

配合教育部职业教育与成人教育司推荐教材
五年制高等职业教育计算机专业课适用

崔武子 编著

C语言习题

与 考级指导

华夏出版社

谭浩强 主编

C 语言习题与考级指导

谭浩强 主编

崔武子 编著

华夏出版社

图书在版编目(CIP)数据

C语言习题与考级指导/谭浩强主编. - 北京:华夏出版社,2005.11

ISBN 7-5080-3841-X

I. C... II. 谭... III. C语言-程序设计-自学参考资料

IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 124740 号

C语言习题与考级指导

主 编:谭浩强

编 著:崔武子

责任编辑:焦 玉

封面设计:刘 颖

出版发行:华夏出版社

(北京市东直门外香河园北里4号 邮编:100028)

经 销:新华书店

印 刷:北京人卫印刷厂

版 次:2005年11月北京第1版

2006年1月北京第1次印刷

开 本:787×1092 1/16开

印 张:9.25

字 数:213千字

定 价:15.00元

本版图书凡印刷、装订错误,可及时向我社发行部调换

前 言

C 语言是目前最为流行的程序设计语言，全国各地组织的计算机统一考试中也都包括《C 语言程序设计》科目。由于 C 语言语法规则繁多，使用灵活，很多初学者感到有一定的难度。为使读者能够尽快入门和熟知 C 语言中的知识，以及更有效地提高计算机统一考试的应试能力和实际编程能力，特编此书。希望读者合理使用本书，及时得到训练，达到预期学习目标。

本书是和崔武子等编著的《C 语言程序设计（基础版）》（华夏出版社 2005 年 6 月出版）配合使用的辅助教材，全书共分 7 章，每章的内容顺序与主教材一一对应。各章分为习题答案、考级指导和训练题。

1. 习题答案

在各章的习题答案一节中给出教材中所有习题的参考答案。为了使读者方便分析，对每一个习题提供参考程序的同时，还给出运行结果。需要说明的是，解决一个问题，可能有多种方法，但本书只提供一种答案。读者做习题中的程序时，不要一上来直接照搬本书中的答案，而要独立试着编写自己的程序，然后与本书中的答案做比较，并找出程序的效率、可读性等方面各自的优缺点，这样才能有更深入的心得。

2. 考级指导

在各章的考级指导一节中，通过补充知识和典型例题分析这两个小节，为等级考试者进行考前补充辅导。教材中为了突出重点，也考虑到各学校的课时限制，将初学者不常使用的内容省略，而对于 C 语言的重点内容用通俗易懂的方式较详细地进行介绍。由于等级考试考核的知识点较多，因此在本节中做了较全面的补充。全书的内容、难度和形式与最近几年的全国计算机等级考试（二级）相类似。本内容也有助于想要提高 C 语言程序设计水平的读者。

3. 训练题

各章中的训练题均包括选择题、填空题和参考答案，且从第 5 章起增设了

改错题和编程题，以适合等级考试的要求。要学好一种程序设计语言，除了听课、看书以外，必须还要进行大量的实践。本部分的内容将会对读者的心身实践带来很大的益处。

本书中的所有程序均在 Turbo C2.0 中调式通过。

本书不仅可以作为《C 语言程序设计（基础版）》的最佳辅助教材，还可作为等级考试的参考资料。

本书由崔武子主持编写并统稿，孙力红、鞠慧敏、赵重敏参加了部分内容的编写。

限于作者水平，书中难免有错，恳请读者指正。

作者

2005 年 7 月

目 录

第 1 章 C 语言的基础知识	1
1.1 习题答案	1
1.2 考级指导	3
1.2.1 补充知识	3
1.2.2 典型例题分析	7
1.3 训练题	8
1.3.1 选择题	8
1.3.2 填空题	9
1.3.3 参考答案	10
第 2 章 程序设计的 3 种基本结构	11
2.1 习题答案	11
2.2 考级指导	18
2.2.1 补充知识	18
2.2.2 典型例题分析	20
2.3 训练题	24
2.3.1 选择题	24
2.3.2 填空题	31
2.3.3 参考答案	36
第 3 章 使用数组处理大批量数据	38
3.1 习题答案	38
3.2 考级指导	43
3.2.1 补充知识	43
3.2.2 典型例题分析	46
3.3 训练题	48
3.3.1 选择题	48
3.3.2 填空题	50
3.3.3 参考答案	52
第 4 章 指针的使用	53
4.1 习题答案	53
4.2 考级指导	58
4.2.1 补充知识	58
4.2.2 典型例题分析	60
4.3 训练题	63
4.3.1 选择题	63
4.3.2 填空题	65
4.3.3 参考答案	67

第 5 章 自定义函数	68
5.1 习题答案	68
5.2 考级指导	74
5.2.1 补充知识	74
5.2.2 典型例题分析	78
5.3 训练题	83
5.3.1 选择题	83
5.3.2 填空题	88
5.3.3 改错题	91
5.3.4 编程题	94
5.3.5 参考答案	98
第 6 章 结构体	102
6.1 习题答案	102
6.2 考级指导	108
6.2.1 补充知识	108
6.2.2 典型例题分析	113
6.3 训练题	115
6.3.1 选择题	115
6.3.2 填空题	118
6.3.3 改错题	120
6.3.4 编程题	122
6.3.5 参考答案	125
第 7 章 文本文件中数据的读写	128
7.1 习题答案	128
7.2 考级指导	131
7.2.1 补充知识	131
7.2.2 典型例题分析	132
7.3 训练题	133
7.3.1 选择题	133
7.3.2 填空题	134
7.3.3 改错题	134
7.3.4 编程题	135
7.3.5 参考答案	136
附录	137
附录 I C 语言关键字	137
附录 II 常用字符与 ASCII 代码对照表	137
附录 III 常用运算符的优先级和结合方向	138
附录 IV 常用 C 库函数	139
附录 V 常见错误、警告信息表	141

第1章 C语言的基础知识

1.1 习题答案

1. 编写程序，输入圆柱体的底面半径和高，计算其体积。

程序代码：

```
#include <stdio.h>

main()
{   float r=0,h=0,v=0;

    printf("Input r and h:");
    scanf("%f%f",&r,&h);
    v=3.14*r*r*h;
    printf("r=%f,h=%f,v=%f\n",r,h,v);
}
```

运行结果：

Input r and h: 5.6.5<回车> 注：本书中用户输入的部分用下划线标出
r=5.000000,h=6.500000,v=510.250000

2. 编写程序，输入一个点的坐标值(x,y)，计算该点与原点的距离。

程序代码：

```
#include <stdio.h>
#include <math.h>

main()
{   float x=0,y=0,s=0;

    printf("Input x and y:");
    scanf("%f%f",&x,&y);
    s=sqrt(x*x+y*y);
    printf("(0,0)--(%f,%f):%f\n",x,y,s);
}
```

运行结果：

Input x and y: 5.5 6.8<回车>
(0,0)--(5.500000,6.800000):8.745856

3. 编写程序，计算表达式 $y = \left| \frac{5x+1}{x^2+1} \right|$ 的值，其中 x 为整数，从键盘输入。

程序代码:

```
#include <stdio.h>
#include <math.h>

main()
{   int x=0;
    float y=0;

    printf("Input x:");
    scanf("%d",&x);
    y=fabs((float)(5*x+1)/(x*x+1));
    printf("x=%d,y=%f\n",x,y);
}
```

运行结果:

Input x:10<回车>

x=10,y=0.504951

4. 编写程序, 随机产生一道 3 位数的加法题, 用户输入答案后, 显示正确的答案并暂停, 按任意键后再产生一道 3 位数的减法题。

程序代码:

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

main()
{   int a=0,b=0,c=0;

    randomize();
    a=random(900)+100;
    b=random(900)+100;

    printf("%d+%d=?",a,b);
    scanf("%d",&c);
    c=a+b;
    printf("The answer:=%d\n",c);
    getch();

    a=random(900)+100;
    b=random(900)+100;
```

```

printf("%d-%d=?",a,b);
scanf("%d",&c);
c=a-b;
printf("The answer:=%d\n",c);
getch();
}

```

运行结果:

148+468=?616<回车>

The answer: =616

231-871=?640<回车>

The answer: =-640

4. 编写程序，在屏幕上显示“a b c d e”后暂停，按任意键时，在下一行立刻显示“b c d e f”后又暂停，再按任意键时，结束程序。

程序代码:

```

#include <stdio.h>
main()
{   char ch='a';

    printf("%c %c %c %c %c\n",ch,ch+1,ch+2,ch+3,ch+4);
    getch();

    printf("%c %c %c %c %c\n",ch+1,ch+2,ch+3,ch+4,ch+5);
    getch();
}

```

运行结果:

a b c d e

b c d e f

1.2 考级指导

1.2.1 补充知识

1. 数据类型

在 C 语言中，可以用 `signed`（有符号型）和 `unsigned`（无符号型）进一步确定一个整型变量的性质。其中用 `signed` 定义的称为有符号类型，在实际中可以省略不写 `signed`。

用 `unsigned` 定义的变量称为无符号型变量。例如：

```

int num;           /* 定义 num 为有符号整型变量 */
unsigned year;    /* 定义 year 为无符号整型变量 */

```

通过定义,对于有符号 int 类型变量 num,系统除为它在内存中分配两个字节的存储空间(即 16 个二进制位)外,还规定其最高位用于表示符号,此位若为 0,表示存放的值为正数;若为 1,值为负数。因此,它所能表示的最小值是 $(1000000000000000)_2$,即 $-32768(-2^{15})$;最大值是 $(0111111111111111)_2$,即 $32767(2^{15}-1)$ 。声明为无符号基本整型的变量 year,在内存中也占两个字节,但它没有符号位,其最高位也用来表示数值,因此它所能表示的最小值是 $(0000000000000000)_2$,即 0;最大值是 $(1111111111111111)_2$,即 $65535(2^{16}-1)$ 。

C 语言中还规定,若要表示一个无符号整数,应在数字的末尾加字母 u 或 U,如 30u, 128U 等。注意,无符号整数不能表示小于 0 的数,如 $-10u$ 是不合法的。对于无符号长整型数,则应在数字之后加 lu 或 LU,如 30lu, 128LU。

表 1.1 给出 C 语言允许定义的 6 种整型变量及其数值范围。

表 1.1 ANSI 标准整数类型

类型名	含义	占用字节数	数值范围
[signed] int	有符号基本整型	2	-32768~32767
[signed] short [int]	有符号短整型	2	-32768~32767
[signed] long [int]	有符号长整型	4	-2147483648~2147483647
unsigned [int]	无符号基本整型	2	0~65535
unsigned short [int]	无符号短整型	2	0~65535
unsigned long [int]	无符号长整型	4	0~4294967295

2. 条件运算符和表达式

运算符“?:”称为条件运算符,在 C 语言中它是惟一由 3 个运算对象组成的运算符。由条件运算符构成的表达式称为条件表达式。

条件表达式的一般形式:

表达式 1 ? 表达式 2 : 表达式 3

条件表达式的运算规则:

如果表达式 1 的值为“真”,取表达式 2 的值为运算结果;如果表达式 1 的值为“假”,取表达式 3 的值为运算结果。条件运算符的优先级高于赋值运算,但低于算术、关系和逻辑运算。

例如,使用条件运算符判断变量 a 和 b 的值,要求输出其中的较大值。程序如下:

```
main()
{
    int a,b;
    printf("Enter a & b:");
    scanf("%d%d",&a,&b);
    printf("max=%d\n",a>=b ? a : b);
}
```

程序中 printf 语句的输出项“a>=b ? a : b”即为条件表达式。程序运行时,若输入的 a

值为 10, b 值为 5, 由于“a>=b”的判断结果为“真”, 根据运算规则, 应取 a 值为表达式的值, 此时输出结果为“max=10”; 若输入的 a 值为 5, b 值为 10, “a>=b”的判断结果为“假”, 则取 b 值做为表达式的值, 输出结果也是“max=10”。

3. 位运算符

位运算是针对数据的二进制位进行的运算。C 语言提供了 6 种位运算符, 其基本功能如表 1.2 所示。

表 1.2 位运算简介

运算符	含义	运算规则	优先级
~	求反	0 变 1, 1 变 0	1 (高)
<<	左移	低位移入 0, 高位移出舍弃	2
>>	右移	高位按符号位扩展, 低位移出舍弃	2
&	按位与	两个对应位都为 1 时结果为 1, 否则为 0	3
^	按位异或	两个对应位相同时为 0, 相异为 1	4
	按位或	两个对应位有一个为 1 时结果即为 1	5 (低)

上述 6 种运算符中, 除求反 (单目运算) 运算符外, 都可以与赋值运算符结合组成扩展赋值运算符。

例如, 已有如下定义和语句:

```
int a,x;
a=9;
x=a<<3;    /* 将 a 的值左移 3 位后赋给 x */
printf("%d\n",x);
```

其中表达式“a<<3”的含义是将 a 的二进制值“0000000000001001”左移 3 位, 即将该二进制值的高端去掉 3 位, 同时在其低端补 3 个 0, 构成一个新二进制值“0000000001001000”, 然后再将该值转成十进制值赋给 x 输出, 此题的运行结果为 72。

从运行结果可以看出, 当左移 n 位时, a 的值为原值的 2^n 倍。同理, 当右移 n 位时, a 的值应为原值的 $1/2^n$ 倍。

再如, 已有如下定义和语句:

```
char a=5,b=11,x;
x=a&b;    /* 按位与运算 */
printf("%d\n",x);
```

其中表达式“a & b”的含义是将 a 的二进制值“00000101”与 b 的二进制值“00001011”从低向高按位进行“与”运算。“与”运算的规则: 当两个二进制位都为 1 时, 结果为 1; 否则为 0。因此表达式“a & b”的运算结果为“00000001”, 然后再将该值转成十进制赋给 x 输出, 此题的运行结果为 1。

注意: 由于 C 语言中的数值常量只有八进制、十进制和十六进制等表示形式, 因此, 在分析位运算结果时, 首先应将参与运算的数据转换为二进制数, 最后再将运算结果按原有进制输出。

对负数进行位运算时应注意：负数表示成二进制原码时，符号位（最高位）为 1。

4. 自增自减运算符++, --

运算符++, --的作用是使变量的值增 1 或减 1，它们既可以放在变量之前，也可以放在变量之后，无论运算符在变量之前还是在变量之后，变量的变化结果都是一样的。

运算符在变量之前，称为前缀运算符，如：

```
int i=0,j=0;
++i;
--j;
```

运算符在变量之后，称为后缀运算符，如：

```
int i=0,j=0;
i++;
j--;
```

以上两例的运算结果是一样的，变量 i 值增 1 后为 1，变量 j 值减 1 后为 -1。

使用自增自减运算符时应该注意，由于运算本身含有赋值功能，如 i++ 相当于 i=i+1，j-- 相当于 j=j-1，因此要求运算对象必须是变量。常量和表达式由于不能被赋值，因此 3++ 和 (a+b)++ 都是不合法的运算式。

++i 或 i++ 等作为表达式可以参与运算，如：

```
int i=1,j=0;
j=(i++)+3;
```

执行语句“j=(i++)+3;”时，首先计算表达式 i++ 的值，然后将该表达式的值与 3 的和赋给 j。C 语言规定，表达式 i++ 的值是对 i 增 1 之前的 i 值，所以 j 得到 4。如果将该语句改为“j=(++i)+3;”，则表达式 ++i 的值是对 i 增 1 之后的 i 值，所以 j 得到 5。

5. 逗号运算符

在 C 程序中“,”也是运算符，其作用是将各表达式连接起来构成逗号表达式。形式如下：

逗号表达式的一般形式：

表达式 1, 表达式 2, …… , 表达式 n

逗号表达式的求解过程：

先计算表达式 1，再计算表达式 2，……，最后计算表达式 n，表达式 n 的值即为逗号表达式的值。

例如，表达式 i=0,i++,i+2 就是一个含有 3 个表达式的逗号表达式，按上述求解规则，应先执行赋值运算式“i=0”，使变量 i 值为 0；再执行“i++”，使 i 值增至 1；最后计算“i+2”，此式值为 3，3 就是该逗号表达式的值。注意：由于在 i+2 的运算中没有给变量 i 赋值，运算结束后 i 的值仍为 1。

逗号运算符在所有运算符中优先级别最低，且具有从左到右的结合性。

1.2.2 典型例题分析

例 1.1 下面叙述错误的是

- A) 在 C 语言中, 各种括号应成对出现
- B) 在 C 语言中, 赋值号的左边不可以出现算术表达式
- C) 在 C 语言中, 区分标识符的大小写
- D) 在 C 语言中, 未经赋值的变量自动被初始化为空

解:

在 C 程序中常常含有一些具有特定意义的符号, 如大括号“{”和“}”用于标识函数体或复合语句体, 符号“/*”和“*/”则做为注释的定界符, 还有函数名后的一对圆括号“(”和“)”, C 语言规定需成对使用它们。

赋值操作就是将运算符右边表达式的值存放到左边以变量名为标识的存储单元中去, 因此赋值运算符左侧不允许是表达式, 因为表达式不能代表任何存储单元。

C 语言规定, 程序中使用的符号, 包括变量名在内, 严格区分大小写英文字母, 如 DATA, data 和 Data 代表的是不同标识符。

根据以上分析, 选项 A, B 和 C 中的叙述都是正确的。在 C 程序中, 如果使用一个未赋初值的变量, 会得到一个不确定值。

答案: D

例 1.2 以下合法的用户标识符是

- A) int
- B) a-5
- C) 3x
- D) student

解:

在 C 语言中, 允许用户自行定义标识符, 如给一个变量命名。同时 C 语言还对标识符的合法性做了严格的规定。本题中, 选项 B 和 C 因违反“标识符可含字母、数字和下划线, 但必须以字母或下划线开头”这一规定, 因此是错误的。选项 A 使用 C 语言的关键字作为用户标识符, 也是错误的。

答案: D

例 1.3 在 C 程序中, 非法的实型常量是

- A) 2E1.2
- B) .1234
- C) 1e3
- D) 2.5e-5

解:

在 C 语言中, 实型常量可以十进制小数和指数两种形式表示。小数形式应由数字和小数点组成, 且必须要有小数点。指数形式则以字母 E (或 e) 代表以 10 为底的指数, E 的左边是数值部分 (有效数字), 可以表示成整数或小数形式; E 的右边是指数部分, 只允许是整数形式。因此选项 B, C 和 D 是正确的表现形式。

答案: A

例 1.4 有以下程序:

```
main()
{
    int x=1,y=10,z=100;
    printf("%d\n", (x++, y+=z, x+y));
}
```


题 1.5 在 C 语言中，合法的常量是

- A) '55' B) .5E+2 C) 55*10 D) u23

题 1.6 下列选项中，优先级最低的运算符是

- A) = B) /= C) + D) ,

题 1.7 C 程序中，运算对象必须为整型数据的运算符是

- A) * B) % C) -- D) /

题 1.8 C 程序中，判断“a 是否在 5 和 10 之内或 a 超过 15”的正确表达式为

- A) a>5 || a<10 || a>15 B) a>5 && a<10 && a>15
C) a>5 && a<10 || a>15 D) a>5 || a<10 && a>15

题 1.9 表达式 1/2*100+5%4+5.0/10 的值是

- A) 51.0 B) 1.5 C) 51.5 D) 1.0

题 1.10 将数学式 $\frac{ab}{c+b}$ 改写为 C 语言表达式，正确的是

- A) ab/(c+b) B) ab/c+b
C) a*b/(c+b) D) a*b/c+b

题 1.11 有以下程序：

```
main()
{
    int a=5;
    printf("%d\n",(++a*7%10));
}
```

程序运行后的输出结果是

- A) 2 B) 3 C) 4 D) 5

题 1.12 有以下程序：

```
main()
{
    int a=30;
    float b=20.5;
    b=a+(int)b;
    printf("%f\n",b);
}
```

程序运行后的输出结果是

- A) 50 B) 50.000000 C) 50.5 D) 50.500000

1.3.2 填空题

题 1.13 若有定义“int a=5,b=6;”，则表达式 (a>b || a+b<10) || a 的计算结果为【1】。

题 1.14 假设变量 x 的值为 135.79，则表达式(int)(x*10+0.5)/10.0 的值是【1】。

题 1.15 C 语言中，关系运算符有【1】，逻辑运算符有【2】。

1.3.3 参考答案

答 1.1 A

答 1.2 C

答 1.3 C

答 1.4 A

答 1.5 B

答 1.6 D

答 1.7 B

答 1.8 C

答 1.9 B

答 1.10 C

答 1.11 A

答 1.12 B

答 1.13 **【1】** 1

答 1.14 **【1】** 135.8

答 1.15 **【1】** <、<=、>、>=、==、!= **【2】** &&、||、!