要基础技术。在当今社会中，人人都应当掌握这一基本的生存技能与必备的文化素质。因此，普及推广和开发应用计算机技术，对国家的生存和发展，具有不可低估的社会经济价值和历史进步意义。

当今在我国开展的“全国计算机等级考试”，对推动全社会学习计算机技术具有不容轻视的影响。计算机技术的开发与应用，说到底，无非是在普及计算机基础知识和基本操作的前提下，进行计算机程序的设计与运用。因此，“如何使学生轻松、愉快地掌握计算机基础知识和基本操作以及学好计算机程序设计，怎样让教师愉快、轻松地教好计算机基础知识和计算机程序设计”，自然地成为“全国计算机等级考试”教材要实现的重要目标，是值得探索与实践的重大课题。

为此，我们根据教育部考试中心 1998 年新颁布的“全国计算机等级考试大纲”，组织富有教学经验的计算机基础课教师编写了这一套系列教材，作为全国计算机等级考试的教学（包括自学）用书。本系列教材深入浅出地讲明了大纲规定的一、二、三级的应知应会内容；同时，编者通过各种类型的模拟试题，立足根本，引导读者掌握审题技术，揭示解题思路，着力于帮助读者逐步提高解题能力和应试技巧。希望读者在学习这些教材时，不要死记硬背，多从理解上下功夫，以达到融会贯通。我们衷心希望这些教材能成为广大考生和计算机爱好者的良师益友，在教学中收到事半功倍的效果。

史济民

1999 年 2 月

前 言

当前，我们已经进入信息时代，计算机和信息技术空前发展，渗透和影响社会生活的各个领域。为了迎接信息时代的挑战，成为跨世纪的建设人才，具有计算机的应用能力是必不可少的。

为了推动计算机应用的普及和深入，对人们的计算机应用水平进行统一、客观、公正的评价，国家教育部考试中心从1994年开始组织“全国计算机等级考试”，并于1998年在总结前几年“等级考试”经验教训的基础上，结合计算机发展的最新形势，对“计算机等级考试”大纲进行了修订。本书就是紧扣最新的考试大纲，组织有经验的专家学者编写的一本应试教材。

本书共分八章：其中第1、2、3、7、8章由港澳证券公司高元鹏工程师、最高人民检察院检察技术科学研究所信息技术室赵进延工程师与四川大学计算机系于中华副教授合作编写，第4章和第5章由四川大学计算机系黄桂钦老师编写，第6章由四川大学计算机系左劭老师编写。在本书的编写和出版过程中，机械工业出版社西南组稿中心给予了大力支持，在此，作者表示衷心的感谢。

在编写本书时作者力求体系结构安排合理、重点突出，语言叙述通俗易懂、逻辑性强，以便读者自学掌握。但由于水平有限，时间仓促，书中难免有不足甚至谬误之处。我们真诚希望读者提出宝贵意见，以便改进修订。

作者
1999年6月

目 录

序言	
前言	
第1章 引论	1
1.1 高级程序设计语言概论	1
1.2 C语言特点	3
1.3 C语言程序结构简介	4
1.4 计算机运行C程序的过程	6
1.5 学习《C语言程序设计》的要点	8
第2章 C语言的简单数据类型、运算符和表达式	10
2.1 C语言数据类型概要	10
2.2 标识符	10
2.3 整数类型	10
2.4 实数类型	12
2.5 字符类型	13
2.6 算术运算符及算术表达式	15
2.7 强制类型转换运算符	18
2.8 关系运算符及关系表达式	19
2.9 赋值运算符(=)及赋值表达式	19
2.10 逗号运算符(,)及逗号表达式	20
2.11 逻辑运算符和逻辑表达式	21
2.12 位运算符及其构成的表达式	21
2.13 条件运算符及条件表达式	23
2.14 小结	24
2.15 习题	24
第3章 C语言的语句	25
3.1 C语言语句概述	25
3.2 表达式语句	25
3.2.1 赋值语句	26
3.2.2 数据输入输出	26
3.3 IF语句	29
3.4 SWITCH语句	31
3.5 GOTO语句	32
3.6 WHILE循环语句	32
3.7 FOR循环语句	33
3.8 DO-WHILE语句	34
3.9 CONTINUE语句	35
3.10 BREAK语句	36
3.11 小结	36
3.12 习题	36
第4章 C语言的构造数据类型	38
4.1 数组	38
4.1.1 一维数组	38
4.1.2 二维数组	41
4.1.3 字符数组和字符串	45
4.2 结构体	53
4.2.1 结构体类型变量的定义	53
4.2.2 结构类型变量的引用(引用结构类型变量)	55
4.2.3 结构变量的初始化	56
4.2.4 结构数组	57
4.2.5 结构指针	61
4.2.6 位结构	62
4.3 共同体	63
4.4 枚举	66
4.5 小结	68
4.6 习题	69
第5章 函数	72
5.1 函数的定义	72
5.2 函数调用	73
5.2.1 函数调用的一般形式	73
5.2.2 函数调用方式及函数的返回值	75
5.2.3 函数参数的传递方式	81
5.3 函数调用的嵌套与递归	88
5.3.1 函数调用的嵌套	88
5.3.2 函数的递归调用	89
5.4 变量的作用域与存储类别	93
5.4.1 变量的作用域	93

5.4.2 变量的存储类别	98	7.3 指针可以进行的运算	125
5.5 内部函数和外部函数	106	7.4 指针的使用	130
5.6 小结	107	7.5 小结	162
5.7 习题	108	7.6 习题	162
第6章 编译预处理	112	第8章 文件	166
6.1 宏定义	112	8.1 C文件概述	166
6.1.1 无参数宏定义	112	8.2 对文件处理的基本步骤及文件类型变量 的说明	167
6.1.2 带参数的宏定义	115	8.3 文件处理函数	168
6.2 文件包含	117	8.4 程序举例	173
6.3 条件编译	120	8.5 小结	175
6.4 小结	122	8.6 习题	175
第7章 指针	124	附录 C 的运算符和结合性	176
7.1 指针类型变量的说明格式	124		
7.2 指针的直观意义	124		

第1章 引 论

C语言是获得广泛使用的一种高级程序设计语言。本章首先概述高级程序设计语言的一般概念和知识,然后介绍C语言的特点,并通过几个C程序实例,说明C程序的结构特征,最后给出在计算机上运行C程序的操作步骤和学习C语言程序设计的线索。由于本章是引论性的,必然涉及全书的后续内容,因此,读者学习本章时,可能对其中某些内容不理解或理解不深入,这很正常,随着本书后续内容的逐步展开,引论部分的内容就会变得越来越容易理解。

1.1 高级程序设计语言概论

语言是交流思想的工具。两个行为主体要达到思想沟通和信息传递的目的,必须借助于某种语言来进行对话。如,汉语和英语等都是人与人之间交流的工具,这些语言不是人为创造的,而是人类在长期发展过程中逐步形成的,因此被称为自然语言。与自然语言对应的是人工语言,这些语言是为某些特殊交流目的而人为设计的,其中就包括程序设计语言。程序设计语言属于一种人工语言,它是为人和计算机之间进行交流而专门设计的。我们用计算机解决问题,必须首先把问题本身及解决该问题的步骤方法(称为算法)用某种计算机能够理解的形式表达出来,交给计算机,然后计算机才能按照我们的要求,经过规定步骤的处理,完成交给它的任务。计算机完成任务后,还需要把结果用约定的形式告诉给人。因此,人和计算机之间也需要进行沟通交流,也需要语言,这种语言就是程序设计语言。用程序设计语言来描述问题及其算法的过程,叫做程序设计。程序设计的结果,即用程序设计语言形式表达的问题及其算法,就是程序。

计算机从诞生之日起,就存在与人进行交流对话的问题。在计算机刚刚出现时,人机对话采用的是机器语言形式。当时人要使用计算机,必须用机器语言来编制自己的程序,在这样的程序中,问题的数据和算法一律被变成二进制的形式来表达。我们知道,计算机是能够直接理解和执行二进制形式的指令和数据的,即计算机“天生”就懂得机器语言,因此,机器语言程序对计算机来说非常方便,可以直接执行,不需要在执行之前进行特殊的处理。但是,人“天生”只懂自然语言,机器语言与自然语言相距甚远,所以,机器语言对人非常不方便。用户为了能够使用机器语言与计算机沟通,必须专门花费很多时间和精力来学习机器语言,然后才能比较熟练地用机器语言来表达自己的问题和算法。而且由于机器语言对人来说太不自然、直观,因此,用机器语言设计程序效率非常低,设计出来的程序可读性非常差,容易出错,出错又难以修改。由于机器语言的这些特性,使得当时的计算机应用范围非常窄,只被专业人员用来进行简单的科学计算。可以说,早期计算机未能获得广泛应用,主要就是由于人机对话的语言对人来说太专业化、太不自然造成的。

为了使得计算机能够获得广泛应用,人们需要提出一种新的人机对话语言,这种语言不应该像机器语言那样一味照顾机器而忽略了参与对话的另一方——人。也就是说,为了使普通人也能很容易学会使用计算机,必须采用与自然语言接近的语言来实现人机对话。在这种需求的推动下,诞生了一种比机器语言更接近自然语言的语言——汇编语言。

汇编语言其实是机器语言的形式化表达。汇编语言的每一条指令，都唯一地对应了一条机器语言的指令，只不过在汇编语言中，指令的操作码都采用有意义的符号串形式(称为助记符)来表达，操作数可以采用人们熟悉的十进制(八进制或十六进制)形式来表达，内存地址和寄存器也可以采用有意义的符号串来表达。这样，使得汇编语言与机器语言比较起来，既容易学习记忆，又容易使用，编程效率和程序的可读性都大大提高，程序调试排错也变得容易。当然，计算机是不能直接运行汇编语言程序的，因为它“天生”只懂机器语言，为此，需要借助于专门的软件工具——汇编系统，把用汇编语言写的程序翻译成机器语言程序，然后将翻译结果——机器语言程序交给计算机运行。汇编语言比机器语言虽然有很大的进步，但是，汇编语言与机器语言比较，只是指令不再用机器码来表示，而是用助记符，没有本质飞跃，因此，为了用汇编语言设计程序，人们同样要花费很多时间来记忆汇编指令，并学习诸如地址、存储器等这样的专业概念和专业知识，对于非计算机专业人员是非常困难的。

为了进一步提高程序设计的效率和质量，提出了高级程序设计语言(简称高级语言)，如BASIC、PASCAL、COBOL、C语言等等。高级语言与汇编语言和机器语言相比，有如下特点：

(1) 高级语言比汇编语言更加贴近人的语言，其表达式与数学公式类似，语句和数据定义格式类似于自然语言的语句(保留关键字均为完整的英语单词)，而且高级语言把与机器底层硬件相关的细节全部隐藏起来，使用者不需要具有计算机底层的概念和知识，所以，学习高级语言变得非常容易，即使没有计算机基础的人也可以比较轻松地掌握。

(2) 汇编语言和机器语言的每条指令代表一个最基本的计算机操作动作，用汇编语言和机器语言设计程序，必须把要解决问题的算法分解为细小的操作序列，然后才能用汇编或机器指令表达出来，高级语言的每个语句描述的不是一个计算机最基本的动作，而是一个计算机基本动作序列，即高级语言的一个语句比机器和汇编语言的一条指令描述能力要强得多，用高级语言设计程序，只需要将解决问题的算法分解为能够用语句描述就可以了，不需要再细分，因此，用高级语言设计程序效率大大提高。

(3) 用高级语言写出来的程序可读性好，出错容易修改，因为高级语言接近人的语言，容易阅读理解，并且现今的许多高级语言都是结构化语言，用它们设计出来的程序具有层次分明的结构，语句也是结构化的，这样，一方面降低了程序设计出错的机会，另一方面也使得错误的定位和排除变得容易。当然，计算机也不能直接识读和运行高级语言程序写的程序，必须借助于专门的系统软件——编译系统，将高级语言写的程序翻译成机器语言程序，然后，由计算机直接运行编译结果——机器语言程序。

高级语言又细分为两类，一类是面向过程的语言，使用这种语言设计程序，需要描述问题数据的特征和解题算法的步骤，如C语言就是属于面向过程的语言；另一类是面向问题的语言，使用这种语言设计程序，只需描述要解决的问题本身，交给计算机，由计算机自动产生解决问题的算法步骤。数据库管理系统的数据定义语言和数据操纵语言就属于面向问题的语言，如我们使用FOXBASE查询数据库时，只需要告诉数据库管理系统要查询的对象和查询条件，数据库管理系统就会自动完成指定的查询要求。当然，到目前为止，面向问题的语言还很不成熟，仅仅用在诸如数据库管理这样简单的狭窄领域中，通用的面向问题的程序设计语言现在还没有实用化。

尽管高级语言已经大大方便了人机对话，但是毕竟它还属于一种机器用语言，离人的自

然语言还有一段距离,人要想使用某种高级语言设计程序,还是需要专门学习这种语言,只不过高级语言比机器和汇编语言学起来要容易得多。从计算机诞生起,人们就开始追求完全采用人的自然语言来实现人机对话和设计程序,以控制计算机。我们可以设想,如果能够采用自然语言,比如我们的母语——汉语来设计程序,那么普通人完全不必花费时间、精力甚至金钱去专门学习计算机的语言。只可惜,这样良好的想法现在还无法在技术上变成现实,因为要实现上述想法,必须首先设计出如同汇编系统和编译系统的软件工具,以便能够把用自然语言写的程序翻译成机器语言程序,而这种工具在目前的技术条件下是无法实现的。

1.2 C语言特点

C语言是目前使用最广泛的一种高级语言,它不但可以用来设计应用软件,而且还可以用来设计系统软件。C语言之所以应用范围这样广,是因为它有下列一些独到的特点:

1) C语言既具有一般高级语言的特性,又具有汇编语言的特点,这样,使得它对开发系统软件特别合适。一般的高级语言不能对物理地址进行直接访问,也不允许对二进制位进行运算和处理,而C语言可以对物理地址进行直接访问,提供了位运算符,可以进行二进制位运算。由于上述特点,所以一些人认为C语言是介于高级语言和汇编语言之间的一种语言,而不是一个真正的高级语言。

2) C语言的运算符非常丰富。除了具有一般高级语言的运算符之外,C语言还有一些专有的运算符,如赋值运算符、条件运算符、逗号运算符、强制类型转换运算符等等,绝大多数运算和处理无须用语句来描述,用C语言的运算符就可以表达。

3) C语言简洁、紧凑。C语言将其他高级语言中不必要的(即去掉不会引起歧义)冗长表达都去掉了,这样,使得相同的操作和处理用C语言表达比用其他高级语言要简洁得多。如PASCAL语言的语句IF X>Y THEN Z:=X ELSE Z:=Y,用C语言的语句等价地表达为if (x>y) z=x; else z=y;C语言的IF语句省略了保留关键字THEN。

4) C语言使用非常灵活。相同的运算和处理,用其他高级语言往往只有一种表达方式,而用C语言可以有多种表达方式,如PASCAL语言语句IF X>Y THEN Z:=X ELSE Z:=Y的功能为求X和Y的最大值,并将该最大值赋予Z。要完成该功能在PASCAL语言中基本只有这样一种表达方式,而用C语言既可以设计为if (x>y) z=x; else z=y;又可以设计为z=x>y?x:y;再比如,要对X、Y、Z都赋值为5,用PASCAL语言只能写三个赋值语句:X:=5; Y:=5; Z:=5,而用C语言可以有許多不同的正确写法:x=y=z=5;或者 x=5,y=z=x; x=5;y=5;z=5等等。

5) C语言的数据结构丰富,具有现代高级语言所具有的各种数据结构,并且这些数据结构在C语言中具有更加丰富的特性,使得用它们可以更加准确地对现实世界进行模型化。如C语言的整数类型又细分为基本整数类型、长整数类型、短整数类型、无符号整数类型等等,C语言将整数类型和字符类型通用,使得程序设计更加方便、自由,C语言的指针类型用法非常灵活,借助于指针可以访问任何类型的内存变量。

6) C语言可移植性好。由于C语言并不过分地依赖于计算机的硬件环境,这样,一个C语言程序,可以不需改动或只进行很少的改动就可以在不同的机器上运行。目前,大、中、小、微型机上都可以使用C语言。

7) 有多种操作系统的支持。UNIX操作系统支持C语言,此外,现今广泛使用的其他操作系统也都支持C语言,因此,C语言不但有丰富的硬件环境支持,而且也有丰富的软件环境支持。

8) C语言是一种结构化的程序设计语言, 它提供丰富的结构化控制语句, 并且允许进行模块化的程序设计, 因此, C语言不但在开发系统软件和应用软件的工程中有广泛应用, 而且也特别适合作为教学语言。学生通过学习C语言, 既可以掌握这种功能强大的编程工具, 又可以在学习中积累结构化程序设计的经验, 养成良好的程序设计风格和习惯。

经过上面的分析和介绍, 我们应该能够初步理解, 为什么C语言从诞生开始一直长盛不衰, 为什么现在C语言能够这样风靡, 同时也应该明白我们要学习C语言程序设计的道理。

1.3 C语言程序结构简介

下面我们给出几个C语言程序的例子, 从下面这几个程序示例, 我们可以了解C语言的概貌、C语言程序的一般结构和特点。

例1.1

```
#include <stdio.h>
main()
{
    printf("Hello,world\n");
}
```

该程序完成的功能是在显示器上显示句子Hello,world。程序的第一行#include <stdio.h>是一个C语言的预编译命令, 表示在编译时要将系统文件Stdio.h嵌入当前源程序文件的前面, 构成一个文件一起编译。其中, 系统文件Stdio.h中存储有系统提供的所有标准输入输出函数的头部说明, 只有将它嵌入我们的源程序文件, 我们才可以在自己的源程序中调用这些标准的输入输出函数, 因此Stdio.h又称为标准输入输出头文件。编译系统一般提供有若干个头文件, 每个头文件都有一类系统库函数的头部说明, 我们要想在程序里调用哪类库函数, 都需要将该类库函数的头文件嵌入我们的源程序文件。从程序的第二行到结尾是一个函数的定义; 第二行main()是被定义函数的头部, 包括函数名为main, 表示当前函数为主函数, ()表示形式参数表, 括号里面什么也没有, 表示该函数没有形式参数; 从第三行的“{”到最后的“}”是函数体, 其中应该列出函数的数据类型定义和语句描述, 本例中主函数体只有一个语句printf(“Hello,world \n”);它是C语言的函数调用语句, 调用C语言的标准输出函数printf, 完成输出工作。

一个C语言程序是由一个或多个函数组成的, 其中必须有而且只能有一个函数名为main, 即名为main的函数是一个程序中的特殊函数, 称为主函数, 它是程序执行的入口, 程序的执行总是从主函数的第一个语句开始的, 由主函数去调用其他函数, 主函数的语句执行完意味着整个程序执行完毕。例1.1就是只由主函数构成的程序, 程序的执行从它的第一个语句printf(“Hello,world \n”);开始, 执行完该语句后, 程序即执行结束(因为主函数只由该语句本身构成)。

例1.2

```
#include <stdio.h>
main()
{ /* 求两个整数的最大值 */
    int a,b,max;

    scanf("%d,%d",&a,&b);

    /* 定义三个整数类型的变量a、b、max, 其中a和b
       用来保存输入的两个整数, max用来保存求出来的最大值 */
    /* 输入两个整数 */
```

```

if (x>y) max=x; else max=y;      /* 求最大值 */
printf("MAX=%d\n", max);      /* 输出最大值 */
}

```

例1.2程序也是只由主函数组成的，程序的功能是求输入的两个整数中的最大值。程序开始定义了三个整数类型(用关键字int来表示)的变量a、b、max，其中a和b用来保存输入的两个整数，max用来保存求出来的最大值，变量定义下面紧接着是描述求两个整数最大值算法的语句：先是通过调用系统输入库函数scanf输入两个整数，分别存储在变量a和b中，然后用if语句判断a和b的大小关系，将其中的大者赋予max，这样，max中保存的肯定是两个整数的最大值，最后一个语句是通过调用系统输出库函数，来输出max的值，即输出求出来的最大值。程序中由/*和*/括起来的部分是程序的注释。注释不是程序的有机组成部分，它只是嵌在源程序中间，起到增加程序可读性的作用。编译系统编译源程序时，会略去其中注释不编译，所以，在形成的目标程序(即编译结果—机器语言程序)中是不存在注释的，只有程序中的语句和变量经过编译后形成的机器指令。因此，程序设计人员可根据自己的习惯和爱好，随意在程序的任何位置加注释或去掉注释，只要不改动程序的语句和变量定义等有机组成部分，程序功能和运行结果是不会受到任何影响的。

程序运行后，首先等待用户从键盘输入数据，若用户输入

4,5 <CR>

则程序输出结果为

MAX=5

例1.3

```

#include <stdio.h>
float min(x,y)
float x,y;
{
    float z;
    if (x<y) z=x; else z=y;
    return(z);
}
main()
{ float a,b,c;
  scanf("%f,%f,%f", &a,&b,&c);
  printf("MIN=%f", min(min(a,b),c));
}

```

例1.3的程序是由两个函数组成的：主函数和用户自定义函数min，由主函数去调用函数min。函数min的功能为求任意两个实数的最小值，当主函数需要求某两个数的最小值时，只需要以这两个数为参数调用min即可。主函数开始定义三个实数类型的变量a、b、c，然后通过调用输入库函数scanf输入三个实数，分别赋予a、b、c，最后通过调用printf输出函数min(min(a,b),c)的值。函数min(min(a,b),c)以min(a,b)和c为参数调用用户自定义函数min，得到min(a,b)和c的最小值，而min(a,b)是以a和b为参数对min的调用，得到的是a和b的最小值，因此函数min(min(a,b),c)的值为a和b的最小值与c中的最小值，即a、b、c三个实数的最小值，所以整个程序完成的功能为求输入的三个实数中的最小值。

程序运行后，首先等待用户从键盘输入数据，若用户输入

4.1,5.5,-7.9 <CR>

则程序输出结果为

MIN=-7.900000

通过上面三个程序示例，可以看出一个C语言程序具有如下结构特点：

- 1) 一个C语言程序是由一个或多个函数构成的，其中只能有而且必须有一个函数名为main，它是程序的主函数。一个程序总是从它的主函数开始执行的。
- 2) 每个函数是由头部和函数体两部分组成的，函数头部说明返回值的类型、函数名和形式参数，函数体由左大括号“{”开始，右大括号“}”结束，中间是函数体的内容。函数体的内容分为两部分：对数据对象描述的变量定义和对算法描述的语句序列。
- 3) 可以在程序中的任何位置为程序加上注释，注释必须以“/*”开始，“*/”结束，中间是注释的内容。程序中的注释只是用来提高程序的可读性，它们并不代表任何计算机的运算和处理。
- 4) C语言的书写格式非常自由，一个语句既可以在一行内书写，也可以分成几行来写；一行可以只写一个语句，也可以写多个语句。
- 5) 每个C语言的语句是以分号“；”结尾的，分号是C语言语句的组成部分，而不像某些语言那样分号作为语句与语句之间的分隔符。

1.4 计算机运行C程序的过程

我们前面曾经介绍，用高级语言书写的程序，计算机是不能直接运行的，必须借助于专门的系统软件——编译系统，由它将高级语言书写的程序翻译成机器语言程序，然后让计算机运行机器语言程序。由于编译系统在翻译时保证了翻译的结果——机器语言程序与被翻译的对象——输入的高级语言程序二者逻辑上严格等价，即功能完全相同，所以运行编译产生的机器语言程序相当于运行了被编译的高级语言程序。

有些编译系统只是将高级语言程序翻译成汇编语言程序(现代编译系统一般都不采用这种做法)，在这种情况下，为了产生机器语言程序，还需要借助于汇编系统，进一步地将编译产生的汇编语言程序再翻译成机器语言程序。还有些编译系统，它翻译产生的是一种可重新定位的指令代码，在这种情况下，还需要借助于连接程序进行连接，将各个模块连接在一起，确定程序变量或常量在内存中的地址，对指令地址进行代真，以形成可直接独立运行的绝对指令代码程序。

根据前面介绍的这些原理，可以看出，将一个用C高级语言书写的程序在计算机上运行的过程包括以下几个步骤(假定编译系统直接将源程序编译成机器代码程序)：

- 1) 编辑 使用文本编辑软件，如EDLIN、Wordstar、Edit、WPS、Windows的记事本等，输入C语言源程序，构成源程序文件。在输入程序的过程中，可以借助于所用编辑软件提供的编辑功能，对源程序进行编辑修改。
- 2) 编译 使用C语言的编译系统，对已经编辑好的源程序文件进行编译。编译系统在编译的过程中，要对源程序进行语法语义检查，并在发现一个或一组错误时，停止继续编译，向用户报告编译错误信息，指出出错位置和错误性质(即是什么错误)。这时程序设计人员就可以在编译系统出错信息的提示下，重新启动编辑系统，对源程序中的出错位置进行编辑修改，然后再启动编译系统，对修改后的源程序进行编译。当然再次编译可能还有错误，因此“编

译→编辑修改→编译”过程可能要重复多次,直到编译系统成功地将源程序编译成机器代码程序为止。

3) 连接 使用连接软件,将属于一个程序而分别编译的各个源程序文件的机器代码模块(可重新定位的机器代码)连接组装起来,形成可直接运行的绝对指令代码程序。在DOS和WINDOWS环境下,绝对指令代码程序文件的扩展名一般为“EXE”。

4) 运行 编译连接成功后,形成了可直接独立运行的绝对指令代码程序,这样的程序机器就可以执行了。如在DOS下,可以把编译连接形成的“EXE”文件的根名直接输入到系统提示符下,然后按回车,机器就开始运行这个“EXE”文件里的程序,所以我们习惯上说“EXE”文件是一种可执行文件。

经过上面的分析,我们知道,要想运行C语言书写的程序,首先要有编辑、C语言编译和连接等软件。现在广泛使用的这种软件是Turbo C,它是一个C语言的集成化开发软件。所谓集成化是指该软件将编辑、编译、连接等多项功能集于一身。这样,用户借助于一个Turbo C软件就可以完成C语言源程序的录入编辑、编译、连接、调试、运行等功能,无须在运行一个C语言程序时反复多次启动和运行不同的软件,这一方面提高了工作效率,另一方面也给用户带来了方便。

要想使用Turbo C,首先我们要拥有它。Turbo C系统是由一组文件组成的。一般情况下,将它们存储在硬盘或软盘的一个子目录下,如Turbo或TP(当然子目录名也可能不是Turbo和TP,因为子目录名用户是可以随便修改的)。在Turbo C系统的所有文件中,有一个文件是主文件,其名往往为Turbo.EXE或TP.EXE(当然文件名也可能不是它们,因为用户对该文件名修改对系统运行也是没有影响的)。下面我们介绍使用Turbo C的大致方法。

假设在C盘TP子目录下存储有Turbo C的所有文件,其中主要文件名为Turbo.EXE,则启动Turbo C的步骤为:

- 1) 在DOS系统提示符下键入CD命令,将当前工作目录转到C盘的TP子目录。
- 2) 键入Turbo,按回车,就完成了Turbo C的启动。

Turbo C启动成功的标志是屏幕上显示如图1-1的画面。

File	Edit	Run	Compile	Project	Options	Debug	Break / watch
Line 1	Col 1	Insert Indent Tab Fill Unindent				C:NONAME.C	
源程序输入区							
F1-Help	F5-Zoom	F6-Switch	F7-Trace	F8-Step	F9-Make	F10-Menu	

图 1-1

屏幕第一行为菜单行,上面列出Turbo C提供的所有菜单名,每个菜单下列出的是一组命令,如“File”菜单下包括Turbo C提供的所有有关文件存取的命令(“SAVE”——将源程序文件存盘,“OPEN”——打开磁盘上的一个源程序文件,等等),“Run”下包括运行程序的有关命令,“Compile”下包括编译程序的有关命令,等等。

屏幕第二行是状态行,这里列出一系列编辑状态,如“Line”后的数字显示当前光标处在源程序输入区的第几行,“Col”后的数字显示当前光标处在源程序输入区的第几列,“Insert”表示当前使用的是插入方式进行编辑,等等。

屏幕中间是源程序输入区,用户从键盘上输入的内容将显示在这里。输入源程序内容的

过程中, 用户可以随时使用Turbo C提供的编辑命令, 对已经录入的内容进行添加和修改。Turbo C提供的编辑命令与Wordstar和WPS是类似的, 如CTRL+Y是删除一行, CTRL+K+B是标记文本块首, 等等。

屏幕最后一行显示的是功能键功能的介绍, 根据这些提示, 我们知道: 按F1键可随时获取Turbo C的联机帮助, 按F8键可以对源程序进行单步跟踪运行, 按F9键可以对源程序进行编译, 按F10键可以使光标上到菜单行, 以便选取需要的菜单命令执行, 完成所希望的功能。

启动Turbo C后输入源程序并运行它的常用操作步骤如图1-2所示。

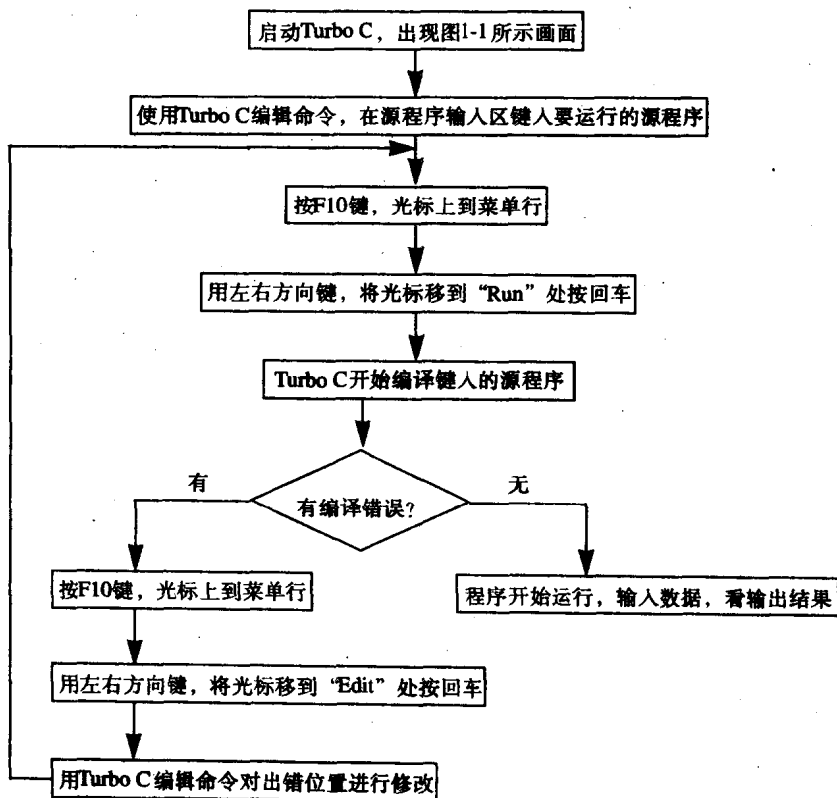


图 1-2

1.5 学习《C语言程序设计》的要点

任何一个语言(包括我们学习的汉语和英语以及程序设计语言)都是由语法和语义两方面定义的一个记号系统。语法又包括词法和句法, 它从形式上描述了一个语言的各类成分如何由更低级的成分构成, 如英语的词法定义了如何由英语字母构成合法的单词, 句法刻画了如何由“单词”构成合法的更高级的单位——短语和句子。而语义则定义了各种语法结构所代表的特定意义。C语言也是由词法、句法和语义三方面来定义的, 词法规定了C语言合法“单词”——关键字、运算符、分界符、标识符、常数等等的构成规则, 句法规定了C语言合法“句子”——表达式、语句等等——是如何由单词构成的, 语义则规定了C语言每种形式的表达式、语句等等所代表的含义, 或者说, 这些形式所代表的计算机内部的操作动作。

因此, 要想学会C语言, 首先我们应该知道C语言的词法, 即知道C语言各类“单词”的

正确写法。C语言的关键字、运算符、分界符是固定的、有限的，我们只需要记住它们的形式及其含义即可，而标识符和各种类型常数是不固定的、无限的，C语言只是规定了它们的构成规则，所以我们应该知道这些规则，并会利用这些规则正确构造标识符和各种类型常数。掌握词法后，我们应该学习C语言的句法，即知道C语言在把单词构成表达式和语句进而整个程序方面的规定，会根据这些规定正确地构造表达式、语句直至整个程序。除了词法、句法外，我们还应该学习C语言的语义，即知道每种格式的表达式、语句直至整个程序所代表的含义，或者说，这些形式在计算机内部的处理和执行过程。

掌握了C语言的词法、句法和语义只能说我们懂得了C语言，但是我们学习任何语言的最终目的总是要能够用这种语言进行交流，对于程序设计语言来说，就是要能够用这种语言书写程序(相当于用这种语言表达自己的思想)和阅读别人用这种语言书写的程序(相当于理解别人用这种语言说的话或写出来的句子和文章)。因此，我们学习C语言的最终目的是学会用C语言正确地写程序以解决实际的问题，以及阅读自己或别人为解决某问题而用C语言写的程序，即用C语言写程序和阅读C程序，前面学习C语言的词法、句法和语义都是为这最终目的服务的。当然，要达到熟练用C语言写程序和阅读C程序，单是掌握C语言的词法、句法和语义还不够，还需要大量的程序设计和阅读程序实践，逐渐积累经验和技巧。