



普通高等教育“十二五”规划教材

C语言程序设计

王立武◎主 编



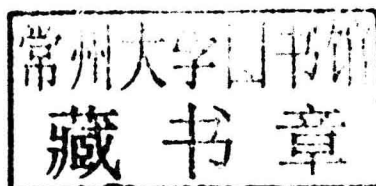
科学出版社

普通高等教育“十二五”规划教材

C 语言程序设计

王立武 主编

张春芳 刘莹昕 副主编
陈 艳 张 宇



科学出版社

北 京

内 容 简 介

C 语言是一种结构化、模块化、可编译的主流程序设计语言。本书根据 C 语言的内容结构特点,按章提供重要的基础知识点和详细讲解,并提供一定量习题覆盖程序设计的常用算法和基本的编程技巧。

本书在编排上分层次、分程度,由浅入深、循序渐进、难易兼顾,具有可读性和逻辑性强的特点。读者可以通过本书的学习,逐步熟悉并掌握 C 语言的基础知识、编程方法和技巧,以拓宽程序设计的思路。本书涉及的所有程序都在 Microsoft Visual C++ 6.0 集成开发环境中进行了调试及运行,使用的算法全部采用 N-S 结构化流程图表示。

本书实用性强,兼顾了各级各类考试的要求,既可以作为高等院校学生学习 C 语言程序设计的教材,也可以作为 C 语言程序设计初学者的学习用书,还可以作为各类考试的用书。

图书在版编目(CIP)数据

C 语言程序设计 / 王立武主编. —北京: 科学出版社, 2011

(普通高等教育“十二五”规划教材)

ISBN 978-7-03-032784-0

I. ①C… II. ①王… III. ①C 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 233749 号

责任编辑: 宋 丽 郭丽娜 / 责任校对: 王万红

责任印制: 吕春珉 / 封面设计: 东方人华平面设计部

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

双 青 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

*

2011 年 12 月第 一 版 开本: 787×1092 1/16

2011 年 12 月第一次印刷 印张: 14 3/4

字数: 337 000

定价: 29.00 元

(如有印装质量问题, 我社负责调换〈双青〉)

销售部电话 010-62142126 编辑部电话 010-62134021

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

C 语言是一种结构化程序设计语言，兼有高级语言的特点和低级语言的功能，既可以用来编写系统软件，如 UNIX 等，也可以用来编写应用软件，如管理系统软件等。C 语言不仅功能强大，而且具有运算丰富、表达能力强、程序执行效率高、较好的可读性和可移植性等特点，是一门较为通用的程序设计语言。

目前，国内外许多高等院校都开设了 C 语言程序设计课程，并将其作为计算机基础教育的入门课程，要求计算机专业和大多数非计算机专业学生必须掌握。C 语言课程教学的任务一方面要求学生掌握 C 语言的语法规则，学会用 C 语言设计基本算法，掌握程序设计的基本规律，在实践中逐步掌握程序设计的思想和方法，培养解决实际问题的能力；另一方面能够使学生举一反三，为学习其他高级程序设计语言打下良好的基础。为此，编者结合多年来的教学实践和经验编写了这本书。

本书是为学习 C 语言的初学者编写的，采用 Microsoft Visual C++ 6.0 作为教学环境，使用的算法全部采用 N-S 结构化流程图表示。本书详细地介绍了 C 程序设计的基本概念、结构化程序设计的基本概念、基本技术和方法，在编写过程中力求体系结构简单明了，基本内容的介绍由浅入深、循序渐进，重点、难点内容的分析深入透彻，易于理解、掌握。同时，本书精选了大量的案例程序，尽可能将知识点与案例相结合，所有例题及习题的参考答案都在 Microsoft Visual C++ 6.0 集成开发环境中通过调试。学生可以通过阅读例题理解相应的语法规则，通过上机和编程实践逐步领会和掌握 C 语言的实际应用，掌握程序设计的基本思想和方法，进而达到灵活运用之目的。

本书由王立武担任主编，张春芳、刘莹昕、陈艳、张宇担任副主编，由张宇担任主审。第 1 章由陈艳编写，第 2 章、第 9 章由张春芳编写，第 5~8 章由刘莹昕编写，第 3 章、第 4 章、第 10 章由王立武编写，附录部分由张宇编写。全书由王立武、陈艳统稿。

由于编者水平有限，书中难免存在缺点和不足，恳请有关专家和读者批评指正。

目 录

第 1 章 C 语言概述及 C 程序的实现	1
1.1 计算机及计算机编程语言	1
1.1.1 机器语言	1
1.1.2 汇编语言	2
1.1.3 高级语言	2
1.1.4 高级语言及软件的发展	3
1.2 C 语言概述	3
1.2.1 C 语言的特点	4
1.2.2 C 语言标识符与关键字	4
1.2.3 C 程序的结构	5
1.3 C 程序举例	7
1.4 运行 C 程序的过程	8
1.5 C 语言程序上机运行步骤	9
习题 1	11
第 2 章 数据类型、运算符与表达式	12
2.1 C 语言的数据类型	12
2.2 常量与变量	13
2.2.1 常量和符号常量	13
2.2.2 变量	14
2.3 整型数据	15
2.3.1 整型常量	15
2.3.2 整型变量	16
2.4 实型数据	16
2.4.1 实型常量	16
2.4.2 实型变量	16
2.5 字符型数据	17
2.5.1 字符型常量	17
2.5.2 字符型变量	18
2.5.3 字符串常量	19
2.6 基本输入/输出函数	19
2.6.1 格式输出函数 printf()	20
2.6.2 格式输入函数 scanf()	23

2.6.3 字符输出函数 putchar()	26
2.6.4 字符输入函数 getchar()	26
2.7 运算符和表达式	27
2.7.1 算术运算符和算术表达式	29
2.7.2 赋值运算符和赋值表达式	31
2.7.3 不同类型数据间的混合运算	33
2.7.4 关系运算符和关系表达式	35
2.7.5 逻辑运算符和逻辑表达式	36
2.7.6 条件运算符和条件表达式	37
2.7.7 逗号运算符和逗号表达式	37
2.7.8 求字节数运算符	38
2.7.9 位运算符和位运算	38
习题 2	40
第 3 章 简单程序设计	42
3.1 程序设计初步	42
3.1.1 利用计算机求解实际问题的步骤	42
3.1.2 算法及表示方法	45
3.1.3 C 语言的语句	48
3.2 顺序结构程序设计举例	49
习题 3	52
第 4 章 选择结构程序设计	54
4.1 if 语句	54
4.1.1 双分支结构	54
4.1.2 双分支结构的特例：单分支结构	56
4.1.3 if 语句的嵌套结构	58
4.1.4 使用条件表达式实现简单的选择语句	61
4.2 switch 语句	62
4.3 选择结构程序设计举例	64
习题 4	66
第 5 章 循环结构程序设计	69
5.1 while 循环	69
5.2 do-while 循环	72
5.3 for 循环	74
5.4 break 语句与 continue 语句	77
5.4.1 break 语句	77
5.4.2 continue 语句	78

5.5 goto 语句	79
5.6 循环的嵌套	80
5.7 循环结构程序设计举例	82
习题 5	87
第 6 章 数组	89
6.1 一维数组	89
6.1.1 一维数组的定义	89
6.1.2 一维数组的引用	90
6.1.3 一维数组的初始化	91
6.1.4 一维数组程序举例	92
6.2 二维数组	96
6.2.1 二维数组的定义	97
6.2.2 二维数组的引用	97
6.2.3 二维数组的初始化	98
6.2.4 二维数组的程序举例	98
6.3 字符数组	102
6.3.1 字符数组的定义	102
6.3.2 字符数组的初始化	102
6.3.3 字符数组的引用	103
6.3.4 字符串的处理函数	104
习题 6	108
第 7 章 函数与编译预处理	110
7.1 函数的概述	110
7.1.1 函数的分类	111
7.1.2 函数的定义	111
7.2 函数的调用	112
7.2.1 函数调用的一般形式	113
7.2.2 函数的调用方式	113
7.2.3 对被调函数的说明	113
7.2.4 函数的返回值	114
7.2.5 函数的参数传递方式	116
7.3 函数的嵌套调用和递归调用	119
7.3.1 函数的嵌套调用	119
7.3.2 函数的递归调用	120
7.4 变量的作用域和存储类别	123
7.4.1 变量的作用域	123
7.4.2 变量的存储类别	125

7.5 内部函数和外部函数	129
7.5.1 内部函数	129
7.5.2 外部函数	129
7.6 编译预处理	130
7.6.1 宏定义	130
7.6.2 不带参数宏定义	130
7.6.3 带参数宏定义	131
7.6.4 文件包含	134
7.6.5 条件编译	135
习题 7	136
第 8 章 指针	139
8.1 变量的地址和指针	139
8.2 指向变量的指针变量	140
8.2.1 指针变量的定义	140
8.2.2 指针运算符	140
8.2.3 指针变量的赋值	141
8.2.4 指针变量的引用	141
8.2.5 指针变量作为函数参数	142
8.3 数组和指针	144
8.3.1 一维数组和指针	144
8.3.2 二维数组和指针	147
8.3.3 指针数组	150
8.4 字符串和指针	151
8.5 指向指针的指针	154
8.6 函数的指针和指向函数的指针变量	155
8.7 返回指针的函数	155
习题 8	156
第 9 章 结构体与共用体	158
9.1 结构体类型与结构体类型变量	158
9.1.1 结构体类型的定义	158
9.1.2 结构体类型变量的定义和初始化	159
9.1.3 结构体变量的引用	161
9.2 结构体数组	164
9.3 指向结构体类型数据的指针	166
9.4 单向链表	169
9.4.1 单向链表的数据结构	169
9.4.2 动态分配和释放空间函数	170

9.4.3 单向链表的基本操作	170
9.5 共用体	182
9.6 枚举类型	185
9.7 用 typedef 定义类型	187
习题 9	188
第 10 章 文件	189
10.1 C 文件概述	189
10.1.1 文件系统概述	189
10.1.2 文件类型指针	190
10.2 文件的打开与关闭	190
10.2.1 文件的打开	191
10.2.2 文件的关闭	192
10.3 常用的文件读写函数	193
10.3.1 字符读写函数	193
10.3.2 字符串读写函数	195
10.3.3 格式化读写函数	196
10.3.4 数据块读写函数	199
10.4 文件的定位函数	200
10.5 出错检验	201
习题 10	201
参考文献	204
附录	205
附录 1 常用字符与 ASCII 码对照表	205
附录 2 C 常用库函数	206
附录 3 习题参考答案	211

第 1 章 C 语言概述及 C 程序的实现

本章学习目标

- 了解计算机程序、编程语言的概念；
- 了解 C 语言的特点；
- 了解 C 语言程序的基本结构；
- 掌握运行一个 C 语言程序的步骤。

1.1 计算机及计算机编程语言

计算机是一种能够按照事先存储的程序（指令的集合），能自动、高速地进行大量数值计算和信息处理的电子设备。构成计算机系统的各种设备如键盘、显示器、鼠标器、磁盘、内部存储器（简称内存）、中央处理器、打印机等称为硬件；在计算机上运行的计算机程序（computer program）称为软件。

计算机程序是人们为了解决某种问题，用计算机可以识别的代码编写的一系列数据处理步骤，就是控制计算机完成特定功能的一组有序指令的集合。编写计算机程序所使用的语言称为程序设计语言，它是程序员与计算机进行交流的工具。在程序中，一般要表达数据（包括定义用于存储数据的变量），并描述数据处理的过程。

程序设计语言有低级语言（机器语言、汇编语言）和高级语言之分，其区别主要是它们对计算机硬件的接近程度不同。

1.1.1 机器语言

计算机只能接受和识别 0 和 1 这样的二进制信息。各种类型的计算机都分别规定了由若干个二进制位的信息（即若干个 0 或 1 组成的信息）组成一个指令。这种由二进制代码形成的指令称为机器指令，它是计算机能直接识别和执行的唯一的一种指令形式。

例如，10000000、10010000、01110000 就是 3 条机器指令。其中，10000000 表示“加法”操作；10010000 表示“减法”操作；01110000 表示“传送”操作。

一条机器指令产生一个相应的机器操作。每一种计算机都有若干种指令（如加法指令、减法指令、传送指令等），以实现各种不同的操作。一种计算机的指令的集合称为该计算机的机器语言，或者称为该计算机的指令系统。

为了解决某一个实际问题，可以从机器语言中选择所需的指令组成一个指令序列，这个指令序列即称为机器语言程序。

机器语言依赖于具体计算机，不是各类计算机都通用的，它是面向机器的。

计算机能直接识别和执行用机器语言描述的算法，执行效率高，但要直接用 0 和 1

这样的二进制代码编写程序，不但难学、难记、难写、难检查、难调试、难推广，而且不适合一般用户学习。

1.1.2 汇编语言

由于机器指令繁琐难记，人们想到可以用一些特定的符号来代替某些操作。例如，用 ADD 代表“加”，用 SUB 代表“减”，用 LD 代表“传送”。

```
ADD  A, B;    /*表示 A+B→A*/  
SUB  A, B;    /*表示 A-B→A*/  
LD   A, B;    /*表示 B→A*/
```

显然，这样书写的指令比机器指令易学易记。其中，ADD、SUB、LD 就是助记符，使用助记符来代替二进制指令的语言称为符号语言，又称为汇编语言。

但是，计算机并不能直接识别和执行汇编语言的指令，需要先将这些汇编指令转换成二进制的机器指令，计算机才能执行。一般地，一条汇编语言指令对应翻译成一条机器指令。这种翻译转换工作不是人工进行的，而是由一个称为汇编程序的软件去完成的，转换的过程称为汇编。用汇编语言书写的程序称为源程序，汇编以后的程序称为目标程序。

汇编语言也是随不同计算机而异的。各种类型的计算机分别有自己的汇编语言。用汇编语言编写程序，也需要了解计算机内部的结构，对使用者的要求较高，因此也不易推广普及。汇编语言程序和机器语言程序一样，执行速度快，占用存储空间小。

1.1.3 高级语言

由于机器语言和汇编语言是面向机器的，可移植性差，而且难学难记，所以人们希望能够找到一种语言，它的表示方法接近于人类的自然语言，易懂易学，而且在各种计算机上都能通用。这种语言不再是面向机器的，而是面向过程的，即不必考虑机器内部的构造和不同机器的特点，只要按照解题的算法写出解题过程，计算机就能执行，这种语言称为算法语言，又称为高级语言。

为了从根本上摆脱语言对机器的依赖，人们经过多年的潜心研究，在 1956 年推出了第一种与具体机器指令系统无关，并且表达方式接近自然语言的计算机语言——FORTRAN 语言。在 FORTRAN 语言程序中，采用了具有一定含义的数据命名方法和人们描述的具体事物。所以，人们把这种与具体机器指令系统无关，表达方式接近人类自然语言的计算机语言称为高级语言。高级语言的出现是编程语言发展史上的一大进步，它缩小了编程语言与自然语言之间的距离。

几十年来，出现了很多种高级语言，其中影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、PL/1、PASCAL、C、PROLOG、C++、VC++、VB、Delphi、Java 等。

计算机不能直接执行高级语言程序，需要将其翻译成机器指令。由于高级语言的一条指令对应多条机器指令，因此这个翻译工作比汇编语言的翻译过程复杂。将高级语言转换成目标程序是由编译程序来完成的。

不是在任何一台计算机上都能使用任意一种高级语言，只有配备了某一种高级语言的编译程序，计算机才能运行这种高级语言的程序。例如，想要运行一个C语言的程序，必须保证所使用的计算机已经配置了C语言的编译程序。

1.1.4 高级语言及软件的发展

计算机软件是用程序设计语言编写出来的，多数软件是用高级语言编写的。高级语言的发展经历了从早期语言到结构化程序设计语言，从面向过程语言到非过程化程序语言的过程。相应地，软件的开发也由最初的个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

20世纪60年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是各自为战，缺乏科学规范的系统规划与测试、评估标准，其后果是大量耗费巨资建立起来的软件系统，由于含有错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，以至于几乎没有不出错的软件。这一切极大地震动了计算机界，史称软件危机。人们认识到，大型程序的编制不同于写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程；程序的设计应易于保证其正确性，也便于验证其正确性。在1969年，人们提出了结构化程序设计的方法。在1970年，第一个结构化程序设计语言Pascal语言的出现，标志着结构化程序设计时期的开始。

从20世纪80年代初开始，在软件设计思想上，软件领域又产生了一次革命，其成果就是面向对象的程序设计。在此之前的高级语言，几乎都是面向过程的，程序的执行是流水线式的，即在一个模块被执行完成前，人们不能干别的事，也无法动态地改变程序的执行方向。这和人们日常处理事物的方式是不一致的，对于用户而言，用户是希望发生一件事就处理一件事，即不能面向过程，而应该是面向具体的应用功能，也就是对象(object)。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装紧密的功能模块，称之为软件集成块，它与具体应用无关，但能够相互组合，完成具体的应用功能，同时又能重复使用。对于用户来说，用户只需关心它的接口（输入量、输出量）及能实现的功能，但至于如何实现的则完全不用关心，C++、VB、Delphi就是典型代表。

高级语言的下一个发展目标是面向应用，即用户只需要告诉程序自己要干什么，程序就能自动生成算法，并自动进行处理，这就是非过程化程序语言。

1.2 C语言概述

C语言是由贝尔实验室的B.W.Kernighan和D.M.Ritchie于1972年提出的，由于C语言具有很多突出的优点，目前成为计算机程序设计语言的主流语言，它是学习和掌握更高层的开发工具C++和VC++的基础。

1.2.1 C 语言的特点

1. C 语言简洁、紧凑

C 语言一共只有 32 个关键字, 9 种控制语句。程序书写形式自由, 主要用小写字母表示, 所以 C 语言入门比较容易。

2. C 语言是一种结构化的程序设计语言

C 语言具有结构化的控制语句 (选择结构语句、循环结构语句), 并用函数作为程序模块以实现程序的结构化。

3. C 语言具有丰富的数据类型

C 语言的数据类型包括整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型、空类型, 能实现各种复杂的数据结构的运算。

4. C 语言提供了丰富的运算符

C 语言提供了多种运算符, 许多操作都可以用运算符和运算对象组成的表达式表示。

5. C 语言可以直接对硬件进行操作

C 语言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, 可以用来编写系统软件。

6. C 语言移植性好

C 语言程序基本上不做修改就能用于各种型号的计算机和各种操作系统, 程序兼容性好。

1.2.2 C 语言标识符与关键字

1. C 语言的字符集

C 语言中可以使用的字符有以下 3 类。

- 1) 英文字母: A~Z 和 a~z。
- 2) 数字: 0~9。
- 3) 特殊字符: +、-、*、/、%、=、_、()、&、#、!、|、< >、.、,、;、:、"、'、?、{ }、~、\、[]、^、空格符、换行符、制表符。

2. 标识符

标识符就是名称, 是程序中用来标识变量、符号常量、函数、数组、类型等程序对象的有效字符序列。在 C 语言程序设计中, 组成标识符的字符序列要遵循如下规则。

1) 标识符只能由英文字母、数字和下划线 3 种字符组成, 并且英文的大写字母和小写字母被认为是不同的字符。例如, “Abc”、“abc” 和 “ABC” 是 3 个不同的标识符。

2) 第一个字符必须是字母或下划线, 不能是数字。例如, “_add12”、“_MAX” 是合法的标识符, 而 “3_Year” 是非法的标识符。

3) 不能使用系统的关键字(保留字)作为用户标识符。

注意: 对于标识符的字符序列的长度, 不同的 C 编译系统有不同的规定。

【例 1-1】 下列字符序列中, 哪些是合法的 C 语言标识符?

a@bc s05 file.c A6C 10page printf _red 北京 中国

合法的标识符有 s05、A6C、printf、_red。

3. 关键字(或保留字)

在 C 语言中, 将系统用到的具有特定含义的标识符称为关键字(或保留字)。这些关键字共有 32 个, 都用小写字母书写, 不允许用户作为自定义标识符使用。按功能将关键字分为如下几类。

(1) 程序控制语句关键字

if else for do while continue
switch break case default goto return

(2) 数据类型关键字

int char float double long short
signed unsigned enum struct union void

(3) 存储类型关键字

auto register static extern

(4) 常量、变量定义和自定义类型关键字

const volatile typedef

(5) 字节长度测试关键字

sizeof

1.2.3 C 程序的结构

1. 两个简单的 C 程序介绍

【例 1-2】 编写程序, 实现从键盘上输入长方形的任意长和宽, 在屏幕上输出长方形的面积的功能。

```
#include<stdio.h>
main()                               /* 主函数 */
{
    float x,y,s;                     /* 定义 3 个变量, 分别代表长、宽、面积 */
    scanf("%f%f",&x,&y);             /* 从键盘上输入长和宽的值 */
    s=x*y;                           /* 计算面积 */
    printf("%f\n",s);                /* 在屏幕上输出面积 */
}
```

当程序运行后,在屏幕上出现闪烁的光标,要求用户从键盘上输入变量 x 和 y 的值,如输入如下数据:

5□4<回车>

其中,5 和 4 分别赋值给变量 x 和 y ; 它们之间的空格(□)作为这两个数据的分隔符。屏幕上显示程序的运行结果:

20.000000

【例 1-3】 定义一个函数,求圆的面积。

```
#include<stdio.h>
#define PI 3.1416          /* 定义 PI 为符号常量,表示圆周率 */

/* 定义一个名称为 area 的函数,其功能是求半径为 x 的圆的面积 */
double area(double x)      /* 函数和参数的类型都是 double */
{ double y;
  y=PI*x*x;
  return y;                /* 返回 y 的值作为圆的面积,返回到调用函数 area 处 */
}

main()                    /* 主函数 */
{ double r,s;
  scanf("%lf",&r);
  s=area(r);              /* 调用函数 area,r 作为参数,求圆的面积 */
  printf("s=%lf \n",s);
}
```

2. C 程序的主要结构特点

1) C 程序是由一个或多个函数组成,函数的内容称为函数体,由一对大括号({})括起来。

2) 主函数 main()是由系统提供的,一个程序**有且仅有一个**主函数 main()。

3) C 程序总是从主函数 main()开始执行,主函数 main()可以放在程序的任何位置。

4) 程序中的#include<stdio.h>和#define PI 3.1416 是编译预处理命令,在源程序被编译之前先由预处理程序进行处理。

#include<stdio.h>是通知预处理程序把标准输入/输出头文件<stdio.h>中的内容包含到程序中,<stdio.h>中包含了编译程序在编译标准输入/输出库函数时要用到的信息。

#define PI 3.1416 的作用是定义一个常量标识符 PI,在程序中用它代替 3.1416,这样做的目的是增加程序的可读性和可移植性。

5) 在 C 语言程序中可以使用注释,不影响程序的编译和运行,程序编译时会忽略这些内容,注释的作用是给程序员阅读程序提供方便,在程序中适当地加上注释是程序员必须养成的良好习惯。

注释语句格式一:

```
/* 注释内容 */
```


在C99标准中,提供了另外一种注释语句格式,注释语句格式二:

// 注释内容

注意:如果使用格式一,注释的内容可以写在若干行上;如果使用格式二,注释的内容只能写在一行上。

C程序由函数组成,函数也称为模块;每个模块实现一个单一的功能,称为模块化程序设计。模块化程序设计的思想是:按照自顶向下的原则,把问题逐层分解。先从总体出发,把问题分成若干个模块,每一个模块代表一个任务。每个任务都能用基本结构(顺序结构、选择结构和循环结构)表示。

3. C程序书写的基本要点

C语言严格区分大小写英文字母,但习惯上通常使用小写英文字母书写语句代码,用大写字母作为常量的标识符和其他特殊用途。

1) C语言程序是由语句组成,每个语句都具有一定的功能。

2) C语言程序使用分号(;)作为语句的终止符。一般情况下,每个语句占用一个书写行的位置,但也可以在一行中写多个语句,或一个语句也可以占用多行。一条语句要换行,只要在空格处换行即可。

3) 在C语言程序中,一个完整的程序模块要用一对大括号({})括起来,表示程序的结构层次范围。注意“{”和“}”必须配对使用。

4) C语言本身没有输入/输出(I/O)语句,输入和输出的操作是通过调用库函数(如scanf和printf)来完成的。

1.3 C程序举例

阅读下面几个典型的C程序,初步了解C程序结构的特点。

【例 1-4】 编写程序,在屏幕上输出“Hello”。

```
#include<stdio.h>
main()
{
    printf("Hello\n");          /* 在屏幕上输出字符串 */
}
```

【例 1-5】 从键盘任意输入一个整数,判断其能否被3整除,如果能整除,则显示“Yes”;否则,显示“No”。

```
#include<stdio.h>
main()
{
    int m;                      /* 定义一个整型变量 m */
    scanf("%d",&m);             /* 从键盘上输入整数赋值给 m */
    if( m%3==0 )                /* 使用选择结构,判断 m 是否能被 3 整除 */
        printf("Yes");
    else
        printf(" No");
}
```

【例 1-6】 在屏幕上输出 100~150 之间能被 3 整除的所有整数。

```
#include<stdio.h>
main()
{
    int m;                                /* 定义一个整型变量 m */
    for( m=100;m<=150;m++)              /* 使用循环结构 */
        if( m%3==0 )
            printf(" %d ",m);
        printf("\n");
}
```

【例 1-7】 编程计算 $1!+2!+\cdots+10!$ 的值。

```
#include<stdio.h>
long ff( int n )                        /* 定义 ff 函数,用于求阶乘 */
{
    long s; int i;
    s=1;
    for( i=1; i<=n; i++ )
        s=s*i;
    return s;
}
main()
{
    long sum; int m;
    sum=0;
    for( m=1;m<=10;m++)
        sum+=ff(m);                    /* 调用 ff 函数,用于求阶乘 */
    printf(" %ld\n ",sum);
}
```

1.4 运行 C 程序的过程

一般把用高级程序设计语言编写的未经编译的程序称为源程序 (source code), 又称为源代码。在 1.3 节已经看到了一些用 C 语言编写的源程序, 但是写出了源程序并不等于问题已经解决了, 因为还没有上机运行, 没有得到最终的结果。一个程序从编写到最后得到运行结果要经历以下步骤。

1. 用 C 语言编写源程序

从语法的角度看, 源程序实际上是一个字符序列。这些字符序列由关键字、常量、运算符、分隔符、程序员自定义的变量名、函数名等组成, 如用于描述数据类型的 `int`, 用于描述分支控制的 `if-else`, 用于描述循环控制的 `for` 语句等。C 语言的源程序是以 `.C` 作为扩展名的。

2. 对源程序进行编译

计算机只能识别和执行机器语言, 而不能识别和执行用高级语言编写的程序。为了