

微机应用  
基本技能丛书



# C语言入门

孝延忠·尤枫 编



C语言入门

TP312C/899

出版社

化学工业出版社

微机应用基本技能丛书

# C 语言入门

孝延忠 尤 枫 编

化学工业出版社

· 北 京 ·

人大复印  
章自具的牛

## (京)新登字 039 号

**内容简介** 本书用简单的、易理解的方式阐明 C 语言功能丰富、灵活方便等诸多特点,书以简明、求实的特点向读者展示 C 语言的编程规则,精选例子配合叙述,一步步地将读者引入 C 语言之精华所在。本书第一章至第三章叙述数据、数制、逻辑、运算和存储等基本概念;第四章为基本程序设计;第五章、第六章为数组和函数;第七章为变量和函数的作用范围及预编译;第八章为复合数据结构,即结构体与共用体;第十章为文件的操作。附录包括 ASCII 码对照表;C 语言关键词;运算符和结合性;常用 C 语言库函数。本书适合于初学计算机编程的读者,也适合于非计算机专业的教师和学生以及业余计算机编程爱好者使用。

### 图书在版编目(CIP)数据

C 语言入门 / 孝延忠, 尤枫编. —北京: 化学工业出版社, 1996  
(微机应用基本技能丛书 / 董小国等主编)

ISBN 7-5025-1656-5

I. C… II. ①孝… ②尤… III. C 语言-基本知识  
IV. TP 312C

中国版本图书馆 CIP 数据核字 (96) 第 02083 号

出版发行: 化学工业出版社 (北京市朝阳区惠新里 3 号)

社长: 俸培宗 总编辑: 蔡剑秋

经 销: 新华书店北京发行所  
印 刷: 北京云浩印制厂  
装 订: 三河市前程装订厂  
版 次: 1996 年 3 月第 1 版  
印 次: 1997 年 7 月第 2 次印刷  
开 本: 850×1168 1/32  
印 张: 7  
字 数: 178 千字  
印 数: 5201—10000  
定 价: 14.00 元

# 编者的话

从 1946 年 2 月第一台电子计算机诞生以来，只有短短 50 年的历史。在这 50 年里，计算机已使人们的生活发生了巨大的变化，特别是 1971 年开始出现的微机（电脑），其发展之快速、种类之繁多、用途之广泛、效益之巨大，使人类的任何其他发明创造都黯然失色。

微机的应用已远远不限于科学家进行数值计算，人们利用微机还可以进行工业控制、信息管理、数据库检索查询、仿真教育、辅助设计与制造、医疗诊断、分析与预测、广告制作、图文演示、文稿编辑排版等等。随着微机体积不断缩小、功能不断增加、价格不断降低、软件不断增多，微机的利用面已经并必将继续在各行各业不断扩大和拓展。近年来，微机与边缘科学的结合使它的应用更具活力，例如：微处理器放在仪表中可做成智能仪表；微机与媒体技术的结合成为人们休闲娱乐的家用电器；微机与现代通讯设备及网络的结合，使人们可以轻而易举地跨接到国际互连网上，完成远程通信、对话、传阅图象和文件、登录检索海量信息。总之，现代社会从科学家到普通百姓，从白发老人到咿呀学语的幼儿，都能感觉到微机的魅力和神奇。

即将步入 21 世纪的人们现已有了一个共同的观点：计算机是人人必需掌握的工具，计算机操作技能应该作为一种特殊文化进行普及教育。

微机的普及教育是一个相当长远的战略任务。为此，化学工业出版社调集长期使用微机进行科研教学的同志编写了本套《微机应用基本技能丛书》。全书共分 15 册。

第一部分：微机基础，包括《DOS 操作系统入门》、《五笔字型入门》、《方便实用的文字编辑 WPS》和《计算机常用英文提示及菜单解说》。

第二部分：语言与数据库，包括《BASIC 语言入门》、《C 语言入门》和《FoxPro 数据库入门》。

第三部分：Windows 及窗口软件，包括《Windows 入

门》、《Word 入门》、《Excel 入门》和《Lotus 1-2-3 入门》。

第四部分：平面设计与动画，包括《平面设计入门》和《动画制作入门》。

第五部分：其他应用软件，包括《AutoCAD 入门》和《北大方正排版入门》。

丛书力求通俗易懂，由浅入深，适合于初学者自学，也是专业技术人员的好帮手。

丛书中的不妥之处，请批评指正。

编 者

1996 年 3 月

# 目 录

## 编者的话

第一章 计算机基础知识.....	1
第一节 数制与数制间的转换.....	1
1. 十进制数.....	1
2. 二进制数.....	1
3. 八进制数.....	2
4. 十六进制数.....	2
5. 二进制数与八进制数、十六进制数之间的转换.....	3
6. 二进制数、八进制数、十六进制数转换成十进制数.....	5
7. 十进制数转换成二进制数、八进制数、十六进制数.....	6
第二节 计算机中字符的表示方法.....	7
第三节 几种基本的逻辑运算.....	8
1. 逻辑“非”运算(NOT).....	8
2. 逻辑“与”运算(AND).....	8
3. 逻辑“或”运算(OR).....	9
4. 逻辑“异或”运算(XOR).....	9
5. 常见的几种逻辑操作.....	10
6. 一个重要的逻辑定理.....	11
第四节 数据在内存中的存储方式.....	12
第二章 C语言概述.....	15
第一节 C语言历史和发展.....	15
第二节 C语言的特点.....	15
第三节 简单的C语言程序介绍.....	17
第三章 数据类型、运算符与表达式.....	21

第一节 C 语言的数据类型.....	21
第二节 常量.....	22
1. 整型常量.....	23
2. 实型常量.....	23
3. 字符常量.....	24
4. 字符串常量.....	25
第三节 变量.....	25
1. 整型变量.....	28
2. 实型变量.....	29
3. 字符变量.....	30
第四节 变量赋初值.....	32
第五节 各类数值型数据间的混合运算.....	33
第六节 算术运算符和算术表达式.....	35
1. 基本的算术运算符.....	35
2. 算术表达式.....	35
3. 运算符的优先级.....	35
4. 运算符的结合性.....	36
5. 强制类型转换运算符.....	36
6. 自增、自减运算符.....	37
第七节 赋值运算符和赋值表达式.....	39
1. 复合的赋值运算符.....	40
2. 赋值表达式.....	40
第八节 逗号运算符和逗号表达式.....	41
第九节 关系运算符和关系表达式.....	42
1. 关系运算符及其优先级.....	42
2. 关系表达式.....	43
第十节 逻辑运算符和逻辑表达式.....	44
1. 逻辑运算符及其优先级.....	44
2. 逻辑表达式.....	44
第四章 C 语言程序设计.....	47
第一节 程序的三种基本结构.....	49
第二节 数据输出与输入.....	51
1. putchar 函数 (字符输出函数).....	52

2. getchar 函数 (字符输入函数) .....	54
3. printf 函数 (格式输出函数) .....	55
4. scanf 函数 (格式输入函数) .....	62
第三节 if 条件分支语句 .....	66
1. if 格式 .....	66
2. if-else 格式 .....	67
3. else-if 格式 .....	67
4. if 语句的嵌套 .....	68
第四节 switch 多路分支语句 .....	71
第五节 for 循环语句 .....	74
第六节 while 循环语句 .....	77
第七节 do-while 循环语句 .....	78
第八节 goto 语句、break 语句与 continue 语句 .....	79
1. break 语句 .....	79
2. continue 语句 .....	80
3. goto 语句 .....	81
4. 总结 .....	83
<b>第五章 数组 .....</b>	<b>87</b>
<b>第一节 一维数组 .....</b>	<b>87</b>
1. 一维数组的定义方式 .....	87
2. 一维数组元素的引用 .....	88
3. 一维数组的初始化 .....	89
<b>第二节 二维数组及多维数组 .....</b>	<b>92</b>
1. 二维数组的定义 .....	92
2. 二维数组的引用 .....	93
3. 二维数组的初始化 .....	94
<b>第三节 字符型数组 .....</b>	<b>96</b>
1. 字符数组的定义 .....	96
2. 字符型数组的初始化 .....	97
3. 字符数组的引用 .....	97
4. 字符串和字符串结束标志 .....	98
5. 字符数组的输入输出 .....	100
6. 常用字符串处理函数 .....	102



<b>第六章 函数</b> .....	113
<b>第一节 函数的定义形式</b> .....	116
1. 无参函数的定义形式.....	116
2. 有参函数定义的一般形式.....	117
3. 空函数定义的形式.....	118
<b>第二节 函数的参数</b> .....	118
<b>第三节 函数的返回值</b> .....	119
<b>第四节 函数调用</b> .....	120
1. 函数调用的一般形式.....	120
2. 函数调用的方式.....	120
3. 对被调用函数的说明.....	121
<b>第五节 函数的嵌套调用</b> .....	121
<b>第六节 函数的递归调用</b> .....	123
<b>第七节 数组作为函数的参数</b> .....	126
<b>第八节 系统提供的常用数学函数</b> .....	127
<b>第七章 变量和函数的作用范围及预编译</b> .....	129
<b>第一节 局部变量和全局变量</b> .....	129
1. 局部变量.....	129
2. 全局变量.....	130
<b>第二节 动态存储变量与静态存储变量</b> .....	131
1. 局部变量的存储方式.....	132
2. 全局变量的存储方式.....	133
<b>第三节 内部函数和外部函数</b> .....	135
1. 内部函数.....	135
2. 外部函数.....	136
<b>第四节 宏定义</b> .....	136
1. 不带参数的宏定义.....	136
2. 带参数的宏定义.....	137
<b>第五节 包含文件</b> .....	139
<b>第六节 条件编译</b> .....	140
1. #ifdef 标识符.....	140
2. #ifndef 标识符.....	141
3. #if 表达式.....	142

<b>第八章 指针</b> .....	143
<b>第一节 指针的概念</b> .....	143
1. 直接访问内存 .....	143
2. 间接访问内存 .....	144
<b>第二节 指针变量</b> .....	145
1. 指针变量的定义 .....	145
2. 指针变量的引用 .....	145
<b>第三节 指针变量作函数的参数</b> .....	147
<b>第四节 数组指针</b> .....	150
1. 定义数组指针 .....	150
2. 通过指针引用数组元素 .....	151
3. 数组名做函数的参数 .....	152
4. 多维数组指针 .....	154
<b>第五节 字符串指针</b> .....	154
<b>第六节 返回指针值的函数</b> .....	155
<b>第七节 指针数组和指向指针的指针</b> .....	156
1. 指针数组和指向指针的指针 .....	156
2. main 函数的形参 .....	157
<b>第九章 结构体与共用体</b> .....	159
<b>第一节 结构体的定义及引用</b> .....	159
1. 先定义结构体类型再定义变量名 .....	160
2. 在定义类型的同时定义变量 .....	160
3. 定义结构类型变量 .....	161
4. 结构体类型变量的引用 .....	163
5. 结构体变量的初始化 .....	163
<b>第二节 结构体数组</b> .....	164
1. 结构体数组的定义 .....	164
2. 结构体数组的初始化 .....	165
<b>第三节 指向结构体的指针</b> .....	166
1. 指向结构体变量的指针 .....	166
2. 指向结构体数组的指针 .....	167
3. 用指向结构体的指针作函数参数 .....	167
<b>第四节 指针链表</b> .....	167

1. 链表的数据结构 .....	168
2. 链表的操作 .....	168
第五节 位段操作 .....	172
1. 位运算符 .....	172
2. 位段 .....	173
第六节 共用体的定义及引用 .....	175
1. 共用体的概念 .....	175
2. 共用体变量的引用方式 .....	176
3. 共用体类型数据的特点 .....	177
第七节 枚举类型 .....	178
第八节 用 typedef 定义数据类型 .....	180
<b>第十章 文件操作 .....</b>	<b>183</b>
第一节 关于 C 语言文件 .....	183
第二节 文件类型指针 .....	184
第三节 文件的打开与关闭 .....	185
1. 文件的打开(fopen 函数) .....	186
2. 文件的关闭(fclose 函数) .....	189
第四节 文件的读与写操作 .....	189
1. fputc 函数和 fgetc 函数 .....	190
2. fread 函数和 fwrite 函数 .....	190
3. fprintf 函数和 fscanf 函数 .....	191
4. getw 函数和 putw 函数 .....	191
5. fgets 函数和 fputs 函数 .....	192
第五节 文件定位 .....	192
1. rewind 函数 .....	192
2. fseek 函数 .....	192
3. ftell 函数 .....	193
<b>附录 I .....</b>	<b>199</b>
<b>附录 II .....</b>	<b>201</b>
<b>参考文献 .....</b>	<b>205</b>

# 第一章 计算机基础知识

## 第一节 数制与数制间的转换

### 1. 十进制数

在日常生活中，最常见的数制是十进制数，即数的元素有0、1、2、3、4、5、6、7、8、9，遵循逢十进一的规则。任意一个十进制数可表示为：

$$a_n a_{n-1} \cdots a_0 . b_1 b_2 \cdots b_m$$

其含义是：

$$a_n \times 10^n + a_{n-1} \times 10^{n-1} + \cdots + a_0 \times 10^0 + b_1 \times 10^{-1} + b_2 \times 10^{-2} + \cdots + b_m \times 10^{-m}$$

其中  $a_i (i=0, 1, \cdots, n)$ 、 $b_j (j=1, 2, \cdots, m)$  是0~9十个数码中的一个。

给出一个十进制数“1234.567”，这只是对该数的一个常规的写法，实际上，其大小具体为：

$$1234.789 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2} + 9 \times 10^{-3}$$

并不就只有十进制一种数制，常见的还有：六十进制数，即逢六十进一，用来表示时间上的“分”、“秒”；二十四进制数，即逢二十四进一，用来表示时间上的“小时”到“天”的进位等等。

在计算机中采用的数制为二进制数、八进制数和十六进制数。

### 2. 二进制数

为便于存储及计算，计算机中采用了二进制数。二进制

数的元素有 0、1，遵循逢二进一的规则。任意一个二进制数可表示为：

$$a_n a_{n-1} \cdots a_0 . b_1 b_2 \cdots b_m$$

其含义是：

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_0 \times 2^0 + b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots + b_m \times 2^{-m}$$

其中  $a_i (i=0, 1, \cdots, n)$ 、 $b_j (j=1, 2, \cdots, m)$  是 0、1 二个数码中的一个。

例如，二进制数“1011.101”，其具体大小为：

$$1011.101 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

### 3. 八进制数

因为用二进制数表示一个值书写起来比较长，不便于使用，所以在早期的计算机中也用到了八进制数。八进制数的元素有 0、1、2、3、4、5、6、7，遵循逢八进一的表示规则。任意一个八进制数可表示为：

$$a_n a_{n-1} \cdots a_0 . b_1 b_2 \cdots b_m$$

其含义是：

$$a_n \times 8^n + a_{n-1} \times 8^{n-1} + \cdots + a_0 \times 8^0 + b_1 \times 8^{-1} + b_2 \times 8^{-2} + \cdots + b_m \times 8^{-m}$$

其中  $a_i (i=0, 1, \cdots, n)$ 、 $b_j (j=1, 2, \cdots, m)$  是 0~7 八个数码中的一个。

一个八进制数“1267.023”，其大小具体为：

$$1237.023 = 1 \times 8^3 + 2 \times 8^2 + 3 \times 8^1 + 7 \times 8^0 + 0 \times 8^{-1} + 2 \times 8^{-2} + 3 \times 8^{-3}$$

### 4. 十六进制数

随着计算机硬件的快速发展，现代计算机中很少再见到八进制数，更常见的是十六进制数。十六进制数的元素有 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。遵循逢十六进一的表示规则。任意一个十六进

制数可表示为:

$$a_n a_{n-1} \cdots a_0 . b_1 b_2 \cdots b_m$$

其含义是:

$$a_n \times 16^n + a_{n-1} \times 16^{n-1} + \cdots + a_0 \times 16^0 + b_1 \times 16^{-1} + b_2 \times 16^{-2} + \cdots + b_m \times 16^{-m}$$

其中  $a_i (i=0, 1, \dots, n)$ 、 $b_j (j=1, 2, \dots, m)$  是 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个数码中的一个。

值得注意的是, 这里的 A、B、C、D、E、F 已不再是字符或字母, 它表示的是一个数的大小, 无所谓大小写, 但通常用大写表示。

一个十六进制数 “10AB.40F”, 其大小具体为:

$$10AB.40F = 1 \times 16^3 + 0 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 4 \times 16^{-1} + 0 \times 16^{-2} + 15 \times 16^{-3}$$

## 5. 二进制数与八进制数、十六进制数之间的转换

因为八和十六均为二的倍数, 所以二进制数与八进制数、十六进制数之间的转换相对比较容易。

### (1) 二进制数转换为八进制数

3 个二进制位正好可以完全表示一个八进制位 (见表 1-1), 所以在进行二进制数转换为八进制数时, 可以以小数点为分界点。小数点左侧, 从左向右每 3 个二进制位对应一个八进制位, 不足 3 个二进制位的用 “0” 在左侧补齐; 小数点右侧, 从右向左每 3 个二进制位对应一个八进制位, 不足 3 个二进制位的用 “0” 在右侧补齐。

表 1-1 二进制数与八进制数对照表

二进制数	000	001	010	011	100	101	110	111
八进制数	0	1	2	3	4	5	6	7

## 例 1-1

将二进制数 11010011.011 转换为八进制数。

二进制数 11010011.011 可表示为 011,010,011.011。从表 1-1 中可查到：011、010、011、011 分别表示八进制的 3、2、3、3，所以相应的八进制数为 323.3。

## (2) 二进制数转换为十六进制数

4 个二进制位正好可以完全表示一个十六进制位（见表 1-2），所以在进行二进制数转换为十六进制数时，可以以小数点为分界点。小数点左侧，从左向右每 4 个二进制位对应一个十六进制位，不足 4 个二进制位的用“0”在左侧补齐；小数点右侧，从右向左每 4 个二进制位对应一个十六进制位，不足 4 个二进制位的用“0”在右侧补齐。

表 1-2 二进制数与十六进制数对照表

二进制数	0000	0001	0010	0011	0100	0101	0110	0111
十六进制数	0	1	2	3	4	5	6	7
二进制数	1000	1001	1010	1011	1100	1101	1110	1111
十六进制数	8	9	A	B	C	D	E	F

## 例 1-2

将二进制数 11010011.011 转换为十六进制数。

二进制数 11010011.011 可表示为 1101,0011.011。从表 1-2 中可查到：1101、0011、0110 分别代表十六进制的 D、3、6，所以相应的十六进制数为 D3.6。

## (3) 八进制数转换为二进制数

把八进制数转换为二进制数，只需要将每个八进制位用其相对应的 3 个二进制位表示，得到的结果即为相应的二进制数。

## 例 1-3

将八进制数“34.7”转换为二进制数。

八进制的 3、4、7 分别等于二进制的 011、100、111，所以相应的二进制数为 011100.111。

(4) 十六进制数转换为二进制数

把十六进制数转换为二进制数，只需要将每个十六进制位用其相对应的 4 个二进制位表示，得到的结果即为相应的二进制数。

#### 例 1-4

将十六进制数“3A.F9”转换为二进制数。

十六进制的 3、A、F、9 分别等于二进制的 0011、1010、1111、1001，所以相应的二进制数为 00111010.11111001。

## 6. 二进制数、八进制数、十六进制数转换成十进制数

要将其他进制的数转换成十进制数相对来说比较容易，只要将其数的元素转换成十进制格式，然后按位代入其含义公式进行计算，得到的就是其相应的十进制数格式。

以下 3 个公式分别为二进制、八进制、十六进制的含义公式：

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \dots + a_0 \times 2^0 + b_1 \times 2^{-1} + b_2 \times 2^{-2} + \dots + b_m \times 2^{-m}$$

$$a_n \times 8^n + a_{n-1} \times 8^{n-1} + \dots + a_0 \times 8^0 + b_1 \times 8^{-1} + b_2 \times 8^{-2} + \dots + b_m \times 8^{-m}$$

$$a_n \times 16^n + a_{n-1} \times 16^{n-1} + \dots + a_0 \times 16^0 + b_1 \times 16^{-1} + b_2 \times 16^{-2} + \dots + b_m \times 16^{-m}$$

#### 例 1-5

将二进制数“1011.011”、八进制数“63.7”、十六进制数“4F.A5”分别转换为十进制数。

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 11.375$$





$$6 \times 8^1 + 3 \times 8^0 + 7 \times 8^{-1} = 51.875$$

$$4 \times 16^1 + 15 \times 16^0 + 10 \times 16^{-1} + 5 \times 16^{-2} = 79.6443125$$

## 7. 十进制数转换成二进制数、八进制数、十六进制数

将十进制数转换成其他进制数，一般常用的是除法。假设将十进制数转换成 N 进制数，对于整数部分，只要用 N 去除该十进制数的整数部分，得到一个除数和余数，保存余数并继续用 N 去除所得的除数，按倒序保存所得的余数，直至除数为零，所得余数的倒序排列即为转换后的 N 进制数的整数部分；对于小数部分，只要用 N 去乘该十进制数的小数部分，保存所得乘数的整数位，并将其整数位清零，再用 N 去乘该乘数，并按正序排列其整数部分，直至达到所要求的精度为止，所得按正序排列的乘数的整数部分即为转换后的 N 进制数的小数部分。

 注意：

小数部分的转换一般不能达到完全转换，只能按精度要求达到近似的结果。

### 例 1-6

将十进制数 34.68 转换成相应的二进制数、八进制数和十六进制数格式。

商	余数	二进制数的整数部分
$34/2 = 17$	0	0
$17/2 = 8$	1	10
$8/2 = 4$	0	010
$4/2 = 2$	0	0010
$2/2 = 1$	0	00010
$1/2 = 0$	1	100010
积	整数部分	二进制数的小数部分
$0.68 \times 2 = 1.36$	1	1