

高职高专计算机系列教材

C语言程序设计

贾学斌 宋海民 主编 胡大威 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机系列教材

C 语言程序设计

贾学斌 宋海民 主 编

胡大威 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书作为 C 语言程序设计的入门与应用教材,共分 13 章,主要包括:C 语言概述、基本数据类型和运算符及表达式、数据的输入输出、结构化程序设计、数组、函数、变量的作用域和存储类型、指针、构造数据类型、文件、编译预处理、位运算、实用项目开发技术简介等。每章后附有习题。书末附有常用字符与 ASCII 代码对照表、运算符的优先级和结合性及常用的 C 语言库函数。有配套实训教材《C 语言程序设计实训教程》。

本书体系合理、概念清晰、例题丰富、逻辑性强、文字流畅、通俗易懂。本书注重基础,突出应用,每章有程序案例,以便读者能够综合运用本章所学知识提高实际编程能力。

本书既可作为本科院校非计算机专业的教材,也可作为高职高专相关专业的课程教材,还可供读者自学使用。

图书在版编目(CIP)数据

C 语言程序设计/贾学斌,宋海民主编. —北京:中国铁道出版社,2007.1
(高职高专计算机系列教材)
ISBN 978-7-113-07776-1

I. C... II. ①贾... ②宋... 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(2006)第 151517 号

书 名: C 语言程序设计

作 者: 贾学斌 宋海民 胡大威

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 秦绪好

责任编辑: 苏 茜 谢立和 高婧雅

封面设计: 高 洋

封面制作: 白 雪

责任校对: 吴媛媛

印 刷: 北京新魏印刷厂

开 本: 787×1092 1/16 印张: 19.25 字数: 449 千

版 本: 2007 年 1 月第 1 版 2007 年 1 月第 1 次印刷

印 数: 1~5 000 册

书 号: ISBN 978-7-113-07776-1/TP·2141

定 价: 25.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

C 语言是近年来在国内外得到迅速推广和使用的一种程序设计语言。C 语言功能丰富、表达能力强、使用方便灵活、应用面广、具有完善的模块程序结构,可移植性好,既有高级语言的特点,又有低级语言的功能,使用 C 语言进行程序设计已成为软件开发的一个主流。因此,我国大部分高等院校都以 C 语言作为学生程序设计入门的一种语言,同时,C 语言也为进一步学习 C++ 以及 Visual C++ 奠定了基础。

本书作为 C 语言程序设计的入门与应用教材,共分 13 章及 3 个附录。主要内容包括: C 语言概述、基本数据类型和运算符及表达式、数据的输入输出、结构化程序设计、数组、函数、变量的作用域和存储类型、指针、构造数据类型、文件、编译预处理、位运算、实用项目开发技术简介等。3 个附录分别是: 常用字符与 ASCII 代码对照表、运算符的优先级和结合性及常用的 C 语言库函数。本书注重基础,突出应用,每章有程序案例,以便读者能够综合运用本章所学知识点提高实际编程能力。

本书具有如下特色:

1. 易于教学和自学,适合初学者

充分考虑到初学者学习 C 语言的难度,本书按照循序渐进、难点分散的原则组织内容。通过通俗易懂的叙述,阐明复杂、灵活的概念。对于难点与重点,通过丰富的例题,进行详尽的解释。力求做到语言通俗、概念清晰、易学实用,使读者能够上手快、学得会、用得着。

2. 注重基础内容,突出实用性

C 语言博大精深。这里我们精选了最基本,对初学者最重要、最实用的内容进行介绍,不刻意追求所谓的全面和详尽。对于较生僻的内容,也从概念讲解入手进行简单介绍,以保证 C 语言的完整性。力求做到内容新颖、实用,逻辑性强,完整性好,且突出重点。

3. 强化编程思想,突出应用性

本书不只是讲解 C 语言的语法规则,更重要的是培养读者的 C 语言程序设计能力。全书始终强化编程思想,通过例题、程序案例、习题,有意识地不断强化,给读者以潜移默化的影响。由于程序设计语言是实践性很强的课程,因此建议读者多上机实践,尽快掌握 C 语言的编程方法和提高调试程序的能力。为了加强读者的实践动手能力以及参加二

级 C 语言等级考试需要, 本书还配套出版了《C 语言程序设计实训教程》。衡量这门课学习的好坏, 不是看“知不知道”, 而是看“会不会用”。

本书介绍的知识 and 程序具有通用性, 基本适用于任何计算机系统和 C 语言版本, 但是应该说明的是, 不同的 C 语言版本是有一些差别的。在 Turbo C 2.0 环境下使用汉字, 需要加载汉字操作系统 (如 UC DOS); 在 Turbo C 3.0 for Windows 环境下可以直接输入汉字。本书所举的全部 C 程序均在 Turbo C 2.0 和 Turbo C 3.0 for Windows 环境下调试通过。

本书既可作为本科院校相关专业的教材, 也可作为高职高专相关专业的课程教材, 还可供读者自学使用。

本书由武汉职业技术学院贾学斌、宋海民主编, 第 1、2、3、9、13 章由宋海民编写; 第 4、5 章由胡大威编写; 第 6、7、8、10、11、12 章及附录由贾学斌编写。以上各编者都是具有丰富教学实践经验的老师。

许多领导及老师对本书的出版给予了热情地支持, 在编写过程中还得到了陈颢、周桂枝、张伟老师的帮助, 在此一并表示感谢。

感谢读者选择使用本书, 由于编者水平有限, 时间仓促, 书中不足之处在所难免, 恳请广大读者批评指正。邮箱地址: jxbin130@sina.com。

编 者

2006 年 8 月

目 录

第 1 章 C 语言概述	1
1-1 C 语言的发展简史	1
1-2 C 语言的特点	2
1-3 C 语言的基本程序结构	3
1-4 算法	6
1-4-1 算法的概念	6
1-4-2 算法的特性	8
1-4-3 算法的描述	9
1-4-4 结构化程序设计	11
1-5 C 语言的语法基础	13
1-5-1 C 语言字符集	13
1-5-2 标识符	14
1-5-3 保留字	14
1-6 赋值语句和基本输入输出语句	15
1-6-1 赋值语句	15
1-6-2 格式输出语句	16
1-6-3 格式输入语句	17
1-7 小结	18
1-8 习题	19
第 2 章 基本数据类型和运算符及表达式	21
2-1 常量与变量	21
2-1-1 常量和符号常量	21
2-1-2 变量	22
2-2 整型数据	23
2-2-1 整型常量	23
2-2-2 整型变量	23
2-3 实型数据	27
2-3-1 实型常量	27
2-3-2 实型变量	28
2-4 字符型数据	30
2-4-1 字符常量	30
2-4-2 转义序列	30
2-4-3 字符串常量	31
2-4-4 符号常量	31
2-4-5 字符型变量	32
2-4-6 字符数据在内存中的存放形式及其使用方法	32

2-5 运算符和表达式	34
2-5-1 运算符的分类	34
2-5-2 运算符的优先级与结合性	34
2-5-3 算术运算符和算术表达式	35
2-5-4 增 1、减 1 运算符及表达式	36
2-5-5 赋值运算符和赋值表达式	38
2-5-6 关系运算符和关系表达式	39
2-5-7 逻辑运算符和逻辑表达式	40
2-5-8 逗号运算符和逗号表达式	42
2-5-9 sizeof 运算符	43
2-6 数据类型的转换	44
2-6-1 隐式类型转换	44
2-6-2 强制类型转换	46
2-7 程序案例	46
2-8 小结	47
2-9 习题	49
第 3 章 数据的输入输出	51
3-1 字符输出函数 putchar	51
3-2 格式输出函数 printf	52
3-2-1 printf 函数的一般格式	52
3-2-2 格式字符	53
3-3 字符输入函数 getchar	58
3-4 格式输入函数 scanf	59
3-4-1 scanf 函数的一般格式	59
3-4-2 格式字符	59
3-4-3 通过 scanf 函数从键盘输入数据	60
3-4-4 scanf 函数输入中常见错误分析	61
3-5 程序案例	62
3-6 小结	63
3-7 习题	63
第 4 章 结构化程序设计	67
4-1 if 语句	67
4-1-1 if 语句的 3 种形式	68
4-1-2 if 语句的嵌套	71
4-1-3 条件运算符和条件表达式	74
4-2 switch 语句	75
4-3 while 语句	78
4-3-1 while 循环的一般格式	78
4-3-2 while 循环的执行过程	78

4-4 do...while 语句	80
4-4-1 do...while 循环的一般格式	80
4-4-2 do...while 循环的执行过程	80
4-4-3 do...while 语句和 while 语句的区别与联系	81
4-5 for 语句	82
4-5-1 for 循环的一般格式	82
4-5-2 for 循环的执行过程	83
4-5-3 for、while、do...while 3 种循环的比较	86
4-6 循环嵌套	86
4-7 break 语句和 continue 语句	87
4-7-1 break 语句	88
4-7-2 continue 语句	89
4-8 语句标号和 goto 语句	90
4-8-1 语句标号	90
4-8-2 goto 语句	90
4-9 程序案例	90
4-10 小结	92
4-11 习题	93
第 5 章 数组	96
5-1 一维数组	97
5-1-1 一维数组的定义	97
5-1-2 一维数组元素的引用	98
5-1-3 一维数组的初始化	99
5-1-4 一维数组的应用	100
5-2 二维数组	103
5-2-1 二维数组的定义	103
5-2-2 二维数组元素的引用	104
5-2-3 二维数组的初始化	104
5-2-4 二维数组的应用	105
5-3 字符数组	106
5-3-1 字符数组的定义	106
5-3-2 字符数组的初始化	106
5-3-3 字符数组的引用	107
5-3-4 字符串和字符串结束标志	107
5-3-5 字符数组的输入输出	108
5-3-6 字符串处理函数	109
5-3-7 字符数组的应用	110
5-4 程序案例	111
5-5 小结	113
5-6 习题	114

第 6 章 函数	117
6-1 函数的概念	117
6-1-1 函数的结构	117
6-1-2 函数的定义	118
6-2 函数的参数与返回值	118
6-2-1 函数的参数	118
6-2-2 返回语句	120
6-3 函数的调用	120
6-3-1 函数调用的一般形式	120
6-3-2 函数调用的方式	121
6-3-3 被调函数的说明	122
6-4 函数的嵌套调用	123
6-5 函数的递归调用	125
6-6 函数与数组	127
6-6-1 一维数组名作实参	127
6-6-2 二维数组名和指针数组作实参	128
6-7 程序案例	129
6-8 小结	130
6-9 习题	131
第 7 章 变量的作用域和存储类型	134
7-1 变量的作用域	134
7-2 变量的存储类型	134
7-2-1 自动变量 (auto)	135
7-2-2 外部变量 (extern)	136
7-2-3 静态变量 (static)	140
7-2-4 寄存器变量 (register)	143
7-3 变量的初始化	144
7-4 程序案例	145
7-5 小结	146
7-6 习题	147
第 8 章 指针	150
8-1 指针的概述	150
8-2 指针变量	151
8-2-1 指针变量的定义与初始化	151
8-2-2 指针变量的引用	153
8-2-3 指针变量的运算	155
8-2-4 指针的应用	157
8-2-5 指针变量作为函数的参数	158

8-3 指针与数组	161
8-3-1 指向数组元素的指针	161
8-3-2 指针和数组的关系	162
8-3-3 指针的下标	163
8-3-4 数组名和指针变量作函数参数	165
8-4 指针与字符串	167
8-4-1 字符串指针变量的定义与使用	167
8-4-2 使用字符串指针变量与使用字符数组的区别	169
8-5 指针与函数	170
8-5-1 函数指针变量	170
8-5-2 指针型函数	171
8-6 指针数组和指向指针的指针	173
8-6-1 指针数组	173
8-6-2 指向指针的指针	174
8-7 main 函数的形参、void 指针和动态存储分配	175
8-7-1 指针数组作为 main 函数的形参	175
8-7-2 指向 void 的指针变量	176
8-7-3 动态存储分配	177
8-8 程序案例	178
8-9 小结	179
8-10 习题	180
第 9 章 构造数据类型	183
9-1 结构体	183
9-1-1 结构体类型说明	183
9-1-2 结构体变量的定义	184
9-1-3 结构体成员的引用和初始化	188
9-2 结构体数组	190
9-2-1 结构体数组的定义	190
9-2-2 结构体数组的初始化	191
9-3 结构体与函数	192
9-3-1 结构体变量作为函数参数	192
9-3-2 结构体变量为函数的返回值	194
9-4 结构体与指针	196
9-4-1 结构体变量指针	196
9-4-2 结构体数组指针	198
9-4-3 用指向结构体的指针作函数参数	200
9-5 链表	200
9-5-1 链表概述	201
9-5-2 链表的基本操作	201

9-6 共用体.....	203
9-6-1 共用体的概念.....	203
9-6-2 共用体变量的引用方式.....	205
9-7 枚举类型.....	205
9-8 用 typedef 定义类型.....	206
9-9 程序案例.....	208
9-10 小结.....	210
9-11 习题.....	211
第 10 章 文件.....	215
10-1 文件概述.....	215
10-1-1 文件的概念.....	215
10-1-2 缓冲文件系统和非缓冲文件系统.....	216
10-2 文件类型指针.....	217
10-3 文件的基本操作.....	218
10-3-1 文件的打开.....	218
10-3-2 文件的关闭.....	220
10-3-3 文件的读函数.....	221
10-3-4 文件的写函数.....	223
10-4 文件的定位函数.....	225
10-5 文件检测函数.....	226
10-6 程序案例.....	226
10-7 小结.....	229
10-8 习题.....	229
第 11 章 编译预处理.....	232
11-1 宏定义.....	232
11-1-1 不带参数的宏定义.....	232
11-1-2 带参数的宏定义.....	234
11-1-3 终止宏定义.....	237
11-2 文件包含.....	237
11-3 条件编译.....	240
11-4 程序案例.....	242
11-5 小结.....	243
11-6 习题.....	244
第 12 章 位运算.....	248
12-1 位运算符与位运算.....	248
12-1-1 “按位与”运算.....	248
12-1-2 “按位或”运算.....	249
12-1-3 “按位异或”运算.....	250

12-1-4 “按位取反”运算.....	251
12-1-5 “左移”运算	251
12-1-6 “右移”运算	252
12-1-7 位运算赋值运算符.....	252
12-2 位段.....	252
12-3 程序案例.....	254
12-4 小结.....	255
12-5 习题.....	255
第 13 章 实用项目开发技术简介.....	259
13-1 图形应用.....	259
13-1-1 图形与硬件	259
13-1-2 文本与图形	259
13-1-3 图形设计	263
13-1-4 C 语言动画设计.....	264
13-2 菜单设计技术.....	267
13-2-1 下拉式菜单的设计.....	267
13-2-2 选择式菜单的设计.....	270
13-3 程序的组织与管理	271
13-4 软件开发过程.....	272
13-4-1 可行性和需求分析.....	273
13-4-2 系统设计	274
13-4-3 软件编码	275
13-4-4 软件测试	276
13-5 小结.....	276
附录 A 常用字符与 ASCII 代码对照表	279
附录 B 运算符的优先级和结合性.....	280
附录 C Turbo C 2.0 函数库分类列表	281
参考文献	295

第 1 章 C 语言概述

本章要点

- C 语言的发展简史
- C 语言的特点
- C 语言的基本程序结构
- 算法
- C 语言的语法基础
- 赋值语句和基本输入输出函数

本章难点

- 算法的描述

C 语言是一种面向过程的通用程序设计语言。它以表达简明、使用灵活、结构化的流程控制、丰富的数据结构和操作集合、良好的程序可移植性和较高效率的目标代码为特征。C 语言不仅具有高级语言的要素，还兼有低级语言的功能，因此既可用于编写系统程序，也可用于编写不同领域的应用程序。

1-1 C 语言的发展简史

C 语言的产生与 UNIX 系统的发展有密切的关系。UNIX 操作系统是美国贝尔实验室的 K.Thompson 和 D.M.Ritchie 于 1971 年设计成功的，第 1 版是在 GE635 机器上产生并通过纸带将可执行代码传送到 PDP-7 上的，后又以汇编语言编写，由于汇编语言不可移植，描述问题效率不高，且可读性差，因此 K.Thompson 决定开发一种高级语言来描述 UNIX 操作系统。这就是 C 语言产生的历史背景。

1972 年 C 语言投入使用，1973 年 K.Thompson 和 D.M.Ritchie 将 UNIX 操作系统用 C 语言进行改写。UNIX 操作系统由于使用了 C 语言而取得成功，几乎成为 16 位微机的标准操作系统，C 语言一诞生就由于编制 UNIX 操作系统的成功而引起了人们的关注。

1973 年之后，C 语言的发展相当迅速：1975 年 UNIX 第 6 版公布，1977 年又研制成功不依赖具体机器的 C 语言编译文本——可移植 C 语言编译程序，推动了 C 语言在各种机型上的广泛应用；UNIX 第 7 版在 1978 年研制成功。以其中的 C 编译系统为基础，美国国家标准化协会于 1983 年制定了一个 C 语言标准草案，即 83 ANSI C，1987 年又公布了 87 ANSI C。1990 年 ISO 通过了 C 程序设计语言国际标准，称之为标准 C。此后陆续出现的各种 C 语言版本，如 Microsoft C 5.0/9.0、Turbo C 2.0/3.0、Quick C 等都是与 ANSI C 兼容的版本。它们的语法和语句功能是一致的，差异表现在各自的标准函数库中收纳的函数种类、格式和功能上，尤其是图形函数库的差异更大一些。

美国 Borland 公司在 Turbo C 的基础上，研制成功了面向对象的程序设计语言 Turbo C++，并应用于 IBM PC 上。它继承发展 Turbo C 2.0 的集成开发环境，并包含了面向对象的基本思想和设计方法。

1991 年为了适用 Microsoft 公司的 Windows 3.0 版本，Borland 公司又将 Turbo C++做了更新，即 Turbo C 的新一代产品 Borlandc C++问世了。

1-2 C 语言的特点

C 语言以其简洁、灵活、表达能力强、产生的目标代码质量高、可读性强和可移植性好为基本特点而著称于世。详细归纳有如下十大特点。

(1) C 语言程序紧凑、简洁、规整。使用一些简单规则和方法就可以构成相当复杂的数据类型、语句和程序结构。

(2) C 语言的表达式简练、灵活、实用。C 语言有多种运算符、多种描述问题的途径和多种表达式求值的方法,这使程序设计者有较大的主动性,并能提高程序的可读性、编译效率以及目标代码的质量。

(3) C 语言具有与汇编语言很相近的功能和描述问题的方法。如地址计算、二进制数位运算、使用寄存器存放变量以及对硬件端口直接操作等,都充分利用了计算机系统资源(如 BIOS 软中断和 DOS 的系统功能调用等)。

(4) C 语言具有丰富的数据类型。在系统软件中,特别是操作系统中,对计算机的所有软件、硬件资源要实施管理和调度,这就要求有相应的数据结构作为操作基础。C 语言具有 5 种基本的数据类型:char(字符型)、int(整型)、float(浮点单精度型)、double(浮点双精度型)、void(无值型)和多种构造数据类型(数组、指针、结构体、共用体、枚举)。如指针类型使用十分灵活,用它可以构成链表、树、栈等。指针可以指向各种类型的简单变量、数组、结构体、共用体以及函数等。

(5) C 语言具有丰富的运算符。C 语言有多达 40 余种运算符。丰富的数据类型与众多的运算符相结合,使 C 语言具有表达灵活和效率高的优点。

(6) C 语言是一种结构化程序设计语言,特别适合大型程序的模块化设计。C 语言具有编写结构化程序所必需的基本流程控制语句。C 语言程序是由函数集合构成的,函数各自独立作为模块化设计的基本单位。它所包含的源文件,可以分割成多个源程序,分别对其进行编译,然后连接起来构成可执行的目标文件。C 语言提供了丰富的库函数,包括图形函数等,可供用户调用。C 语言还允许用户根据需要自定义函数。C 语言还提供了多种存储属性,可以使数据按其需要在相应的作用域内起作用。

(7) C 语言为字符、字符串、集合和表的处理提供了良好的基础,它能够表示和识别各种可显示的以及起控制作用的字符,也能区分和处理单个字符和字符串。

(8) C 语言具有预处理程序和预处理语句,给大型程序的编写和调试提供了方便。

(9) C 语言程序具有较高的可移植性。在 C 语言的语句中,没有依赖于硬件的输入输出语句,程序的输入输出功能是通过调用输入输出函数实现的,而这些函数是由系统提供的独立于 C 语言的程序模块,从而便于在硬件结构不同的计算机之间实现程序的移植。

(10) C 语言是处于汇编语言和高级语言之间的一种中间型的记述性程序设计语言。C 语言既具有面向硬件和系统,像汇编语言那样可以直接访问硬件的功能,又有高级语言面向用户,容易记忆、便于阅读和书写的优点。

由于 C 语言具有上述诸多特点,近年来迅速得到广泛地普及和应用。本书以目前广为流行的 Turbo C 2.0 版本为基础,同时兼顾其他不同版本中的通用性、一致性的内容予以叙述。

综上所述,C 语言是一种功能很强的语言。但是,它也有一些不足之处:C 语言语法

限制不严谨,虽然熟练的程序员编程灵活,但安全性低;运算符丰富,完成功能强,但难记、难掌握。因此,学习、使用C语言不妨先学基本部分,先用起来,用熟练后再学不规范的语法规则,进而全面掌握C语言。

1-3 C语言的基本程序结构

本节将通过下面这个例题说明一下C语言程序的结构、特点及其设计风格。

【例 1-1】下列程序完成从算式 $1+2+3+\dots+N$ 中找出一个其和大于 5 000 的最小整数 N。

```
1 /* example 1.1 cl-1.c */
2
3 #include <stdio.h>
4 #define SUM_VALUE 5000
5
6 /* This function calculates the sum of 1+2+3+...+N.*/
7 sum()
8 {
9     int total=0,n=0;
10    while(total<=SUM_VALUE)
11    {
12        N++;
13        total=total+n;
14    }
15    return(n);
16 }
17
18 /* This is the main program.*/
19 main()
20 {
21    printf("This least value of N is %d.\n",sum());
22 }
```

将上述程序输入计算机并命名为 cl-1.c, 存入磁盘, 编译运行结束后, 按【Alt+F5】键, 屏幕上显示出下面信息:

```
This least value of N is 100.
```

上述程序虽不复杂, 但可以说明C语言程序的结构及其特点。现将其总结归纳如下:

1. 函数与主函数

C语言中函数的概念类似于其他语言中的子程序概念, 有所不同的是C语言中的主程序也被称为函数, 并且被定义成名字为 main 的主函数。因此, 一个C语言程序是由一个或多个函数组成, 其中必须有一个主函数 main。一个可执行的C语言程序总是从 main 函数开始执行。

C语言的函数有两类: 一类是用户定义的函数; 另一类是系统提供的标准函数, 在例 1-1 中所用到的系统标准函数是 printf(), 它是系统提供的面向标准输出设备的格式输出函数。而 sum() 和 main() 是两个用户定义的函数, 其中 main() 是该程序的主函数。函数名后是由圆括号“()”括起来的形式参数表。

2. 注释

为了使程序易读, 在C语言程序中, 设置注释部分, 帮助阅读和理解程序。程序中以

“/*”开头到“*/”结尾中的所有内容表示的意义是一个注释。在编译时注释行被忽略掉，并不产生代码行。注释可以写在程序的任何位置，“/*”与“*/”可以不在同一行上。如果注释写在程序的开头，一般用来说明整个程序的功能或注意事项等。也可以嵌在程序语句行中获某行语句的尾部，一般用来说明一段程序的功能或某行程序的作用。当然，注释越多，程序的可读性和可维护性也就越好。

3. 花括号“{”和“}”的作用

花括号在 C 语言程序中是用来构成函数体和复合语句（也叫程序块，即逻辑上相关的一组语句的集合）的分隔符。一个 C 语言程序中至少要有一对花括号，以表示程序体的开始和结束。

4. 程序语句

在 C 语言程序中，一条完整的语句必须以分号“;”结束。C 语言程序语句可分为下面 4 类。

(1) 说明语句

用来说明变量的类型和初值。例如，例 1-1 中的语句：

```
int total=0,n=0;
```

它说明 total 和 n 都是整型变量，并且初值都为 0。并列的标识符或项之间用逗号“,”分隔；两个关键词相邻时，中间用空格（至少一个）相间。

(2) 表达式语句

用来描述逻辑运算、算术运算或产生某种特定动作的语句。例如，程序中的语句：

```
n++;  
total=total+n;  
printf("This least value of N is %d.\n",sum());
```

这些都是表达式语句。Printf 函数是格式输出函数，此处是调用该函数完成输出功能。双引号内的字符串原样输出。“\n”表示回车换行。

(3) 程序控制语句

用来描述语句的执行条件与执行顺序的语句。例如，程序中的语句：

```
while(total<=SUM_VALUE)  
{循环体语句}
```

就是控制语句。

(4) 复合语句

由花括号“{”和“}”括起来的逻辑上相关的一组语句。例如，程序中的语句：

```
{  
    n++;  
    total=total+n;  
}
```

即为一个复合语句。

5. 大小写字母敏感性

C 语言惯用小写字母。例如，C 语言中的保留字及系统提供的标准库中所有函数的名称均使用小写字母。在 C 语言中，变量 A 和 a 是两个完全不同的变量，因为它们在内存中所分配的不是同一个地址。因此，字母大小写的敏感性应在 C 语言程序设计中引起足够的重视。

另外，在 C 语言中，以下划线“_”字符开头的标识符一般由内部使用，所以，在 C 语言程序设计时，最好不要用下划线字符作为标识符的开头字符。

6. 程序的书写格式

C语言在书写时是比较自由的，几个说明项或几个语句可以写在一行，一个语句可以分为几行写，但是一个词或一个数不能分两行写。例如，程序中的sum函数完全可以写成下面的形式：

```
sum(){ int total=0,n=0;
while(total<=SUM_VALUE)
{ n++;total=total+n;}return(n);}
```

这种写法虽然允许，并且也是正确的，但却不如前面所写的那样在逻辑上清晰易懂。因此，良好的C语言程序书写格式，会给程序带来可读性和可操作性。建议使用锯齿形的程序书写风格，因为这种书写风格具有较强的层次感和逻辑性。

7. 预处理特性

在C语言中除了上面所述的4类语句外，还有一类语句，这类语句的作用不是实现程序的功能，而是给C编译系统发布信息，它告诉C编译系统在对源程序进行编译之前应该做些什么。所以，这类语句被称为编译预处理语句。这类语句以“#”号开头，占用一个单独的书写行。

在例1-1中使用了下面两个类型的预处理语句：

```
#include <stdio.h>
#define SUM_VALUE 5000
```

现将这两个语句作一个简单介绍：`#include <stdio.h>`是包含预处理语句。它指明C编辑器在对源程序编译之前，用`stdio.h`文件中的内容来取代该预处理语句，使之成为源程序的一部分。`stdio.h`称为包含文件或标题文件。在C语言的编译系统中，提供了若干个包含文件。包含文件以`.h`为后缀，且为文本文件。在`stdio.h`文件中，有使用标准输入输出函数时所需要的函数说明语句和符号常量的定义等。

`#define SUM_VALUE 5000`是宏定义预处理语句，其功能是将一个符号常量`SUM_VALUE`定义成数字量`5000`。在整个程序执行期间，`SUM_VALUE`被当作数字`5000`来使用。

8. 容易犯的错误

在C语言程序设计中，语句的结束符为分号“;”。但在下面3种情况下不允许有分号。

- (1) 预处理语句后面不使用分号。
- (2) 所定义的函数名称后面不使用分号。
- (3) 在右花括号“}”后面不使用分号。

因此，下面横线上的分号均是错误的：

```
#include <stdio.h>_;  
#define SUM_VALUE 5000_;  
sum();_  
{ int total=0,n=0;  
while(total<=SUM_VALUE)  
{ n++;total=total+n;}_;  
return(n);  
}_;
```

9. 了解程序功能

在了解了C语言程序结构、特点及其设计风格之后，来解释一下`cl-1.c`程序的功能（见