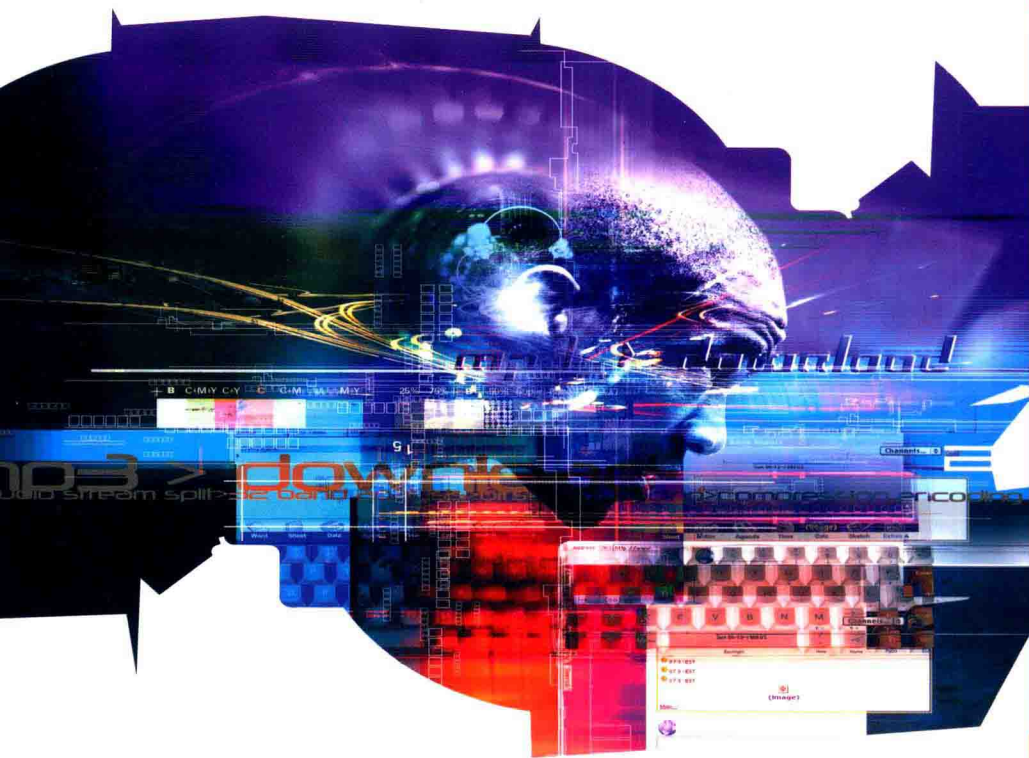




信息素养文库·高等学校信息技术系列课程规划教材

# C 语言 二级考试辅导教程

田祥宏 编著



南京大学出版社



信息素养文库·高等学校信息技术系列课程规划教材

# C 语言 二级考试辅导教程

田祥宏 编著



南京大学出版社

## 图书在版编目(CIP)数据

C语言二级考试辅导教程 / 田祥宏编著. —南京:  
南京大学出版社, 2018.12

(信息素养文库)

ISBN 978-7-305-21052-5

I. ①C… II. ①田… III. ①C语言—程序设计—水平  
考试—自学参考资料 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 230772 号

出版发行 南京大学出版社  
社 址 南京市汉口路 22 号 邮 编 210093  
出 版 人 金鑫荣  
丛 书 名 信息素养文库  
书 名 C语言二级考试辅导教程  
编 著 田祥宏  
责任编辑 吕家慧 王南雁 编辑热线 025-83592655

照 排 南京理工大学资产经营有限公司  
印 刷 南京人民印刷厂有限责任公司  
开 本 787×1092 1/16 印张 9.75 字数 238 千  
版 次 2018 年 12 月第 1 版 2018 年 12 月第 1 次印刷  
ISBN 978-7-305-21052-5  
定 价 28.00 元

网 址: <http://www.njupco.com>  
官方微博: <http://weibo.com/njupco>  
微信服务号: njyuyexue  
销售咨询热线: (025)83594756

\* 版权所有, 侵权必究

\* 凡购买南大版图书, 如有印装质量问题, 请与所购  
图书销售部门联系调换

# 前 言

本书编者从事 C 语言教学近二十年,在全国计算机等级考试(二级 C 语言)、江苏省计算机等级考试(二级 C 语言)培训上经验丰富,在此基础上编写成本书。本书以 C 语言知识强化与编程能力训练为主要内容,介绍了 C 语言的基础知识、常见算法编程技巧与方法,以及 C 语言知识强化训练等。以 C 语言编程能力培养为主要目标,满足 C 语言进一步学习与教学的需求,重点突出应用性和技巧方法训练。按照“知识点+编程训练”的思想关联知识,以编程能力训练由简单到复杂为主线贯穿全书,将 C 语言基础知识和编程能力训练有机结合。全书共 12 章,内容包括 C 语言基础知识、控制结构、数组与字符串、函数、指针、结构体与共同体、文件、递进式编程训练题、常用算法编程、计算机等级考试题解析等内容。

本书可作为普通高校应用型本科学生或工科本/专科学生学习 C 语言程序设计的辅助教材及教学参考书,特别适合全国计算机等级考试(二级 C 语言)、江苏省计算机等级考试(二级 C 语言)培训教材或参考书。也可作为有关程序设计人员和自学者的参考书。

本书在编写过程中,得到了金陵科技学院计算机工程学院吕艳琳、常子楠老师的大力支持和帮助,在此表示感谢!由于编者水平有限,书中难免存在不妥之处,恳请广大读者批评指正。

编 者

2018 年 10 月

# 目 录

第1章 C语言基础知识 .....	1	第5章 函数 .....	42
1.1 数据类型与输入/输出 .....	1	5.1 函数概念 .....	42
1.2 常量、变量 .....	10	5.2 函数参数与返回值 .....	43
1.3 运算符 .....	16	5.3 变量的作用域 .....	44
第2章 控制结构 .....	19	5.4 存储类别 .....	45
2.1 顺序结构 .....	19	5.5 内部函数与外部函数 .....	47
2.2 选择结构 .....	19	5.6 函数嵌套调用/递归调用 .....	47
2.3 循环结构 .....	22	5.7 函数与指针 .....	48
第3章 数组与字符串 .....	26	5.8 main()函数的参数和返回值 .....	49
3.1 数组(顺序表) .....	26	第6章 结构体与共同体 .....	50
3.2 字符串处理 .....	32	6.1 结构体 .....	50
第4章 指针 .....	36	6.2 自定义类型 .....	52
4.1 指针概念 .....	36	6.3 枚举型 .....	52
4.2 用指针访问一维数组 .....	37	6.4 单链表 .....	53
4.3 用指针访问二维数组 .....	37	6.5 共同体类型 .....	58
4.4 用指针处理字符串 .....	38	第7章 文件 .....	60
4.5 二级指针与指针数组 .....	39	7.1 文件概念与分类 .....	60

7.2	文件的打开、关闭	60	10.10	数制转换	97
7.3	文件的操作函数	62	10.11	字符串的一般处理	98
7.4	文件检测与定位函数	62	10.12	穷举法	99
<b>第 8 章</b>	<b>典型程序函数</b>	<b>64</b>	10.13	递归算法	100
<b>第 9 章</b>	<b>递进式编程训练题目</b>	<b>82</b>	10.14	将数组元素逆置	100
9.1	基本输入/输出编程训练题		10.15	方程求根	101
	.....	82	<b>第 11 章</b>	<b>全国计算机等级考试试题解析</b>	
9.2	选择结构编程训练题	82		.....	108
9.3	循环结构编程训练题	83	11.1	选择题	108
9.4	综合应用编程训练题	84	11.2	基本操作题	116
<b>第 10 章</b>	<b>常见算法程序</b>	<b>85</b>	11.3	程序改错题	117
10.1	简单数值类算法	85	11.4	程序设计题	118
10.2	最大公约数、最小公倍数	86	<b>第 12 章</b>	<b>江苏省计算机等级考试试题解析</b>	
10.3	判断素数	88		.....	120
10.4	求最值	89	12.1	理论试题	120
10.5	排序问题	89	12.2	上机考试题	132
10.6	查找问题	92	<b>附录一</b>	<b>C 语言中几个常见小技巧</b>	<b>147</b>
10.7	插入法	93	<b>附录二</b>	<b>C 语言二级考试中必会的标准库函数</b>	<b>149</b>
10.8	矩阵(二维数组)运算	94		.....	
10.9	迭代法	96			

# 第 1 章

## C 语言基础知识

### 1.1 数据类型与输入/输出

#### 一、标识符

标识符命名规则如下:

- (1) 标识符只能由字母、数字和下划线组成。
- (2) 标识符的第一个字符必须是字母或下划线。
- (3) 标识符不能与 C 语言关键字相冲突。
- (4) 标识符的大小写字母是有区别的,如 abc 和 ABC 是两个完全不同的标识符。

标识符的分类:

关键字:关键字是具有特定的含义、专门用来说明 C 语言特定成分的一类标识符。

预定义标识符:在 C 语言中它也有特定的含义。预定义标识符可以用于“用户标识符”使用,只是这将使这些标识符失去了原来的意义。

用户标识符:由用户根据需要定义的标识符称为用户标识符。用户标识符用来给变量、函数、数组或者指针等命名。

#### 二、C 语言的数据类型

##### 1. C 语言基本数据类型

整型、浮点型、字符型、(枚举型)通常称为“三基”(三种基本类型)。

##### 2. 构造类型

数组、结构体、共同体、(自定义类型)。

##### 3. 指针类型

格式:类型说明符 \* 指针变量名。

##### 4. 空类型

格式:void

#### 三、C 语言数据类型涉及的关键字

int:用于定义普通整型变量、指针等,一个变量在 ANSI - C(标准 C)中占 2 个字节,在

VC++ 中占 4 个字节。

**long**: 用于字义长整型变量、指针等, 一个变量在 ANSI-C 中占 2 个字节, 在 VC++ 中占 4 个字节。

**short**: 用于字义短整型变量、指针等, 一个变量在 ANSI-C 中占 2 个字节, 在 VC++ 中占 2 个字节。

**unsigned**: 用于字义无符号整型变量等, 一个变量在 ANSI-C 中占 2 个字节, 在 VC++ 中占 4 个字节。有一个对应的关键字 **signed** 是用于定义有符号型整型, 但是用的很少, 主要原因是在缺省情况下, 整型定义的变量就是有符号型的, 比如: `int a;` 就是定义了一个有符号的整型变量 `a`, 在关键字 `int` 前面省略了 `signed`。

**float**: 用于定义浮点型(单精度型)变量、指针等, 一个变量在 ANSI-C 中占 4 个字节。

**double**: 用于定义浮点型(双精度型)变量、指针等, 一个变量在 ANSI-C 中占 8 个字节。

**long double**: 用于定义浮点型(长双精度型)变量、指针等, 一个变量在 ANSI-C 中占 16 个字节。

**char**: 用于定义字符型变量、指针等, 一个变量在 ANSI-C 中占 1 个字节。

#### 四、输入/输出

所谓输入/输出是以计算机为主体而言的。在 C 语言中, 不提供输入/输出语句, 输入和输出操作是由库函数来实现的。

在 C 标准函数库中提供了一些输入/输出函数, 例如: `printf` 函数和 `scanf` 函数。`printf`、`scanf` 不是 C 语言的关键字, 而只是函数名。C 语言提供的函数程序代码被保存在库文件(.obj 或 .lib)中, 它们不是 C 编译器负责编译的 C 语言成分。因此, 在使用 C 语言库函数时, 要用预编译命令“`#include`”将有关“头文件”包括到源文件中。使用标准输入/输出库函数时要用到“`stdio.h`”文件, 因此源文件开头应有以下预编译命令:

```
#include <stdio.h> 或 #include "stdio.h"
```

由于 `printf` 和 `scanf` 函数使用频繁, 系统允许直接使用这两个函数, 而不用预编译命令。

##### (一) 字符输入/输出

###### 1. 字符数据输出

字符数据输出使用的是 `putchar` 函数, 其作用是向显示设备(如显示器)输出一个字符, 其语法格式: `int putchar(int ch);`, 其中的 `ch` 是要进行输出的字符可以是字符型常量、变量或者整型变量, 如输出一个字符 `a`, 语句为: `putchar('a');`, 每次只能输出一个字符。

###### 2. 字符数据输入

字符数据输入使用的是 `getchar` 函数, 其作用是从控制台输入一个字符, 语法格式: `int getchar();` 例: `char c; c = getchar();` 就是从键盘输入一个字符, 然后将输入的字符赋值给字符变量 `c`。

##### (二) 字符串输入/输出

###### 1. 字符串输出函数

字符串输出使用的是 `puts` 函数, 作用是输出一个字符串到屏幕上, 语法格式: `int puts`

(char \* str);其中形式参数 str 是字符指针类型(或字符串名),可以用来接收要输出的字符串,例:puts("abcdefgh");在这里不用使用\n进行换行,puts函数会在字符串中判断结束符\0,自动进行换行。

**例 1.1 字符串的输出。**

```
char s1[80] = " I'm a", s2[80] = "Chinese boy!";
```

```
puts(s1); puts(s2);
```

输出结果为:

```
I'm a
```

```
Chinese boy! /* 每次 puts 函数在输出结束时会自动换行 */
```

## 2. 字符串输入函数

字符串输入使用的是 gets 函数,作用是将读取的字符串保存在形式参数 str 变量中,读取过程直到出现新的一行为止,其中新的一行的换行字符将会转换为字符串中的空终止符\0,语法格式:char \* gets(char \* str);其中的 str 字符串指针变量为形式参数。例:char s1[80]; gets(s1);

### (三) 格式输入/输出

#### 1. printf() 函数

用于向终端输出若干个任意类型的数据。

##### • 一般格式

```
printf(格式控制, 输出表列)
```

```
例:printf("%d, %c\n", i, c);
```

其中:

格式控制:是由" "括起来的字符串,由“格式说明”和“普通字符”组成。“格式说明”的作用是将输出的数据转换为指定的格式输出,并且总是由“%”字符开始,由%及格式字符组成。例如% d,% c等。“普通字符”将原样输出。例如上例“% d,% c\n”中的“,”及“\n”就是普通字符(可以称为原样输出部分)。再如,printf("c = % cf = % fs = % s", c, f, s);中的“c = ”“f = ”“s = ”均为普通字符。

输出表列:指需要输出的一些数据,可以是常量、变量或表达式。例如:

```
printf("a = % d \ b = % d", a, b);
```

输出结果:a = 3 \ b = 4

其中,普通字符“a = ”“\ (空格)”“b = ”为原样输出。3,4 则为变量 a,b 中的值(注意 3,4 无前导空格和尾随空格)。

##### • 格式字符

对于不同类型的数据输出,采用不同的格式字符。

(1) d 格式字符:用来输出带符号的十进制整数。

① % d,按整型数据的实际长度输出。

② % md,按指定的长度输出。如果数据位数小于 m,则左端补以空格,若大于 m,则按实际位数输出。例如:

```
printf("% 4d, % 4d", a, b);
```

若  $a = 123, b = 12345$

则输出结果为:  $\cup 123, 12345$

③ %ld, 输出长整型数据。例如:

```
long a = 123456;
```

```
printf("%ld", a);
```

则输出结果为: 123456

也可指定长整型输出宽度, 例如:

```
printf("%8ld", a);
```

则输出结果为:  $\cup 123456$

注意, 对于长整型数据输出应该采用 %ld 格式, 如果采用 %d 输出, 则会出错。

(2) o 格式字符: 以八进制形式输出整数。是一种无符号数。例如:

```
int a = -1;
```

```
printf("%d, %o", a, a);
```

输出结果为: -1, 177777

因为 int 型数 -1 在内存中以补码形式存放, 即存放形式为  $(1111111111111111)_2$ , 最高位为符号位 1, 表示负数。当以八进制无符号数形式输出时, 则变为 177777。

可以使用 "%lo" 输出长整型, 也可以使用 "%8o" 进行定长输出。

(3) x 格式字符: 以十六进制形式输出整数。是一种无符号数。例如:

```
int a = -1;
```

```
printf("%x, %d", a, a);
```

输出结果为: ffff, -1

输出结果中的 ffff 道理同上。也可以使用 "%lx" 输出长整型, 使用 "%8x" 进行定长输出。

(4) u 格式字符: 输出 unsigned 数据, 即无符号数。以十进制形式输出。

一个 int 型数据可以用 %u 格式输出; 反之, 一个 unsigned 型数据也可以用 %d 格式输出。按内存中实际存储形式相互赋值。

unsigned 型数据也可以按 %o, %x 格式输出。

**例 1.2** 无符号数的输出。

```
main()
```

```
{
```

```
    unsigned int a = 65535;
```

```
    int b = -2;
```

```
    printf("a = %d, %o, %x, %u\n", a, a, a, a);
```

```
    printf("b = %d, %o, %x, %u\n", b, b, b, b);
```

```
}
```

运行结果为:

```
a = -1, 177777, ffff, 65535
```

```
b = -2, 177776, fffe, 65534
```

因为 65535 为无符号数时, 最高位不是符号位, 而当以 %d 格式输出时, 是以有符号数

的格式输出的,即最高位为符号位,则输出为负数-1。

(5) c 格式字符:用于输出一个字符。例如:

```
char c = 'a';
```

```
printf("%c", c);
```

注意,整数也可以用字符形式输出;反之,字符数据也可以以整数形式输出。例如:

```
int i = 321;
```

```
printf("%c, %d\n", i, i);
```

输出结果为:A,321

输出字符为'A'是因为字符为无符号整数,仅能表达0~255的整数,则模为256,所以 $321 \% 256 = 65$ 为字符'A'的ASCII码。

也可以指定字符输出的宽度,例如`printf("%3c", c)`,此时输出的3列中,前2列补空格。

(6) s 格式字符:用于输出一个字符串。有五种用法:

① %s:例:`printf("%s", "CHINA");` 输出为:CHINA

② %ms:指定输出字符串的宽度,如字符串本身长度大于m,则突破限制;若串长小于m,则左补空格。

③ %-ms:在m列范围内,字符串向左靠,右补空格。

④ %m.ns:输出占m列,但只取待输出字符串中左端n个字符。这n个字符输出在m列的右侧,左补空格。

⑤ %-m.ns:含义同上,只是n个字符靠左对齐,右补空格。如果 $n > m$ ,则m自动取n值,即保证n个字符正常输出。例:

```
printf("%3s, %7.2s, %.4s, %-5.3s\n", "CHINA", "CHINA", "CHINA", "CHINA");
```

输出结果:CHINA, CH, CHIN, CHI

**注意:**%.4s中只给出了n,没给m,自动使 $m = n = 4$ 。

(7) f 格式符:输出实数(包括单、双精度)。

① %f:不指定输出宽度。整数部分全部输出,小数部分占6位。注意:输出的数字并非全部是有效数字。单、双精度实数的有效数字一般为7、16位。

**例 1.3** 输出单、双精度数时的有效位数。

```
main()
```

```
{
```

```
float x, y;
```

```
double a, b;
```

```
x = 123456.123; y = 654321.321;
```

```
a = 1234401190110.123456789;
```

```
b = 7531031008655.23456321;
```

```
printf("%f\n", x + y);
```

```
printf("%f\n", a + b);
```

```
}
```

运行结果为:

777777.437500

8765432198765.357420

由于单精度数前 7 位有效,双精度数前 16 位有效,小数部分均占 6 位,所以上述结果中超出有效位数的小数部分均存在无效数字,如.437500 中的 37500,.357420 中的 420 是无意义的。

② %m.nf:输出的总长度为 m 列,且包含 1 位小数点及 n 位小数。当数据位数小于 m 时,则左补空格。当数据位数多于 m 时,则整数部分按实际长度输出,小数部分按指定长度的 n 值输出。当没有指定小数部分位数 n 时,则小数部分默认为 6 位。

③ %-m.nf:含义同上,只是输出数值靠左对齐,当数据位数小于 m 时,右补空格。例如,当有 float f=123.456;时,则:

```
printf("%f %11f %10.2f %-10.2f %.2f %2.1f \n", f, f, f, f, f, f);
```

输出结果:123.456001 123.456001 123.46 123.46 123.46 123.5

(8) e 格式符:以指数形式输出实数。可用以下形式:

① %e:系统自动指定小数位数为 5 位,指数部分为 4 位(如 e+02),数值按规范化指数形式输出(即小数点前必须有且只有 1 位非零数字),因此 %e 输出正实数时,总位数为 11 位(含小数点 1 位及整数 1 位),输出负实数时,总位数为 12 位(多出 1 位符号位)。例如,printf("%e %e", 123.96, -123.95452166);

输出结果:1.23960e+02 -1.23955e+02

② %m.ne 及 %-m.ne:m、n 及 - 的含义与前相同。此处的 n 是指指数形式中的尾数位数且含小数点 1 位。例如,若 f=123.96,则:

```
printf("%15e %10.2e %-10.2e %.2e %7.1e", f, f, f, f, f);
```

输出结果:1.23960e+02 1.2e+02 1.2e+02 1.2e+02 1e+02

从结果 1e+02 中看出,当尾数 n=1 时,输出时没有尾数显示。

(9) g 格式符:用来输出实数,根据数值的大小,自动选 f 格式或 e 格式,选择占位较少的一种,且不输出无意义的 0。此格式使用较少。例如,若 f=123.468,则:

```
printf("%f %e %g", f, f, f);
```

输出结果:123.468002 1.23468e+02 123.468

从结果中可以看出 %g 选择了 %f 格式,且取消了末尾的无意义的 002。

以上介绍的格式符,归纳见表 1.1。

表 1.1 printf 函数的格式字符

格式字符	说 明
d,i	输出带符号的十进制整数(正数不带符号)
u	输出无符号的十进制整数
o	输出无符号的八进制整数(不输出前缀 0)
x,X	输出无符号的十六进制整数(不输出前缀 0x),用 x 则输出十六进制数 a~f 时以小写形式输出,用 X 时,则以大写形式输出

续表

格式字符	说 明
c	以字符形式输出单个字符
s	输出字符串。与其对应的输出项应为以"\0"结尾的字符数组名、字符串常量或指向字符串的指针变量名
f	以小数形式输出单、双精度实数,隐含输出6位小数
e,E	以规范化指数形式输出单、双精度实数。用e时指数以“e”表示(如1.2e+02),用E时指数以“E”表示(如1.2E+02)
g,G	选用%f或%e格式中输出宽度较短的一种,不输出无意义的0。用G时,若以指数形式输出,则指数以大写表示
p	输出变量或数组的地址

在格式说明中,在%和上述格式字符间可以插入以下几种附加符号(或修饰符),见表1.2。

表 1.2 printf 函数的格式修饰符

字符	说 明
L(字母)	输出长整型数据(可用%ld,%lu,%lo,%lx)以及double型数据(用%lf或%le)
m(一个正整数)	指定输出数据的最小宽度。当实际数据宽度>m时,以实际宽度为准
n(一个正整数)	对实数,表示输出n位小数;对字符串,表示截取的字符个数
-	输出的数字或字符在域内向左靠
+	输出的结果总是带有“+”号或“-”号
0	当域宽m>实际数据长度时,不足数位以0补足

**注意:**修饰符可以多个一起使用,例如:

```
printf("%+08d\n",2346);
```

输出结果为: +0002346

可见,三个修饰符“+”“0”“8(域宽)”一起使用,使输出数据带有+号,且总宽度为8,不足数位补0。

## 2. scanf() 函数

又称作格式输入函数,其作用是按用户指定的格式从键盘输入任意类型的数据到指定的变量之中。通常用来输入整数和浮点数。它是一个标准库函数,它的函数原型在头文件“stdio.h”中。

### • 一般形式

```
scanf(格式控制,地址表列);
```

其中:

格式控制:用于控制输入数据的类型、个数、间隔符等,是由" "括起来的字符串,由“格式说明”和“普通字符”组成。“格式说明”的作用是将输入的数据转换为指定的格式输入,

总是由“%”字符开始,并由%及格式字符组成,例如% d,% f等。而“普通字符”则必须原样输入,例如在 scanf("a=%d",&a);语句中,"a=%d"为格式控制,其中的"a="为普通字符,在从键盘输入数据时必须原样输入。

地址表列:由若干个地址组成的表列,可以是变量的地址,或字符串的首地址。例如“scanf("%d%d%d",&a,&b,&c);”,其中&为取地址运算符,&a是指取出变量a在内存中的地址,该地址将作为从键盘输入数据存放的内存地址。而变量a、b、c的地址是在编译连接阶段分配的。

**注意:**在输入数据之间一般以空格分隔,也可以用回车键(Enter)或跳格键 Tab。

例如,对于“scanf("%d%d%d",&a,&b,&c);”来说:

(1) 3 4 5

(2) 3

4 5

(3) 3(按 Tab 键) 4

5

以上三种方式均正确。

**注意:**3,4,5方式不允许。输入数据时,不能加入多余的普通字符。如果上面的语句改为“scanf("%d,%d,%d",&a,&b,&c);”,则输入数据时,必须采用加入“,”的方式,即只有3,4,5的方式才是被允许的。

#### • 格式说明

以%开头,以一个格式符为结束,中间可以插入格式修饰符,如l、h、\*等。

表 1.3 scanf 格式字符

格式字符	说 明
d,i	用来输入有符号的十进制整数
u	用来输入无符号的十进制整数
o	用来输入无符号的八进制整数。键入数据时不能出现8及以上数字,否则出错。键入的数据可不必加前缀0
x,X	用来输入无符号的十六进制整数(大小写作用相同),键入的数据可不必加前缀0x
c	用来输入单个字符
s	用来输入字符串,将字符串送到一个字符数组中,在输入时以非空格字符开始,以第一个空格字符结束。字符串末尾自动添加'\0'作为串结束标志
f	用来输入实数,可以用小数形式或指数形式输入
e,E,g,G	与f作用相同,e与f,g可以互相替换(大小写作用相同)

表 1.4 scanf 的格式修饰字符

字符	说 明
l	用于输入长整型数据(可用%ld,%lo,%lx)以及 double 型数据(用%lf或%le)
h	用于输入短整型数据(可用%hd,%ho,%hx)
域宽	指定输入数据所占宽度(列数),系统自动截取所需数据。域宽应为正整数
*	表示本输入项在读入后不赋给相应的变量,即跳过该输入值。可称禁止赋值符

**说明:**

(1) 对 unsigned 型变量所需数据,可用%u(无符号十进制),%d(有符号十进制),%o(八进制),%x(十六进制)格式输入。

(2) 可指定输入数据所占列数,系统自动按它截取所需数据。例如:

```
scanf("%3d %3u", &a, &b);
```

输入 123456,则 123 赋值给 a,456 赋值给 b。

此方法也可用于字符型:

```
scanf("%3c", &ch);
```

输入:abc(不加"),由于字符型变量 ch 只能容纳一个字符,系统把第 1 个字符 'a' 赋给 ch。

(3) 如果使用禁止赋值符 "\*",则表示跳过它指定的列数。例如:

```
scanf("%2d %*3d %2d", &a, &b);
```

输入信息如下:

```
123456789
```

则 12 赋值给 a,%\*3d 表示读入 3 位整数但不赋给任何变量,也即跳过 345 不用,67 赋值给 b。

(4) 输入实数只能用%f,此时不能指定精度。例如:

```
scanf("%7.2f", &a);
```

是不合法。

(5) 输入字符串时只能用%s,且以空格符作为输入结束的标志。例:

```
char a[20], b;
```

```
scanf("%s %c", a, &b);
```

若使字符数组 a 中存放字符串"1234",字符变量 b 中存放 '5',则键入内容为:

```
1234 5
```

可见,在输入字符串后要有一个空格表示字符串输入结束,并且在 scanf 中的 "%s %c" 之间也要有空格。

**3. 使用 scanf 函数时应注意的问题**

(1) scanf 函数中的“地址列表”中的变量名前的 &(地址运算符)不能忘记。例如:scanf("%d,%f", a, f);中变量 a, f 前未加 &,所以会出错。

(2) 若“格式控制”字符串中除了格式说明以外,还有其他字符,则在输入数据时应输

人与这些字符相同的字符。例如:

```
scanf("%d, %d", &a, &b);
应输入: 3,4
```

**注意:**因为在“格式控制”中出现“,”，所以键入数据时必须给出“,”作为分隔符，且不能用空格代替。例:

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

输入时两个数据间至少应该有一个空格分隔。例如:

```
1 2 或 10 35
```

以上方法是为了输入时,输入的数据较为易于理解,直观,不易发生输入错误。

(3) 用“%c”格式输入字符时,空格字符和“转义字符”都作为有效字符输入。例:

```
scanf("%c %c %c", &c1, &c2, &c3);
```

输入:a b c (从键盘输入时,无须“\”)

**注意:**'a' 字符送入 c1 中,' ' 空格字符送入 c2 中,'b' 则送入 c3。因为%c 只能接收一个字符,所以%c %c 之间无须空格。

(4) 在输入数据时,遇以下情况时认为一个数据输入结束:

- ① 遇空格,或按“回车”或“跳格”(Tab)键。
- ② 按指定的宽度结束,如“%3d”,只取3列。
- ③ 遇非法输入。

例如:scanf("%d %c %f", &a, &b, &c);

输入:1234a1230.26 (注意输入的各种类型数据之间是无须空格)

说明:1234 之后为一个字符 a,则 1234 遇非法输入 a 时会自动停止赋值。同理,123 后为字母 O,认为遇非法输入,则 O 后的小数省略。

将 1234 赋值给 a,'a' 字符赋值给 b,123 赋值给 c。

```
printf("%d %c %f\n", a, b, c);
```

输出结果:1234 a 123.000000

## 1.2 常量、变量

### 一、整型

常见的整型常量(十进制):345 -456 2344 ..... (普通整型常量、短整型)

无符号整型常量:735U 878u

长整型常量:78998L 2351

8 进制整型常量:034 067 0234 (0 开头整型常量)

16 进制整型常量:0xFA 0x57B (0x 开头整型常量)

定义变量格式:类型 变量名列表,例如:

```
int i_sum, i;    i_sum = 5435;    scanf("%d %d", &i_sum, &i); /* 输入时两个整数之间用空格或回车分隔 */
```

scanf() 格式控制串包括:格式控制项(%开头),默认分隔符是空格或回车,例:

```
printf("djdf; aslkda %d", <表达式列表>);
```

printf() 格式控制串包括:原样输出部分 格式控制项(%开头),例:

```
long k, j;    scanf("%ld, %ld", &k, &j);    printf("k = %ld, j = %ld", k, j);
```

整型变量占2个字节的存储空间,长整型变量占4个字节的存储空间,例:

```
unsigned k;    long m, n;    short p, q;
```

k, p, q 占2个字节的存储空间, m, n 占4个字节的存储空间。

## 二、浮点型(实型)

### (1) 常量

78.5 90.32 34.4

1.234E5 表示  $1.234 * 10^5$

### (2) 变量

float 变量名列表;例如: float f\_x, f\_y; (每个变量占4个字节的内存空间)

double 变量名列表;(每个变量占8个字节 Bytes 内存空间)

long double 变量名列表;(每个变量占16个字节的内存空间)

**注意:**在标准C中(ANSI-C),实型常量在内存中占8个字节的空间。

```
float x, y; scanf("%f, %f", &x, &y); /* 输入时,两个实数中间用,做分隔 */
```

```
printf("sum is %f", x + y);
```

```
double x, y; scanf("%f, %f", &x, &y);
```

```
printf("sum is %f", x + y);
```

```
long double x, y; scanf("%lf, %lf", &x, &y);
```

```
printf("sum is %lf", x + y);
```

## 三、字符型

### 1. 常量

用单引号括起来的。例如:'a' 'A' '7' '0' - " + " !.....

转义字符:

```
'\n' '\r' '\t' '\a' '\b' '\f'
```

'\ddd' 是由1~3位八进制组成的数对应的ASCII码表中的字符,如:'\025'\067'\040'\0'.....

'\xhh' 是由1~2位十六进制组成的数对应的ASCII码表中的字符,如:'\x4A'

全部转义字符见表1.5。