第 2 章　数据类型、运算符、表达式

知识点 1　C 语言的数据类型

C 语言的数据结构是以数据类型形式体现的。C 语言的数据类型可以分为基本类型和构造类型两种。其中基本类型有：整型，字符型，实型（也叫浮点型，又可分为单精度型和双精度型两种），枚举类型，指针类型和空类型；构造类型有：数组类型，结构体类型，共用体类型等。

C 语言中数据有常量与变量之分，它们分别属于以上这些类型。由以上这些数据类型还可以构成更复杂的数据结构。例如利用指针和结构体类型可以构成表、树、栈等复杂的数据结构。在程序中对用到的所有数据都必须指定其数据类型。

1.1　常量和变量

1．常量和符号常量

在程序运行过程中，其值不能被改变的量称为常量。常量可以分为不同的类型，如 10、0、-10 为整型常量，12.34、-12.34 为实型常量，'D'、'd'为字符型常量。常量一般从其字面形式即可判别，这种常量称为直接常量；也可以用一个标识符代表一个常量，即符号常量。符号常量在使用前需要先用编译预处理命令定义，如：

```
#define    PI    3.1415926
```

在此定义后，在当前程序中需要使用 3.1415926 的地方都可以用符号常量 PI 代替。

习惯上，符号常量名用大写，变量名用小写，以示区别。但这不是规定，仅是习惯而已。

2．变量

在程序运行过程中，其值可以改变的量称为变量。一个变量应该有一个名字，在内存中占据一定的存储单元。在该存储单元中存放变量的值。变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符。简单地说，标识符就是一个名字。如 sum、average 是合法的标识符，也是合法的变量名，而 123abc、x+y 等是不合法的标识符和变量名。变量在使用前需要进行定义，即先定义，后使用。变量的定义格式如下：

类型　变量名列表;

例如：

```
int    a,b,c;       /* 定义 a、b、c 为整型变量*/
```

3．整型常量的表示方法

整型常量即整常数。C 语言中整常数可用以下 3 种形式表示：

（1）十进制整数。如 100、-123、0。

（2）八进制整数。以 0 开头的数是八进制数。如 012 表示八进制数 12，即$(12)_8$，其值等于十进制数 10。

（3）十六进制整数。以 0x 开头的数是十六进制数。如 0x12，代表十六进制数 12，即 $(12)_{16}$，其值等于十进制数 18。

4．整型变量的定义方法

整型变量的基本类型符为 int。可以根据数值的范围将变量定义为基本整型、短整型或长整型。在 int 之前可以根据需要分别加上修饰符：short（短整型）或 long（长整型），因此有以下 3 种整型变量：

（1）基本整型，以 int 表示。

（2）短整型，以 short int 或 short 表示。

（3）长整型，以 long int 或 long 表示。

若再加上修饰符，可以使用以下 6 种整型变量，即：

- 有符号基本整型：[signed]　int。
- 无符号基本整型：unsigned　int。
- 有符号短整型：[signed]　short [int]。
- 无符号短整型：unsigned　short　[int]。
- 有符号长整型：[signed] long　[int]。
- 无符号长整型：unsigned long　[int]。

以上定义变量格式中，方括号及其中的内容可以省略。以下是定义整型变量举例：

```
int a,b;            /*指定变量 a、b 为整型变量，每个整型数据在内存中占据 2 个字节单元
                      （TC 环境）或 4 个字节单元（VC 环境）*/
unsigned int c,d;   /*指定变量 c、d 为无符号短整型变量，每个短整型数据及无符号短整型
                      数据在内存中均占据 2 个字节*/
long e,f;           /*指定变量 e、f 为长整型变量，每个长整型数据在内存中占据 4 个字节单元*/
a=10;               /*给 a 赋以整数 10*/
```

表 2.1 列出了整数类型的数据的取值范围。

表 2.1　整数类型数据的取值范围

类型	取值范围	
	TC 环境	VC 环境
[signed] int	-32768～32767	-2147483648～2147483647
unsigned int	0～65535	0～4294967295
[signed]　short　[int]	-32768～32767	-32768～32767
unsigned short [int]	0～65535	0～65535
long　[int]	-2147483648～2147483647	-2147483648～2147483647
unsigned　long [int]	0～4294967295	0～4294967295

5．实型常量的表示方法

实数又称浮点数。实数有两种表示形式：

（1）十进制小数形式。它由数字和小数点组成（注意必须有小数点）。如 1.23、0.123、12.3、0.0 都是十进制小数形式。

（2）指数形式。如 1.23e2 或 12.3e1 都代表 1.23×10^2。但注意字母 e（或 E）之前必须有数字，且 e 后面的指数必须为整数。

6．实型变量的定义方法

C 语言中实型变量常用的有单精度型（float）、双精度型（double）两类。

以下是定义实型变量举例：

```
float   x,y;          /*指定变量 x、y 为单精度实型变量，每个单精度实型数据在内存中占据 4 个
                        字节单元，自左向右前 7 位数据有效*/
double   z;           /*指定变量 z 为双精度实型变量，每个双精度实型数据在内存中占据 8 个
                        字节单元，自左向右前 16 位数据有效*/
x=12.34;              /*给 x 赋以单精度实数 12.34*/
z=1234.5678;          /*给 z 赋以双精度实数 1234.5678*/
```

7．字符常量

C 语言中的字符常量是用单引号（即撇号）括起来的一个字符。如'a'、'A'、'!'、' '等都是字符常量。注意，'a'和'A'是不同的字符常量。

除了以上形式的字符常量外，C 语言中还允许使用一种特殊形式的字符常量，即以一个“\”开头的字符序列。例如，前面已经遇到过的在 printf 函数中的'\n'，它代表一个“换行”符。这是一种“控制字符”，即转义字符，在屏幕上是不能显示的，在程序中也无法用一个一般形式的字符表示，只能采用特殊形式来表示。常用的转义字符如表 2.2 所示。

表 2.2　常用转义字符及其含义

符号	含义
\n	换行，将当前位置移到下一行开头
\t	水平制表（横向跳格）
\b	退格，将当前位置移到前一列
\r	回车，将当前位置移到本行开头
\f	换页，将当前位置移到下页开头
\\	反斜杠字符（\）
\'	单撇号（'）
\"	双撇号（"）
\ddd	1～3 位八进制数所代表的字符
\xhh	1～2 位十六进制数所代表的字符

8．字符变量

字符型变量用来存放字符常量，一个字符型的变量中只能存放一个字符，不能存放一个字符串。字符型变量的定义形式如下：

```
char ch1,ch2,ch3;                /*指定变量 ch1、ch2、ch3 为字符型变量，每个字符型数据在内存中
                                   占据 1 个字节单元*/
ch1='a';ch2='\101';ch3='\x41' ;  /*给 ch1 赋以字符 a，给 ch2 赋以字符 A，给 ch3 赋以字符 A*/
```

9．字符串常量

字符常量是由一对单引号括起来的单个字符。C 语言除了允许使用字符常量外，还允许使用字符串常量。字符串常量是一对双引号括起来的字符序列。例如"hello!"、"china"、"A"、"1.235"等都是字符串常量。可以用输出函数输出一个字符串，例如：

```
printf("how do you do.");
```

不要将字符常量与字符串常量混淆。'a'是字符常量，"a"是字符串常量，二者不同。可以使用一个字符型的数组存放一个字符串。在字符串的最后，系统会自动添加'\0'作为字符串的结束标志。字符串常量"china"在内存中的实际存放形式为"china\0"。

1.2 数据类型转换

表达式中不同类型的数据运算时可以进行转换，数据类型转换可分自动（隐式）转换和强制转换两种。

1．自动转换

一般两个操作数运算时，要求两个操作数的类型相同，若不相同则系统自动将低类型数据转换成高类型数据，然后实现运算得高类型数据结果（在转换过程中数据的精度保持不变）。各种数据类型的高低顺序如图 2.1 所示。图中横向向左的箭头表示必定的转换，如字符数据必定先转换为整数，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（即使是两个 float 型数据相运算，也都先转化成 double 型，然后再运算）。纵向的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 型的数据直接转换成 double 型，然后在两个同类型（double 型）数据间进行运算，结果为 double 型。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 型先转化成 unsigned int 型，再转化成 long 型，再转化成 double 型。

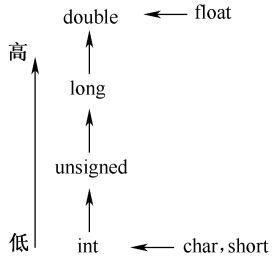

图 2.1　数据类型转化

2．强制转换

强制转换格式如下：

(类型)表达式

或

(类型)(表达式)

其作用是把表达式的值强制转换成所要求的数据类型，如：

```
double x=1.23;
int a,b;
a=(int)x+2.9;
b=(int)(x+2.9);
```

以上代码倒数第二行语句先将 x 强制转换成整型数据 1，即小数部分被舍弃后与实数 2.9 相加得实数 3.9，再将其和的整数部分赋值给变量 a，a 的值为 3。以上代码最后一行语句先将 x 和 2.9 相加得 4.13，再将其强制转换成整型数据 4，即小数部分被舍弃后再赋值给变量 b，b 的值为 4。注意数据类型转换后 x 的值仍然是 1.23。另外在赋值表达式中，当左边的变量和右

边表达式的值类型不一致时，也有一个类型强制转换的问题，此时一律将右边表达式值的类型强制转换成跟左边变量的类型相同，然后赋值。如 a=x;，运算时将右边变量 x 的值强制转换成 int 类型值 1 后赋值给整型变量 a，但 x 的值仍然是 1.23。注意，当含字节数多的类型转换成字节数少的数据类型时，可能会引起精度的降低或出现错误的结果。

【案例 2-1】分析下面程序的输出结果并上机验证。

```
#include   "stdio.h"
main()
{
    int a=10,b=12345;
    float x=12.3456,y=-789.1234;
    char c='A';
    long n=1234567;
    unsigned u=65535;
    printf("%d,%d\n",a,b);
    printf("%4d,%4d\n",a,b);
    printf("%f,%f\n",x,y);
    printf("%-10f,%-10f\n",x,y);
    printf("%7.2f,%7.2f,%.3f,%.3f,%4f,%4f\n",x,y,x,y,x,y);
    printf("%e,%10.2e\n",x,y);
    printf("%c,%d,%o,%x\n",c,c,c,c);
    printf("%ld,%lo,%x\n",n,n,n);
    printf("%u,%o,%x,%d\n",u,u,u,u);
    printf("%s,%5.3s\n","computer","computer");
}
```

知识点 2　C 语言的运算符和表达式

C 语言的运算符范围很宽，把除了控制语句和输入输出以外的几乎所有的基本操作都作为运算符处理，例如将符号“=”作为赋值运算符，方括号作为下标运算符等。C 的运算符有以下几类：

- 算术运算符：+、-、*、/、%、++、--。
- 关系运算符：>、<、==、>=、<=、!=。
- 逻辑运算符：!、&&、||。
- 位运算符：<<、>>、~、|、∧、&。
- 赋值运算符：=及其扩展赋值运算符。
- 条件运算符：?:。
- 逗号运算符：,。
- 指针运算符：*和&。
- 求字节数运算符：sizeof。
- 强制类型转换运算符：(类型)。

- 分量运算符：.、->。
- 下标运算符：[]。
- 其他：如函数调用运算符()。

2.1 算术运算符和算术表达式

1．算术运算符

+：加法运算符或正值运算符，如 1+2、+10。

-：减法运算符或负值运算符，如 1-2、-10。

*：乘法运算符，如 1*2。

/：除法运算符，如 5/3。

%：模运算符，或称求余运算符，%两侧均应为整型数据，如 5%3 的值为 2。

需要说明的是，两个整数相除的结果为整数，如 5/3 的结果值为 1，舍去小数部分。但是，如果除数或被除数中有一个为负值，则舍入的方向是不固定的。例如，-5/3 在有的机器上得到结果-1，有的机器则给出结果-2。多数机器采取“向零取整”的方法，即 5/3=1，-5/3=-1，取整后向零靠拢。如果参加+、-、*、/运算的两个数中有一个数为实数，则结果是 double 型，因为所有实数都按 double 型进行运算。

2．算术表达式和运算符的优先级与结合性

用算术运算符和括号将运算对象（也称操作数）连接起来的、符合 C 语言语法规则的式子称为 C 算术表达式。运算对象包括常量、变量、函数等。例如，下面是一个合法的 C 算术表达式：a*b/c-1.2+'A'。

C 语言规定了运算符的优先级和结合性。在表达式求值时，先按运算符的优先级别高低次序执行，例如先乘除后加减。如表达式 a-b*c，b 的左侧为减号，右侧为乘号，而乘号优先于减号，因此，相当于 a-(b*c)。如果在一个运算对象两侧的运算符的优先级别相同，则按规定的“结合方向”处理。如 a-b+c，先做 a-b，再将其结果和 c 相加得到最后结果。

C 规定了各种运算符的结合方向（结合性），算术运算符的结合方向为“自左至右”，即先左后右。“自左至右的结合方向”又称“左结合性”，即运算对象先与左面的运算符结合。以后可以看到有些运算符的结合方向为“自右至左”，即右结合性（例如赋值运算符）。如果一个运算符的两侧的数据类型不同，则会先自动进行类型转换，使二者具有同一种类型，然后再进行运算。

3．自增、自减运算符

自增、自减运算符的作用是使变量的值增 1 或减 1，例如：

++i，--i：在使用 i 之前，先使 i 的值加（减）1。

i++，i--：在使用 i 之后，使 i 的值加（减）1。

++i 和 i++的作用都相当于 i=i+1。但其不同之处在于：++i 是先执行 i=i+1 后，再使用 i 的值，即先自增后操作；而 i++是先使用 i 的值后，再执行 i=i+1，即先操作后自增。如果 i 的原值等于 2，则执行下面的赋值语句（设这两条语句分别执行）：

```
k=++i;   /*i 的值先变成 3，再赋给 k，k 的值为 3*/
k=i++;   /*先将 i 的值 2 赋给 k，k 的值为 2，然后 i 变为 3*/
```

又如：

```
i=3;
printf("%d",++i);
```

输出 4；若改为

```
printf("%d",i++);
```

则输出 3。

注意

（1）自增运算符（++）和自减运算符（--），只能用于变量，而不能用于常量或表达式，如 5++或(a+b)++都是不合法的。

（2）++和--的结合方向是“自右至左”。

自增（减）运算符常用在循环语句中，使循环变量自动加（减）1，也可用于指针变量，使指针指向下一个地址。

2.2 赋值运算符及赋值表达式

1．赋值运算符

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“a=10”的作用是执行一次赋值操作（或称赋值运算），把常量 10 赋给变量 a。也可以将一个表达式的值赋给一个变量，如 a=3*4-1，该操作执行后 a 的值为 11。

2．复合赋值运算符

在赋值运算符“=”之前加上其他运算符，可以构成复合赋值运算符。如果在“=”前加上一个“+”运算符就成了复合运算符“+=”。例如：

a+=10	等价于	a=a+10
b*=c+6	等价于	b=b*(c+6)
d%=2	等价于	d=d%2

注意

如果赋值号右边是包含若干项的表达式，则相当于它有括号。例如，b*=c+6 等价于 b=b*(c+6)。

凡是二元（二目）运算符，都可以与赋值运算符一起组合成复合赋值运算符。C 语言规定可以使用 10 种复合赋值运算符，即+=、-=、*=、/=、%=、<<=、>>=、&=、∧=、|=。

C 语言采用这种复合运算符，一是为了简化程序，使程序精炼，二是为了提高编译效率。

3．赋值表达式

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><赋值运算符><表达式>

如 a=10 是一个赋值表达式。对赋值表达式求解的过程是：将赋值运算符右侧的“表达式”的值赋给左侧的变量。赋值表达式的值就是被赋值的变量的值。上述一般形式的赋值表达式中的“表达式”又可以是一个赋值表达式。如 a=(b=10)，括号内的 b=10 是一个赋值表达式，它的值等于 10。a=(b=10)相当于 b=10 和 a=b 两个赋值表达式，因此 a 的值等于 10，整个赋值表达式的值也等于 10。赋值运算符按照“自右至左”的顺序结合。

2.3 关系运算符和关系表达式

1．关系运算符

关系运算实际上就是“比较运算”。将两个值进行比较，判断其比较的结果是否为真。例如，x>10 是一个关系表达式，大于号（>）是一个关系运算符，如果 x 的值为 15，则满足给定的“x>10”条件，因此关系表达式的值为“真”（即“条件满足”）；如果 x 的值为 5，不满足“x>10”条件，则称关系表达式的值为“假”。C 语言中以 1 代表“真”，以“0”代表“假”。

C 语言提供了 6 种关系运算符：

优先级相同（高）的有：<（小于）、<=（小于或等于）、>（大于）、>=（大于或等于）。

优先级相同（低）的有：==（等于）和!=（不等于）。

关于优先次序：

（1）前 4 种关系运算符（<、<=、>、>=）的优先级别相同，后两种（!=和==）也相同。但前 4 种的优先级高于后两种。例如，“>”优先于“==”，而“>”与“<”优先级相同。例如，3==3<5 的结果值为 0（假），而不为 1（真）。因为是先做 3<5，结果值为 1，将其与 3 比较，不相等，所以结果值为假；而不是先做 3==3，再将其结果与 5 比较。

（2）关系运算符的优先级低于算术运算符。

（3）关系运算符的优先级高于赋值运算符。

2．关系表达式

用关系运算符将两个表达式（可以是算术表达式、关系表达式、逻辑表达式、赋值表达式、字符表达式）连接起来的式子，称为关系表达式。例如，下面都是合法的关系表达式：

x>y　x+y<z-y　'x'<'y'

关系表达式的结果值是一个逻辑值，即“真”或“假”。例如，关系表达式 10==20 的值为“假”，20>=10 的值为“真”。现假设：x=10，y=20，z=30，则关系表达式 x<y 的值为“真”，表达式的结果值为 1；关系表达式(x>y)!=z 的值为“真”，表达式的结果值为 1；关系表达式 x+y<z 的值为“假”，表达式的结果值为 0。

2.4 逻辑运算符和逻辑表达式

1．逻辑运算符及其优先次序

C 语言提供了 3 种逻辑运算符：

（1）&&（逻辑与）。

（2）||（逻辑或）。

（3）!（逻辑非）。

“&&”和“||”是双目（元）运算符，它要求有两个运算量（操作数），如(a>b)&&(x>y)、(a>b)||(x>y)。“!”是单目（元）运算符，只要求有一个运算量，如!(a>b)。逻辑运算的结果为逻辑量真（1）或假（0），在一个逻辑表达式中如果包含多个逻辑运算符，如!x &&y||a<b||z，在运算时会按以下的优先次序进行运算：

（1）!（非）→&&（与）→（或），即“!”为三者中优先级最高的。

（2）逻辑运算符中的“&&”和“||”低于关系运算符，“!”高于算术运算符。

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就是逻辑表达式，逻辑表达式的结果值应该是一个逻辑量“真”或“假”。C 语言编译系统在给出逻辑运算结果时，以数值 1 代表“真”，以 0 代表“假”，但在判断一个量是否为“真”时，以 0 代表“假”，以非 0 代表“真”。即作为条件时将一个非零的数值认作为“真”；作为结果时用数值“1”作为“真”。例如：

（1）若 x=10，则!x 的值为 0。

（2）若 x=10，y=20，则 x&&y 的值为 1，x||y 的值为 1，!x||y 的值也为 1。

在逻辑表达式的求解中，并不是所有的逻辑运算符都被执行，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。例如：

（1）x&&y&&z。只有 x 为真（非 0）时，才需要判断 y 的值，只有 x 和 y 都为真的情况下才需要判断 z 的值。只要 x 为假，就不必判断 y 和 z（此时整个表达式已确定为假）。如果 x 为真，y 为假，也不判断 z。

（2）x||y||z。只要 x 为真（非 0），就不必判断 y 和 z；只有 x 为假，才判断 y；x 和 y 都为假才判断 z。

也就是说，对&&运算符来说，只有 x≠0，才继续进行右面的运算；对‖运算符来说，只有 x=0，才继续进行其右面的运算。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 而仍保持原值 1。

2.5　条件运算符

若 if 语句中，在表达式为“真”或“假”时都只执行其中一个赋值语句给同一个变量赋值，则可以用简单的条件运算符来处理。例如，若有以下 if 语句：

```
if   (a>b)     max=a;
else           max=b;
```

可以用下面的条件运算符来处理：

```
max=(a>b)? a:b;
```

该语句的执行结果就是将 a 和 b 二者中的大者赋给 max。其中(a>b)?a:b 是一个“条件表达式”。它是这样执行的：如果(a>b)条件为真，则条件表达式取值 a，否则取值 b。

条件运算符是 C 语言中唯一的一个三目运算符，其要求有 3 个操作对象，故称三目（元）运算符。条件表达式的一般形式为：

表达式 1? 表达式 2:表达式 3

在执行时先求解表达式 1，若为非 0（真），则求解表达式 2，此时表达式 2 的值就作为整个条件表达式的值；若表达式 1 的值为 0（假），则求解表达式 3，表达式 3 的值就是整个条件表达式的值。条件运算符的优先级别高于赋值运算符，但比关系运算符和算术运算符都低。条件运算符的结合方向为“自右至左”。在条件表达式中，表达式 1 的类型可以与表达式 2 和表达式 3 的类型不同。

【案例 2-2】分析下面程序的输出结果并上机验证。

```
#include   "stdio.h"
void   main()
{
    int a,b,c,i,j,k;
    int   m,n;
    double   x,y,z;
    a=b=c=7;
    i=10;m=1;
    x=2.5;y=4.7;
    b+=a;
    c*=a-2;
    z=x+a%3*(int)(x+y)%2/4;
    j=++i;
    k=i++;
    m+=k;
    n=a>=b&&b<c;
    printf("a=%d,b=%d,c=%d,m=%d,n=%d,i=%d,j=%d,k=%d\n",a,b,c,m,n,i,j,k);
    printf("z=%f\n",z);
}
```

2.6 位运算符和位运算

位运算是指进行二进制位的运算。C 语言提供位运算的功能，具有很大的优越性。

1.“按位与”运算符（&）

参加运算的两个数据，按二进位进行“与”运算。如果两个相应的二进位都为 1，则该位相与的结果值为 1，否则为 0，即 0&0=0；0&1=0；1&0=0；1&1=1。

例如，7&5 按位与运算：

```
  7        00000111
  5        00000101
(&)        00000101
```

从以上运算可知，7&5 的值为 5。如果参加“&”运算的数是负数（如-7 & -5），则以补码形式表示为二进制数，然后按位进行“与”运算。按位与运算的主要用途有：清零、取一个数中的某些指定位、保留某一个指定位等。需要注意，7&&5 的结果值为 1，因为&&为逻辑运算符“与”，7 和 5 作为条件其值为真，真值和真值相与，结果为真。

2.“按位或”运算符（|）

两个相应的二进位中只要有一个为 1，该位相或的结果值就为 1，即 0|0=0；0|1=1；1|0=1；1|1=1。

例如，7|6 按位或运算：

```
  7        00000111
  6        00000110
(|)        00000111
```

从以上运算可知，7|6 的值为 7。如果参加“|”运算的是负数（如-7|-6），则以补码形式

表示为二进制数，然后按位进行“或”运算。按位或运算常用来对一个数据的某些位置 1。需要注意，7||6 的结果值为 1，因为||为逻辑运算符“或”，7 和 6 作为条件，其值为真，真值和真值相或，结果为真。

3.“异或”运算符（∧）

参加运算的两个数据，若两个二进位同号，则结果为 0（假）；异号则为 1（真），即 0∧0=0；0∧1=1；1∧0=1；1∧1=0。

例如，7∧5 按位异或运算：

```
  7        00000111
  5        00000101
(∧)        00000010
```

从以上运算可知，7∧5 的值为 2。如果参加“∧”运算的是负数（如-7∧-5），则以补码形式表示为二进制数，然后按位进行“异或”运算。“异或”的意思是判断两个相应的位值是否为“异”，为“异”（值不同）则取真（1），否则为假（0）。需要注意，在 C 语言中“∧”运算符不是表示乘方的意思，若要表示乘方，需要使用 pow()函数，如 x^3 可以写为 pow(x,3)，该函数存于头文件 math.h 中。

4.“取反”运算符（～）

～是一个单目（元）运算符，用来对一个二进制数按位取反，即将 0 变 1，1 变 0。例如～037 是对八进制数 37（即二进制数 00011111）按位求反，结果为 11100000。但!037 的结果值为 0，因为！是逻辑运算符。

5.“左移”运算符（<<）

用来将一个数的各二进制位全部左移若干位。例如 a=a<<2，表示将 a 的各二进制位左移 2 位，右补 0。若 a=5，即二进制数 00000101，左移 2 位得 00010100，即十进制数 20，相当于乘以了 2^2。

6.“右移”运算符（>>）

用来将一个数的各二进制位全部右移若干位。例如 a=a>>2，表示将 a 的各二进制位右移 2 位。移到右端的低位被舍弃，对无符号数，高位补 0。

7. 位运算赋值运算符

位运算符与赋值运算符可以组成复合赋值运算符，如&=、|=、>>=、<<=、∧=。例如，a &=b 相当于 a = a & b。

2.7 逗号运算符和逗号表达式

逗号运算符（,）是 C 语言提供的一种特殊的运算符，用它将两个表达式连接起来。例如：

3+2,3-2

用逗号运算符连接的式子称为逗号表达式，逗号表达式的一般形式为：

表达式 1,表达式 2

逗号表达式的求解过程是：先求解表达式 1，再求解表达式 2。整个逗号表达式的值是表达式 2 的值。例如，上面的逗号表达式“3+2,3-2”的值为 1。逗号表达式的一般形式可以扩展为：

表达式 1,表达式 2,表达式 3,…,表达式 n

它的值为表达式 n 的值。逗号运算符是所有运算符中级别最低的。

执行表达式 a=3+2,3*2 后，a 的值为 5，整个表达式的值为 6；但执行表达式 a=(3+2,3*2)后，a 的值为 6，整个表达式的值也为 6。

【案例 2-3】分析下面程序的输出结果并上机验证。

```
#include   "stdio.h"
void   main()
{
    int a=12,b=10,c,d,e,f,g,h;
    c=a&b;
    d=a|b;
    e=a^b;
    f=a<<2;
    g=a>>2;
    h=~b;
    printf("c=%d\n",c);
    printf("d=%d\n",d);
    printf("e=%d\n",e);
    printf("f=%d\n",f);
    printf("g=%d\n",g);
    printf("h=%d\n",h);
}
```

知识点 3　格式化输入/输出

C 语言本身不提供输入输出语句，输入和输出操作是由函数来实现的。在 C 标准函数库中有一批“标准输入输出函数”，它是以标准的输入输出设备（一般为终端设备）为输入输出对象的。其中常用的有：putchar（输出单个字符）、getchar（输入单个字符）、printf（格式输出）、scanf（格式输入）、puts（输出字符串）、gets（输入字符串）。需要注意，printf 和 scanf 等不是 C 语言的关键字，而只是函数的名字。在使用 C 语言库函数时，需要用编译预处理命令“#include”将有关的“头文件”包含到用户源文件中。在头文件中包含了与用到的函数有关的信息。例如使用标准输入输出库函数时，要用到 stdio.h 文件。文件后缀“h”是 head 的缩写，#include 命令都是放在程序的开头，因此这类文件被称为“头文件”。在调用标准输入输出库函数时，文件开头应有以下编译预处理命令：

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

或

```
#include   "stdio.h"
```

3.1　格式输出函数（printf 函数）

printf 函数的作用是向终端（或系统隐含指定的输出设备）输出若干个任意类型的数据。通过 printf 函数可以输出多个数据，且为任意类型。

printf 函数的一般格式为：

printf("输出格式",输出表列);

输出格式是用双引号括起来的字符串，主要由格式说明、按原样输出的字符及转义字符组成。格式说明由“%”和格式字符组成，如%d、%f 等。它的作用是将输出的数据转换为指定的格式输出。格式说明总是由“%”字符开始的。输出表列是指需要输出的一些数据，可以是常数、变量或表达式。例如：

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

1．格式字符

对不同类型的数据用不同的格式字符。常用的格式字符有以下几种：

（1）d 格式符：用来输出十进制整数。有以下几种用法：

- %d：按整型数据的实际长度输出。
- %md：m 为指定的输出字段的宽度。如果数据的位数小于 m，则左端补以空格；若大于m，则按实际位数输出。例如：

```
printf("%3d,%3d",a,b);
```

若 a=10，b=12345，则输出结果为：

```
10,12345
```

- %ld：输出长整型数据。例如：

```
long   a=135790;
printf("%ld",a);
```

（2）o 格式符：以八进制数形式输出整数。由于是将内存单元中各位的值（0 或 1）按八进制形式输出，因此输出的数值不带符号，即将符号位也一起作为八进制数的一部分输出。

（3）x 格式符：以十六进制数形式输出整数。同样不会出现负的十六进制数。

（4）u 格式符：用来输出 unsigned 型数据，即无符号数，以十进制数形式输出。

一个有符号整数（int 型）也可以用%u 格式输出；反之，一个 unsigned 型数据也可以用%d 格式输出，按相互赋值转换的规则处理。unsigned 型数据也可以用%o 或%x 格式输出。

【案例 2-4】整型数据的输出示例。

```
#include   <stdio.h>
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);
}
```

TC 环境下的运行结果为：

```
a=-1,177777,ffff,65535
b=-2,177776,fffe,65534
```

VC 环境下的运行结果为：

```
a=65535,177777,ffff,65535
b=-2,37777777776,fffffffe,4294967294
```

（5）c 格式符：用来输出一个字符。例如：

```
char   c='a';
printf("%c",c);
```

输出字符‘a’，请注意，“%c”中的 c 是格式符，逗号右边的 c 是变量名，不要弄混。一个整数，只要它的值在 0～255 范围内，则也可以用字符形式输出，在输出前，系统会将该整数作为 ASCII 码转换成相应的字符；反之，一个字符数据也可以用整数形式输出。

【案例 2-5】字符数据的输出示例。

```
#include   <stdio.h>
main()
{
    char ch='a';
    int i=97;
    printf("%c,%d\n",ch,ch);
    printf("%c,%d\n",i,i);
}
```

运行结果为：

```
a,97
a,97
```

（6）s 格式符：用来输出一个字符串。有以下几种用法：

- %s：例如 printf("%s","china");输出"china"字符串（不包括双引号）。
- %ms：输出的字符串占m列，如果字符串本身长度大于 m，则突破m的限制，将字符串全部输出；若串长小于 m，则左补空格。
- %-ms：如果串长小于m，则在m列范围内，字符串向左靠，右补空格。
- %m.ns：输出占m列，但只取字符串中左端的 n 个字符。这 n 个字符输出在m列的右侧，左补空格。
- %-m.ns：其中m、n 含义同上，n 个字符输出在m列范围的左侧，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。

【案例 2-6】字符串的输出示例。

```
#include   <stdio.h>
main()
{
    printf("%4s,%7.2s,%.3s,%-5.3s\n","china","china","china","china");
}
```

运行结果为：

```
china, ch,chi,chi
```

（7）f 格式符：用来输出实数（包括单精度型、双精度型），以小数形式输出。有以下几种用法：

- %f：不指定字段宽度，由系统自动指定，使整数部分全部如数输出，并输出 6 位小数。应当注意，并非全部数字都是有效数字。单精度实数的有效位数一般为 7 位；双精度数也可以用%f 格式输出，它的有效位数一般为 16 位，其中包括 6 位小数。
- %m.nf：指定输出的数据共占m列，其中有 n 位小数。如果数值长度小于m，则左端补空格。

- %-m.nf：与%m.nf 基本相同，只是使输出的数值向左端靠，右端补空格。

【案例 2-7】实数的输出示例。

```
#include   <stdio.h>
main()
{
    float x,y;
    x=111111.111;y=222222.222;
    float f=123.456;
    printf("%f",x+y);
    printf("%f   %10f   %10.2f   %.2f   %-10.2f\n",f,f,f,f,f);
}
```

输出结果如下：

```
333333.328125
123.456001 123.456001        123.46    123.46    123.46
```

（8）e 格式符：以指数形式输出实数。其常用形式有%e、%m.ne 和%-m.ne 三种。

（9）g 格式符：用来输出实数，它根据数值的大小自动选择 f 格式或 e 格式，选择输出时占宽度较小的一种，且不输出无意义的 0。

2．注意事项

在使用 printf 函数时，需要注意以下几点：

（1）除了 x、e、g 外，其他格式字符必须用小写字母，如%d 不能写成%D。

（2）可以在 printf 函数中的“格式控制”字符串内包含转义字符，如“\n”、“\t”、“\b”、“\r”、“\f”、“\377”等。

（3）上面介绍的 d、o、x、u、c、s、f、e、g 等字符，如用在“%”后面就作为格式符号。一个格式说明以“%”开头，以上述 9 个格式字符之一为结束，中间可以插入附加格式字符（也称修饰符）。

（4）如果想输出字符“%”，则应该在“格式控制”字符串中用连续两个%表示。

3.2 格式输入函数（scanf 函数）

scanf 函数的一般形式为：

scanf("输入格式",地址表列);

输入格式的含义和 printf 函数中的格式说明相似，以%开始，以一个格式字符结束，中间可以插入附加的字符。地址表列是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。在输入格式中若给 unsigned 型变量输入数据，使用%u、%d、%o、%x 格式均可；可以指定输入数据所占的列数，系统自动按它截取所需数据，但输入数据时不能规定精度，例如：

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

是不合法的。

【案例 2-8】用 scanf 函数输入数据示例。

```
#include   <stdio.h>
main()
```

```
{
   int a,b,c;
   scanf("%d%d%d",&a,&b,&c);
   printf("%d,%d,%d\n",a,b,c);
}
```

运行时按以下方式输入 a、b、c 的值：

```
1    2       3      /* 给 a、b、c 变量输入数据*/
```

运行结果为：

```
1,2,3
```

地址表列中的&a、&b、&c 中的“&”是“地址运算符”，&a 指 a 在内存中的地址。上面 scanf 函数的功能是：按照 a、b、c 在内存中的地址将 a、b、c 的值存进去。输入格式中的“%d%d%d”表示按十进制整数形式输入数据。输入数据时，在两个数据之间以一个或多个空格间隔，也可以用 Enter 键、跳格键 tab 间隔。下面的输入均是合法的：

① 1　2　3

② 1
　2
　3

③ 1（按 tab 键）2　（按 tab 键）3

此外，在使用 scanf 函数时还需要注意以下几点：

（1）scanf 函数中的“格式控制”后面应该是变量地址，而不应是变量名。例如，如果 a、b 为整型变量，则

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

是不对的，应将“a,b”改为“&a,&b”。

（2）如果在“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入与这些字符相同的字符。例如：

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

输入时应用如下形式：

```
1,2
```

注意 1 后面是逗号，它与 scanf 函数中的“格式控制”中的逗号对应。如果输入时不用逗号而用空格或其他字符则是不对的。

（3）在用“%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，例如：

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

如输入 a b c，则会将字符'a'送给 c1，空格字符' '送给 c2，字符'b'送给 c3，因为%c 只要求读入一个字符，后面不需要用空格作为两个字符的间隔，因此空格字符'　'作为下一个字符送给 c2。

（4）在输入数据时，遇到空格或按 Enter 键或“跳格键”（Tab）或达到指定的宽度或遇非法输入都被认为该数据输入结束。

习题 2

一、选择题

1．以下正确的标识符是（　）。

A．b_2　　B．2b　　C．b=2　　D．b-2

2．表达式“100!=200”的值是（　）。

A．true　　B．非零值　　C．0　　D．1

3．以下变量名全部合法的是（　）。

A．abc、a10、a_1、_a1　　B．? 123、*a、a-b、_ab

C．123?、wang、*a、abc　　D．wang_wang、B、while、123

4．在下面的运算符中，优先级最低的运算符是（　）。

A．>=　　B．=　　C．%　　D．||

5．C 语言中，运算对象必须是整型数据的运算符是（　）。

A．%　　B．/　　C．!　　D．*（乘）

6．设 x、y 均为整型变量，且 x 的值为 5，y 的值为 2，以下值为 1 的表达式是（　）。

A．!(y==x/2)　　B．y!=x%3

C．x>0&&y<0　　D．x!=y||x>=y

7．执行下列语句，要使 x、y 均为 1.23，从键盘上的正确输入是（　）。

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

A．1.23,1,23　　B．1.23 1.23

C．x=1.23,y=1.23　　D．x=1.23 y=1.23

8．设整型变量 m、n、a、b、c、d 的值均为 1，执行语句(m=a>b)||(n=c>d)后，m、n 的值为（　）。

A．0　0　　B．0　1　　C．1　0　　D．1　1

9．设 ch 为 char 型变量，其值为'A'，则下面表达式的值是（　）。

```
ch=(ch>='A' && ch<='Z')?(ch+32):ch
```

A．A　　B．a　　C．Z　　D．z

10．在 C 语言中运算符的优先级从高到低的排列顺序是（　）。

A．关系运算符　算术运算符　赋值运算符

B．算术运算符　赋值运算符　关系运算符

C．赋值运算符　关系运算符　算术运算符

D．算术运算符　关系运算符　赋值运算符

11．以下正确的选项是（　）。

A．10++　　B．(x+y)++

C．++(x-y)　　D．(j++)+(j++)+(j++)

12．以下正确的赋值表达式是（　）。

A．x=2+y--=3+c　　B．(x=10-2,y+3),z-3

C．x=y--=z--　　D．x=y+5=y-z

13．执行以下语句后，a 和 b 的值分别为（　）。

```
int a=1,b=2;
a=a^b;
b=b^a;
a=a^b;
```

A．a=1　b=2　　B．a=2　b=2

C．a=2　b=1　　D．a=1　b=1

二、填空题

1．设 a、b、c 均为 int 型变量，且值均为 6，则执行“a-=b-c”后，a=_____；再执行“a%=b+c”后，a= _____。

2．表达式'A'+32-3/5*6 的值是_____，a=b=c=2+3/5 的值是_____。

3．能表示 2<a<3 或 a<-10 的 C 语言表达式是_____。

4．设 a=3，b=4，c=5，则 a+b>c&&b==c 的值为_____，!(b>c)+(b!=a)||(a+b)&&(b-c)的值为_____。

5．表达式 !(x<0) 的等价表达式是_____。

6．以下程序的运行结果是_____。

```
void  main()
{
    int  i,j;
    float  a,b;
    long  k;
    i=9;j=-4;
    a=12.3;b=4.5;
    k=a/b;
    k=a-i/j;
    printf("%ld\n",k);
}
```

7．表达式 6&&5 的值是_____，6&5 的值是_____，6||5 的值是_____，6|5 的值是_____，3<<2 的值是_____。

三、编程题

1．要将“china”译成密码，密码规律是：用原来字母后面的第 4 个字母代替原来的字母。例如，字母“a”后面的第 4 个字母是“e”，用“e”代替“a”。因此，“china”应译为“glmre”。请编写一个程序，用赋初值的方法使 c1、c2、c3、c4、c5 五个变量的值分别为'c'、'h'、'i'、'n'、'a'，经过运算，使 c1、c2、c3、c4、c5 分别变为'g'、'l'、'm'、'r'、'e'，并输出。

2．编程求下面算术表达式的值：

```
(float)(a+b)/2+(int)x%(int)y
```

设 a=2，b=3，x=3.5，y=2.5。

3．编程输出下面表达式运算后 a 的值，设原来 a=10。a 和 b 已定义为整型变量。

（1）a+=a。

（2）a-=5。

（3）a*=2+4。

（4）a/=a+a。

（5）a%=(b%=2)，b 的值等于 3。

（6）a+=a-=a*=a。

4．编程输入一个华氏温度，要求输出摄氏温度。公式为 c=5/9(f-32)。

5．编程从键盘输入圆的半径 r，求以 r 为半径的圆的周长、面积、圆球体积，并输出结果。