“十四五”职业教育国家规划教材

扫描二维码
共享立体资源

C语言程序设计

第二版

主　编　张小梅

C语言程序设计（第二版）
主　编　张小梅
北京出版集团
北京出版社

北京出版集团
北京出版社

图书在版编目（CIP）数据

C 语言程序设计 / 张小梅主编. —2 版. —北京：
北京出版社，2020.9（2023 重印）
高职十二五规划教材 ：2014 版
ISBN 978-7-200-15907-3

Ⅰ. ①C… Ⅱ. ①张… Ⅲ. ①C 语言—程序设计—高等
职业教育—教材 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2020）第 197341 号

C 语言程序设计（第二版）
C YUYAN CHENGXU SHEJI（DI-ER BAN）

主　　编：张小梅
出　　版：北京出版集团
　　北 京 出 版 社
地　　址：北京北三环中路 6 号
邮　　编：100120
网　　址：www.bph.com.cn
总 发 行：北京出版集团
经　　销：新华书店
印　　刷：定州启航印刷有限公司
版 印 次：2020 年 9 月第 2 版　2023 年 7 月修订　2023 年 7 月第 2 次印刷
成品尺寸：185 毫米 ×260 毫米
印　　张：15
字　　数：321 千字
书　　号：ISBN 978-7-200-15907-3
定　　价：45.00 元

教材意见建议接收方式：010-58572341　邮箱：jiaocai@bphg.com.cn
如有印装质量问题，由本社负责调换
质量监督电话：010-82685218　010-58572341　010-58572393

目 录

单元二

C 程序设计基础

2019 年 8 月，由华为自主研发的着眼于 5G 物联网时代的新一代操作系统——鸿蒙（HarmonyOS）公布于世。在“科技兴则民族兴，科技强则国家强”的今天，鸿蒙的诞生着实让国人为之振奋！它全面开源，捏沙成团，对中国的软件业无疑是一条振兴之路。

单元概述

你知道鸿蒙操作系统的内核是由什么语言开发的吗？是以 C 语言为基础开发的。

古语有云，“千里之行，始于足下，不积跬步，无以至千里”。无论多庞大的程序或系统，都是由基本的指令和数据组成。本单元，将带你学习 C 语言的基本编程元素，它是学好 C 程序设计的基石。

任务 1　C 的数据类型

任务目标

1. 能正确描述数据类型的概念；
2. 能为程序中的变量定义合理的数据类型；
3. 能通过学习“数据类型”，感受“凡事有度，过犹不及”的意义，做一个做事“有度”的人。

任务描述

分析下列问题中各用到了哪些数据？应该定义成什么类型比较合适？

（1）输入一个年份 year，判断它是否为闰年。

（2）输入圆的半径 r，计算它的面积 area。

（3）输入一个大写字母存入 upper，将它转换成对应的小写字母存入变量 lower 并输出。

预备知识

计算机处理数据时，必须明确它是什么类型，以便为其分配合适大小的存储空间，并按照相应的规则进行操作。因此，在编写程序时，首先要对数据进行类型说明。

数据类型定义了一个运算集，即对特定的一类数据定义了特定运算。如数值型数据可进行算术运算，字符型数据可进行连接运算，而逻辑型数据可进行逻辑运算等。C 语言的数据类型如图 2-1 所示。

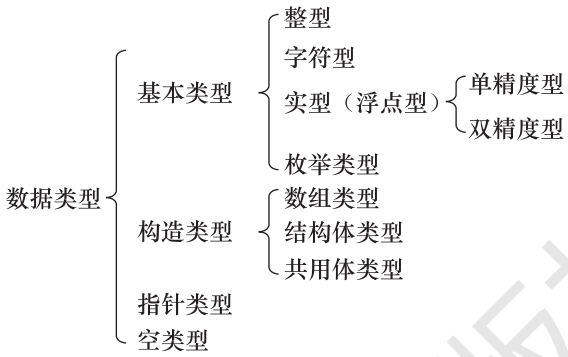

图 2-1　C 语言的数据类型

C 的数据类型

数据类型同时也定义了数据在内存中的存储方式。不同类型的数据所占存储空间及表示范围不同，如表 2-1 所示。

表 2-1　C 语言常用数据类型存储表示

类型		符号	占字节数	数据范围
整型	短整型	short	2	$-32768 \sim +32767$（$-2^{15} \sim +2^{15}-1$）
	基本整型	int	2（TC 环境）	$-32768 \sim +32767$（$-2^{15} \sim +2^{15}-1$）
			4（VC 环境）	$-2147483648 \sim +2147483647$（$-2^{31} \sim +2^{31}-1$）
	长整型	long	4	$-2147483648 \sim +2147483647$（$-2^{31} \sim +2^{31}-1$）
实型	单精度	float	4	$3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$
	双精度	double	8	$1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$
字符型		char	1	ASCII 值（一个整数）
空类型		void	0	{}

小贴士

因为不同数据占的内存空间大小不一样，所以分为不同的类型。在编程时，为变量定义合理的数据类型可以避免浪费存储空间。

任务实施

题号	用到的数据	类型
1	存放年份的变量 year	int
2	存放半径的变量 *r* 和存放面积的变量 area	float
3	存放大写字母的变量 upper 和存放小写字母的变量 lower	char

任务 2　常量和变量

任务目标

1. 能熟练使用不同的方式表示程序中的常量，具备变通思维；
2. 能够为程序中的变量定义合理的数据类型，具备质量意识；
3. 能够为变量取一个正确且合理的名字，具备规范编程意识。

任务描述

下列程序的功能是计算圆的面积，分析其中包含的数据元素。

```
#define  PI 3.14159
#include <stdio.h>
void main()
{
  float r,area;
  printf(" 请输入圆的半径 :\n");
  scanf("%f",&r);            /* scanf 是格式输入函数 */
  area=PI*r*r;
  printf(" 圆的面积是 :%f\n",area);
}
```

预备知识

常量与变量

一、常量

在程序运行的过程中，其值不可改变的量称为常量。C 语言中的常量分为直接常量和符号常量。

1. 直接常量

直接常量也称为“字面常量”，即日常所说的“常数”（常量）。直接常量可分为不同的类型，有整型常量、实型常量、字符常量和字符串常量。

（1）整型常量

整型常量又称整数，在 C 语言中，整数可以使用十进制、八进制、十六进制的形式来书写，如表 2-2 所示。

表 2-2　整型常量的三种表示方式

表示方式	前置符号	构成	示例
十进制	无	0~9、正负号	37，-37
八进制	0	0~7、正负号	037，-037
十六进制	0x 或 0X	0~9、a~f（或 A~F）、正负号	0x37，0Xff

【例 2-1】分析下列三个程序的功能，写出它们的执行结果。

程序 1：

```
#include <stdio.h>
void main()
{   int a,b,sum;
    a=39;                          /* a,b 为十进制 */
    b=41;
    sum=a+b;
    printf("%d\n",sum);
}
```

程序 2：

```
#include <stdio.h>
void main()
{   int a,b,sum;
    a=037;                         /* a,b 为八进制 */
    b=041;
    sum=a+b;
    printf("%o\n",sum);
}
```

程序 3：

```
#include <stdio.h>
void main()
 { int a,b,sum;
    a=0x39;                        /* a,b 为十六进制 */
    b=0x41;
    sum=a+b;
    printf("%x\n",sum);
}
```

程序分析：

这三个程序的功能相同，均是用来求 a 和 b 两个整数的和。区别在于三个程序采用的整数表示形式不同，分别是十进制、八进制和十六进制。

程序 1 的执行结果是十进制：80

程序 2 的执行结果是八进制：100

程序 3 的执行结果是十六进制：7a

1. 数据以 0 打头，即为八进制；以 0x（或 0X）打头，即为十六进制。

2. 实际应用中，我们一般选用自己最熟悉的十进制进行程序设计。

（2）实型（浮点型）常量

实型常量即实数，只能用十进制的形式表示，不能用八进制或十六进制的形式表示。实型常量可以用小数形式和指数形式表示，如表 2-3 所示。

表 2-3　实型常量的不同表示方式

表示方式	符号标志	构成	示例	规则
十进制小数	小数点 .	0~9、正负号、小数点	1.23，.23，-1.	必须有小数点
指数	字母 e 或 E	0~9、正负号、e 或 E	1.23e3，1.23E3	字母 e 或 E 前必须有数字，e 或 E 的后面必须为整数

（3）字符常量

字符常量是用单引号括起来的一个字符，其中，单引号是字符常量的定界符。一个字符常量的值是由该字符的 ASCII 码来确定的。例如：‘a’ ‘A’ ‘@’ ‘；’等都是合法的字符常量，其中‘a’和‘A’是不同的字符常量。

在 C 语言中，还允许使用一些特殊形式的字符常量，它是一个以反斜杠“\”开头的字符序列，称为“转义字符”，用于表示 ASCII 字符集内的控制代码和某些功能定义，如换行、回车等。C 语言中常见的转义字符如表 2-4 所示。

表 2-4　转义字符功能表

字符	功能	ASCII 码
\0	字符串结束	0
\n	换行，将当前位置换到下一行行首	10
\t	横向跳格，即从当前位置跳到下一个输出区	9
\v	竖向跳格	11
\b	退格，将当前位置移到前一列	8

续表

字符	功能	ASCII 码
\r	回车，将当前位置移到本行行首	13
\f	换页，将当前位置移到下一页开头	12
\'	单引号字符	39
\"	双引号字符	34
\\	反斜杠字符	92
\ddd	1~3 位八进制数表示的 ASCII 码所代表的字符，如 \101 表示‘A’	0~255
\xhh	1~2 位十六进制数表示的 ASCII 码所代表的字符，如 \x61 表示‘a’	0~255

（4）字符串常量

字符串常量是由一对双引号括起来的字符序列。

例如：“ABC” “How are you!” “English” “12345”等都是合法的字符串常量。

①字符串长度。字符串中包含的字符个数称为该字符串的长度。

②存储方式。为了便于对字符串常量的存储和操作，C 语言的编译程序在存放字符串时，总是自动地在字符串的结尾加转义字符‘\0’作为字符串结束的标志。因此字符串常量存储时，系统为其分配的存储空间是字符串常量的长度加 1 个字节。

例如：字符串“China”在内存中存储如下：

C	h	i	n	a	\0

字符在内存中都是用其 ASCII 码来存储的，上面的字符串实际存放的形式如下：

67	104	105	110	97	0

‘A’和“A”是不同的。前者是字符常量，后者是字符串常量，它们在内存中的示意图如下：

‘A’ | 65 |　　“A” | 65 | 0 |

【测一测】

回答下列问题：

（1）数据 7，7.0，‘7’有什么区别？

（2）字符数据‘a’和“a”有何区别？

（3）字符串“123\n456”和“123\0456”的长度分别是多少？

2. 符号常量

可以用一个标识符代表一个常量，该常量称为符号常量。在 C 语言中，定义符号常量的方法有两种。

（1）使用编译预处理命令 define

```
#define N 100
#define PI 3.14159
```

（2）使用常量说明符 const

```
const float PI=3.14159
```

【例 2-2】符号常量示例。

```
#define HIGH 10                    /*  定义符号常量 HIGH 代表 10  */
#include <stdio.h>
void main()
{
  int a,b;
  a=10;
  b=a*HIGH;
  printf("b=%d\n",b);
}
```

程序运行结果：

```
b=100
```

使用符号常量的好处：

1. 含义清楚。
2. 在需要修改一个常量时，能做到“一改全改”。

二、变量

变量可以理解为存放数据的容器（即存储单元）。它的功能是用来存放程序中需要处理的数据。在前面的示例中，已经多次使用过变量，如 a、b、sum 等，这些量的值随着程序的运行而发生变化。

一个变量必须有一个名字，系统可以通过变量名访问变量。和其他高级语言一样，用来标识变量名、符号常量名、函数名、数组名、文件名的有效字符序列称为标识符（identifier）。简单地说，标识符就是一个名字。

1. 变量名的命名规则

变量名又叫做变量标识符。C 语言规定，标识符只能由字母、数字和下划线组成，且第一个字符必须为字母或下划线。变量名不允许使用 C 语言关键字、系统函数名和系统类名。

关键字是 C 语言系统规定的、具有特定含义和专门用途的一些字符序列。如前面例题中出现的 main、int、float、printf、include、return、define 等。关键字必须用小写字母，不能用错，也不能挪作他用。

除了满足变量名命名规则外，为了保证程序代码的易读性和高质量，程序员们通常会对变量名做一些约束，例如：

（1）选择有意义的变量名使程序能够“自说明”。也就是说，变量名最好“见名知意”，以增加程序的可读性，如 sum、area 等。

（2）遵循最小化长度与最大化信息量原则。即：在保证一个标识符意思明确的同时，应当尽量缩短其长度。

（3）变量名区分大小写，因此，sum 和 SUM 被认为是两个不同的变量名。一般来说，变量名用小写字母表示，与人们日常习惯一致。

（4）变量名中不允许插入空格。

【测一测】

区分以下标识符中，哪些是合法的标识符，哪些是不合法的标识符。

```
a1          sum         int         include     for
m.y         2x          b_1         while       _5
a*b         x>y         *p          xyz         π
float       main        aver        ABC         abc
```

2. 变量的定义

变量必须遵循“先定义，后使用”的原则，以便编译系统为它分配相应的存储单元。变量定义的一般形式为：

```
类型标识符　变量名 1，变量名 2，……变量名 n;
```

【例 2-3】变量声明示例。

```
int a,b;                    /* 定义 a,b 为整型变量 */
float e,f,g;                /* 定义 e,f,g 为实型变量 */
char c1,c2;                 /* 定义 c1,c2 为字符型变量 */
```

3. 变量的赋值

将某一数值赋给某个变量的过程称为赋值。所赋的“值”可以是数字，也可以是字符串和表达式。

可以在定义变量的同时给变量赋一个初值，称为变量的初始化。例如：

```
int sum=0;
float d=3.2;
```

也可以先定义变量，然后再为其赋值。例如：

```
int sum;
float d;
sum=0;
d=3.2;
```

【测一测】

（1）定义 a，b，c 为整型变量，分别赋值 1，2，3。

（2）定义 x1，x2 为双精度型变量，并赋初值 0。

任务实施

解答：计算圆面积的任务中包含如下数据元素：

（1）包含 2 个变量：半径 *r* 和圆面积 area。

（2）包含 1 个符号常量：PI（代表圆周率 π）。

（3）包含 7 个关键字：define、include、void、main、float、printf、scanf。

（4）包含 2 个运算符：*（乘号）和 =（赋值）。

任务 3　算术表达式与赋值表达式

任务目标

1. 能正确书写并使用 C 语言的算术运算符与算术表达式；
2. 能正确书写 C 语言的赋值运算符与赋值表达式；
3. 能够把“小事”做细，树立“细节决定成败”的意识，做一个做事严谨的人。

任务描述

键盘输入一个 3 位整数，求它的百位、十位和个位，并输出。

预备知识

一、算术运算符与算术表达式

1. 算术运算符

C 语言的算术运算符有两类：基本的算术运算符和自增自减运算符。

算术运算和赋值运算

（1）基本算术运算符

基本的算术运算符有 5 个，如表 2-5 所示。

表 2-5　基本的算术运算符

运算符	含义	举例	结果	说明
+	加法运算符	5+2	7	无
−	减法运算符	5−2	3	无
*	乘法运算符	5*2	10	书写规则：*
/	除法运算符	5/2	2	书写规则：/ 运算规则：两个整数相除结果为整数（舍去小数部分，不四舍五入）
%	求余运算符	5%2	1	求余运算仅用于两个整数间的运算，若存在负数，余数的符号与被除数的符号相同 例如：−17%5 的余数为 −2，17%−5 的余数为 2

1. +、−、*、/、% 的运算对象都是两个，因此叫“双目”运算符。

2. 优先级：先乘除，后加减。求余运算与乘除同级。

3. 结合性均为“左结合性”，即同优先级算术运算按“自左向右”方向进行。

【测一测】分析下列程序的执行结果。

```
#include <stdio.h>
void main()
{   int a=-7,b=3;
    printf("%d,%d,%d,%d,%d\n",a+b,a-b,a*b,a/b,a%b);
}
```

（2）自增自减运算符

C 语言中有两个特殊的运算符：自增运算符“++”和自减运算符“--”。它们都是单目运算符，而且运算对象必须是整型变量。如表 2-6 所示。

表 2-6　自增、自减运算符

运算符	含义	举例	结果	说明
++	自增 1 运算	i++ 或 ++i	使 i 的值加 1	1. ++i 和 --i：运算符前置。即先使变量 i 自增 1 或自减 1，再使用 i。 2. i++ 和 i--：运算符后置。即先使用 i，再使 i 的值自增 1 或自减 1。
--	自减 1 运算	i-- 或 --i	使 i 的值减 1	

1.++ 和 -- 的结合方向是“自右至左”。

2. 在只需对变量本身进行加 1 或减 1 而不考虑表达式的情况下，前置运算和后置运算的效果完全相同，否则，结果是有区别的。

【例 2-4】分析下列程序的输出结果，注意其中前置运算和后置运算的区别。

```
#include <stdio.h>
void main()
{
int i,x,y;
i=5;
x=i++;          /* 后置运算，先把 i 的值赋给 x，然后 i 的值加 1*/
printf("i=%d,x=%d\n",i,x);
i=5;
y=++i;          /* 前置运算，先使 i 的值加 1，然后将 i 的值赋给 y*/
printf("i=%d,y=%d\n",i,y);
}
```

程序运行结果：

```
i=6,x=5
i=6,y=6
```

使用 ++ 和 -- 运算符时，常常会出现一些想不到的副作用，如：a+++b，是理解为（a++）+b 呢，还是 a+（++b），为了避免二义性，可以加上括号，如（a++）+b。

【测一测】

下面两个程序段执行后，变量 x 的值是多少？

程序段 1：

```
int x,a=7;
x=(a++)+3;
printf("%d",x);        /* 变量 x 的值是 _____*/
```

程序段 2：

```
int x,a=7;
x=(++a)+3;
printf("%d",x);        /* 变量 x 的值是 _____*/
```

2. 算术表达式

用算术运算符或小括号将运算对象（也称操作数）连接起来的、符合 C 语法规则的式子，称为 C 的算术表达式。运算对象包括常量、变量、函数等。算术表达式在使用时要注意书写形式。例如：

数学式	C 语言算术表达式
$2a^2+b$	2 *a*a+b
$\|x\|-2$	fabs（x）−2
$\frac{1}{x^2-2x+2}$	1/（x*x−2*x+2）
$\sqrt{b^2-4ac}$	sqrt（b*b−4*a*c）

【测一测】

将下列式子转换成 C 语言表达式：

$$\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

当描述较复杂的算术表达式时，会用到括号 () 和数学函数。C 语言中，算术表达式中用到的括号只能用小括号 ()；当算术表达式中需要用数学函数时，程序中需要包含头文件“math.h”。

二、赋值运算符与赋值表达式

1. 简单赋值

赋值符号“=”就是赋值运算符，它的作用是将一个数据赋给一个变量，如“x=5”是把常量 5 赋给变量 a。

由赋值运算符将一个变量和一个表达式连接起来的式子称为“赋值表达式”。它的一般形式为：

```
< 变量 >=< 表达式 >
```

例如：“x=3+7”是一个赋值表达式。对赋值表达式的求解过程是：先计算表达式“3+7”的值（10），然后再将 10 赋给变量 a。

在赋值表达式后加上分号就构成了赋值语句。例如：

```
a=3+7;
```

赋值语句允许给多个变量连续赋值，但在定义变量时，不允许连续给多个变量赋初值。

例如：a=b=c=d=5；是合法的。

而 int a=b=c=d=5；是不合法的，应该写成：int a=5，b=5，c=5，d=5；

2. 复合赋值

在赋值运算符“=”之前加上其他运算符，就构成了复合赋值运算符。例如：

```
a+=b;       /*  等价于 a=a+b  */
a-=b;       /*  等价于 a=a-b  */
a*=b;       /*  等价于 a=a*b  */
a/=b;       /*  等价于 a=a/b  */
a%=b;       /*  等价于 a=a%b  */
```

赋值运算符和复合赋值运算符的优先级高于逗号运算符，结合方向均为“从右到左”。

【注意】

（1）赋值运算符的左边必须是变量名，被赋值变量的值就是赋值表达式的值。

（2）当赋值表达式中左边的变量与右边的表达式的数据类型不同时，需要进行类型转换。转换的规则是：右边表达式的值被转换成左边变量的数据类型，然后再赋值给变量。

（3）赋值表达式也可以包含复合的赋值运算符。如：

```
a+=a-=a*a
```

如果 a 的初值为 12，此赋值表达式的求解步骤如下：

①先进行“a-=a*a”的运算，它相当于 a=a-a*a=12-144=-132。

②再进行“a+=-132”的运算，它相当于 a=a+（-132）=-132-132=-264。

【测一测】

执行下列程序段后，变量 a 和 b 的值各是多少？

```
程序段 1:int a,b; a=1;b=2;a=b;
程序段 2:int a,b; a=1;b=2;b=a;
程序段 3:int a=1; a=a+1;
```

任务实施

```
/*****************************************************************/
/* 程序 : program2_1.c                                           */
/* 功能 : 键盘输入一个 3 位整数，求它的百位、十位和个位，并输出。          */
/*****************************************************************/
#include <stdio.h>
void main()
{ int number,ge,shi,bai;
  printf(" 请输入一个 3 位整数 :");
  scanf("%d",&number);
  bai=number/100;                    /* 分离百位 */
```

```
    shi=number/10%10;                    /* 分离十位 */
    ge=number%10;                        /* 分离个位 */
    printf("百位数:%d\n",bai);           /* 输出百位 */
    printf("十位数:%d\n",shi);           /* 输出十位 */
    printf("个位数:%d\n",ge);            /* 输出个位 */
}
```

任务 4 条件判断

任务目标

1. 能正确书写 C 的关系表达式和逻辑表达式；
2. 会计算关系表达式和逻辑表达式的值；
3. 能够变抽象为具体，具有举一反三、自主获取知识的能力。

任务描述

用关系表达式或逻辑表达式描述下列条件：

（1）判断 x 是否为奇数。

（2）判断字符变量 ch 是否为英文字母。

（3）判断 year 是否为闰年。

关系运算与逻辑运算

预备知识

一、关系运算符和关系表达式

1. 关系运算符

关系运算符用于比较两个运算对象的大小。C 语言中提供了 6 个关系运算符，如表 2-7 所示。

表 2-7 关系运算符

序号	关系运算符	含义
1	<	小于
2	<=	小于等于
3	>	大于
4	>=	大于等于
5	==	等于
6	!=	不等于

使用关系运算符时，应注意其优先级。

（1）“<”“<=”“>”“>=”这四个运算符的优先级相同，“==”和“!=”的优先级相同。前 4 个运算符的优先级高于后两个。

（2）“==”与“=”不同，前者是关系运算符，后者是赋值运算符。

（3）关系运算符的优先级低于算术运算符而高于赋值运算符。它们的结合性是“自左至右”。

例如：

a!=b+c 等价于 a!=（b+c）；a=b>c 等价于 a=（b>c）。

2. 关系表达式

关系表达式就是用关系运算符将两个或两个以上运算对象连接起来的式子。其中，运算对象可以是常量、变量或表达式。

关系表达式的运算结果有两种：“真”或“假”。在 C 语言中用 1 表示“真”，用 0 表示“假”。例如：3*5>6*2，表达式的执行结果为 1。

【测一测】

printf（"%d"，3>2>1）的值是多少？

二、逻辑运算符和逻辑表达式

1. 逻辑运算符

逻辑运算符用来进行逻辑运算。C 语言中一共提供了 3 个逻辑运算符，如表 2-8 所示。

表 2-8　逻辑运算符

序号	逻辑运算符	含义
1	!	逻辑非
2	&&	逻辑与
3	||	逻辑或

使用逻辑运算符时，应注意它的优先级和结合性。

（1）逻辑运算符从高到低的优先次序为：!（逻辑非）→ &&（逻辑与）→ ||（逻辑或）。

（2）逻辑非“!”的结合方向是“自右至左”，逻辑与“&&”和逻辑或“||”的结合方向是“自左至右”。

（3）各种运算符的优先次序如图 2-2 所示。

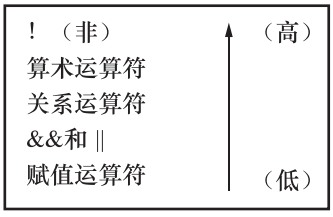

图 2-2　常见运算符的优先次序

2. 逻辑表达式

用逻辑运算符将关系表达式或逻辑量连接起来的式子称为逻辑表达式。逻辑表达式的值也有两种，即“真“和“假”。C 语言的编译系统在给出逻辑运算结果时，以数值 1 代表“真”，以数值 0 代表“假”；在判断一个量的真假时，以 0 代表“假”，以非 0 代表“真”。即将一个非 0 的数值认作为“真”。具体通过以下几个例子进行说明。

（1）若 a=4，则 !a 的值为 0。（因为 a 的值为非 0，被认作“真”，对它进行“非”

运算，得“假”，“假”用 0 表示。）

（2）若 a=4，b=5，则 a&&b 的值为 1。（因为 a 和 b 均为非 0，被认为是“真”，因此 a&&b 的值也是“真”，“真”用 1 表示。）

（3）若 a=4，b=5，则 a||b 的值为 1。

（4）若 a=4，b=5，则！ a||b 的值为 1。

（5）4&&0||2 的值为 1。

通过以上几个例子可以看出，由系统给出的逻辑运算结果不是 0 就是 1，不可能是其他数值。而在逻辑表达式中作为参加逻辑运算的运算对象（操作数）可以是 0（0 按“假”对待）或任何非 0 的数值（非 0 的数值按“真”对待）。

【注意】如果在一个表达式中不同位置上出现数值，应区分哪些是作为数值运算或关系运算的对象，哪些是作为逻辑运算的对象。例如：

```
5>3&&2||8<4-!0
```

表达式自左至右扫描求解，上式的求解过程如下。

（1）处理“5>3”，此处的 5 和 3 作为数值参加关系运算，“5>3”的值为 1。

（2）处理“1&&2”，此时的 1 和 2 均为逻辑运算对象，均作“真”处理，因此结果为 1。

（3）处理“1||8<4−!0”，根据优先次序，先进行“！ 0”运算得 1，要运算的表达式变为“1||8<4−1”，即“1||8<3”，关系运算符“<”两侧的 8 和 3 作为数值参加比较，“8<3”的值为 0（“假”），最后得到“1||0”的结果为 1。

实际上，逻辑运算符两侧的运算对象不但可以是 0 和 1，或是 0 和非 0 的整数，也可以是任何类型的数据，如字符型、实型或指针类型等。系统最终以 0 和非 0 来判定它们属于“真”或“假”。例如：

```
'c'&&'d'
```

的值为 1（因为 c 和 d 的 ASCII 值都不为 0，按“真”处理）。

表 2-9 为逻辑运算的真值表。

表 2-9　逻辑运算的真值表

a	b	!a	!b	a&&b	a||b
非 0	非 0	0	0	1	1
非 0	0	0	1	0	1
0	非 0	1	0	0	1
0	0	1	1	0	0

【注意】在逻辑表达式的求解中，并不是所有的逻辑运算都会被执行到，只是在必须执行下一个逻辑运算符才能求出表达式的解时，才执行该运算符。具体通过以下几个例子进行说明。

（1）a&&b&&c　只有 a 为真（非 0）时，才需要判别 b 的值；只有 a 和 b 都为真时，

才需要判别 c 的值。只要 a 为假，就不必判别 b 和 c（此时，整个表达式已确定为假）。若 a 为真，b 为假，就不必判别 c，如图 2-3 所示。

（2）a||b||c　只要 a 为真（非 0），就不必判断 b 和 c；只有 a 为假，才判别 b；a 和 b 都为假，才判别 c。如图 2-4 所示。

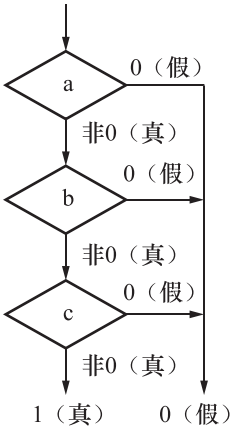

图 2-3　a&&b&&c

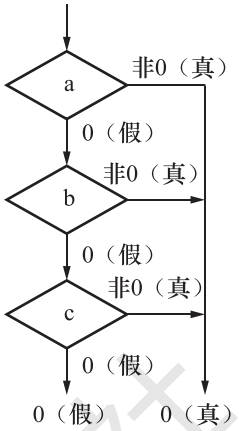

图 2-4　a||b||c

也就是说，对于 && 运算符来说，只有 a ≠ 0 时，才继续进行右边的运算；对于 || 运算符来说，只有 a=0 时，才继续进行右边的运算。因此，如果有下面的逻辑表达式：

```
(m=a>b)&&(n=c>d)
```

当 a=1，b=2，c=3，d=4，m 和 n 的原值为 1 时，由于“a>b”的值为 0，因此 m=0，而“n=c>d”不被执行，因此 n 的值不是 0，而仍保留原值 1。这一点在学习时需注意。

任务实施

下列条件对应的关系表达式或逻辑表达式为：

（1）判断 x 是否为奇数。

```
x%2==1
```

（2）判断字符变量 ch 是否为英文字母。

```
ch>='a' && ch<='z' || ch>='A' && ch<='Z'
```

（3）判断 year 是否为闰年。

```
year%4==0  && year%100!=0 || year%400==0
```

【测一测】

判断三个实数 a，b，c 能否构成一个三角形。

任务 5 数据类型的转换

任务目标

1. 能正确描述 C 的编译系统对不同类型数据进行运算时的转换规则；
2. 能根据编程需要进行强制类型转换；
3. 能够领会“变则通，通则存，存则发”的精髓，拒绝做一个“墨守成规”的人。

任务描述

若有定义：int b=7；float a=2.5，c=4.7；则下面表达式的值是多少？

```
a+(int)(b/3*(int)(a+c)/2)%4
```

预备知识

数据类型转换

在 C 语言的表达中，当运算对象是不同类型的数据时，系统会自动将操作数转换成相同类型后再进行计算。如果这种方法不能满足需要，那么也可以使用类型转换运算符对操作数的类型进行强制转换。

一、自动类型转换

在 C 语言中，允许不同类型的数据进行混合运算。例如：20+1.5−345.678*‘a’是一个合法的表达式。在进行运算时，C 编译系统将自动进行必要的数据类型转换，然后再进行运算。转换规则如图 2−5 所示。

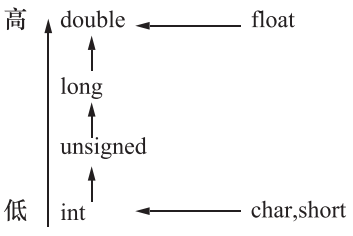

图 2−5 自动类型转换规则

图中，横向箭头表示必定的转换，如字符数据必先转换为整数，short 型转换为 int 型，float 型数据在运算时一律先转换成双精度型，以提高运算精度（即使两个 float 型数据相加，也先都转化成 double 型，然后再相加）。

纵向箭头表示当运算对象为不同类型时转换的方向。例如，int 型与 double 型数据进行运算，先将 int 型数据转换成 double 型，然后进行运算。

【注意】箭头方向只表示数据类型级别的高低，由低向高转换，不要理解为 int 型先转换成 unsigned int 型，再转换成 long 型，再转换成 double 型。如果一个 int 型数据与一个 double 型数据运算，直接将 int 型转换成 double 型；同理，一个 int 型数据与一个 long 型数据运算，直接将 int 型转换成 long 型。

换言之，如果有一个数据是 float 型或 double 型，则另一数据要先转换为 double 型，运算结果为 double 型。如果参加运算的两个数据中最高级别为 long 型，则另一数据先转换为 long 型。其他依此类推。

二、强制类型转换

当自动类型转换不能满足用户的使用需求时，可以在表达式中通过强制类型转换运算符将操作对象强制转换成所需要的数据类型。

强制类型转换的格式如下。

```
（类型标识符）表达式
```

例如：

```
5*(int)6.23        /* 将 6.23 转换成整型，再进行其他运算 */
(float)10/4        /* 将 10 转换成 float 型，再计算除法，表达式的值为 2.5 */
(float)(10/4)      /* 将 (10/4) 转换成 float 型，表达式的值为 2.0 */
```

【说明】

（1）如果要对整个表达式进行转换，表达式应使用括号括起来，否则（float）10/4 是将 10 转换成 float 型，再计算除法。

（2）强制转换时，得到一个所需类型的中间变量，原来变量的类型未发生变化。

任务实施

解题步骤：

（1）先计算 b/3 的值，为 2；得到如下式子：

```
a+(int)(2*(int)(a+c)/2)%4
```

（2）再计算（int）（a+c），值为 7，得到如下式子：

```
a+(int)(2*7/2)%4
```

（3）计算 2*7：值为 14，得到如下式子：

```
a+(int)(14/2)%4
```

（4）计算（int）（14/2），值为 7，得到如下式子：

```
a+7%4
```

（5）先计算 7%4，然后与 a 相加，最终表达式的值为：5.5

【测一测】有下列语句：

```
int x=3,y=2;
float a=2.5,b=3.5;
```

则表达式（x+y）%2+（int）a/（int）b 的值为 _____。

知识拓展

其他运算符

C 语言运算符丰富，除了前面介绍的运算符外，还有以下运算符：

一、条件运算符与条件表达式

1. 条件运算符

条件运算符是由问号“?”和冒号“：”组成的，用于连接 3 个运算对象的运算符，是 C 语言中唯一的一个三目运算符。

条件运算符的优先级高于赋值运算符和逗号运算符，而低于其他运算符。其结合性为“自右至左”。

2. 条件表达式

用条件运算符将运算对象连接起来的式子称为条件表达式。它的一般格式如下。

```
表达式 1？表达式 2：表达式 3
```

条件表达式的求值过程是：先计算表达式 1，若为非 0（真）值，则计算表达式 2，并将其作为整个条件表达式的值；否则计算表达式 3，并将其作为整个条件表达式的值。

例如：

```
a=5>8?100:50
```

由于条件运算符的优先级高于赋值运算符，所以先运算“5>8?100:50”，再将其结果赋给变量 a，最终 a 的值为 50。

二、逗号运算符和逗号表达式

1. 逗号运算符

C 语言提供了一种特殊的运算符——逗号运算符（又称为“顺序求值运算符”），用于将两个表达式连接起来，它是所有运算符中优先级别最低的。例如：

```
3+5,6+8
```

2. 逗号表达式

用逗号运算符将各种类型的表达式连接起来的式子称为“逗号表达式”。

逗号表达式的一般形式如下。

```
表达式 1，表达式 2，…表达式 n
```

逗号表达式的求解过程是：先计算表达式 1，再求解表达式 2，最后计算表达式 n，整个逗号表达式的值是最后一个表达式的值。例如，逗号表达式“3+5，6+8”的值为 14。

再例如：

```
a=3*5,a*4
```

由于赋值运算符的优先级高于逗号运算符，因此应先求解 a=3*5，即将“a=3*5”作为一个表达式计算，经过计算和赋值后得到 a 的值为 15，然后求解 a*4，得 60，整个逗号表达式的值为 60。

一个逗号表达式还可以与另一个表达式组成一个新的逗号表达式，例如：

```
(a=3*5,a*4),a+5
```

先计算出 a 的值等于 15，再进行 a*4 的运算得 60（注意 a 值未变，仍为 15），再进行 a+5 得 20，即整个逗号表达式的值为 20。

【注意】并不是任何地方出现的逗号都是逗号运算符。例如，函数参数也是用逗号来间隔的。例如：

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

此处的“a，b，c”并不是一个逗号表达式，它是 printf 函数的 3 个参数，参数之间用逗号隔开。有关 printf 函数的使用在单元三中详细叙述。

三、其他运算符

1. sizeof 运算符

sizeof 为求字节数运算符，是 C 语言中唯一的关键字运算符，可以计算某一种类型数据所占存储单元的字节个数，它是单目运算符。

例如：假设变量 a 为字符型，则 sizeof（a）的结果为 1（含义是，字符型数据在内存中占 1 个字节）。

2. 小括号运算符

小括号运算符“()”是优先级最高的运算符。用它将某些运算符和运算对象括起来后，这些被括起来的运算符和运算对象要优先运算。例如：

2*3+6 和 2*（3+6），前面表达式的执行结果是 12，后者为 18。

单元小结

通过本单元学习，学生应该掌握以下内容：

（1）C 语言的基本数据类型。编写程序时，能够根据数据的实际情况，选用合理的数据类型。

（2）常量和变量

常量的用法比较简单，可以通过其自身的书写格式说明该常量的类型。

变量在使用之前必须先定义，否则程序无法为其分配存储空间。变量的类型决定变量的取值范围、变量在内存中所占存储空间大小以及变量所能参与运算的种类。变量在

命名时要遵循其命名规则，并且应该“见名知意”。

（3）算术运算与数学中的四则运算类似，但要注意其书写规则与运算规则。

（4）使用自增自减运算符的目的是为了简便程序，程序设计过程中，不建议写出诸如“c=--a+++b”；这类令人容易混淆的语句。可以用下列语句代替：

```
a=a-1;
b=b+1:
c=a-b;
```

（5）赋值语句是程序设计中使用较多的语句，赋值语句具有先计算，后赋值的功能。

（6）关系表达式和逻辑表达式是选择结构程序设计的基础，学习时，要注意相应的运算规则。

（7）不同类型的数据在参加运算之前会自动转换成相同类型，然后再进行运算。此外，还可以使用强制类型转换。

思考与练习

一、选择题

1. 设有说明：char w；int x；float y； double z；则表达式 w*x+z-y 值的数据类型是（　　）。

A. float　　B. char　　C. int　　D. double

2. 语句 printf（“a\bre\' hi\' y\\\bou\n”）；的输出结果是（　　）。（说明：‘\b’是退格符）

A. a\bre\' hi\' y\\\bou　　B. a\bre\' hi\' y\bou

C. re' hi' you　　D. abre' hi' y\bou

3. 下列程序的输出结果是（　　）。

```
#include <stdio.h>
void main()
{
  double d=3.2;
  int x,y;
  x=1.2;
  y=(x+3.8)/5.0;
  printf("%d\n",d*y);
}
```

A. 3　　B. 3.2　　C. 0　　D. 3.07

4. 下列可用作 C 语言用户标识符的一组是（　　）。

A. _data, aaa, main　　B. z x, aver,void

C. x_1,_123,Abc　　D. switch, a-1, sss

二、填空题

1. 若 a 是 int 型，且 a 的初值为 6，则计算表达式 a+=a−=a*a 后，a 的值为 _____。

2. 在 C 语言中的实型变量分两种类型，它们是 _____ 和 _____。

3. 数学式 πr^2 对应的 C 语言表达式为 _____。

4. 语句：

```
int i=-19,j=i%4;
printf("%d\n",j);
```

的输出结果为 _____。

5. 下列程序的输出结果为 _____，功能是 _____。

```
#include <stdio.h>
void main()
  {
  int a=1,b=2;
  a=a+b;
  b=a-b;
  a=a-b;
  printf("%d,%d\n",a,b);
  }
```

实训任务

1. 设变量 a=3，b=7，编程，借助第三个变量实现 a 和 b 值的互换。

2. 下列程序的功能是：输入一个华氏温度 f，要求输出对应的摄氏温度 c，公式为 c=5/9*（f−32）。程序中有若干处错误，请调试改正，使之能够正确运行。

```
#include <stdio.h>
void main()
{
  int c;f;
  c=5/9*(f-32);
  f=120.3;
  printf("c=%f\n",c);
}
```

3. 数据分离与重组。

编写一个程序，将两个两位数的正整数 a 和 b 合并成一个整数放在 c 中。合并方式是：将 a 数的十位和个位依次放在 c 数的个位和百位上，将 b 数的十位和个位放在 c 数的十位和千位上。例如：假如 a=12，b=34，则 c=4231。