第 2 章　数据类型及表达式

本章导读

C 语言程序的基本操作对象是数据。数据以某种特定类型存在，如整型、实型、字符型等。数据类型规定了数据的表示形式、取值范围、存储格式和相关的运算方式。数据往往以常量或变量的形式存在，它们又分别属于各种数据类型。C 程序中的每个数据都具备类型与数值两个属性。本章通过 C 程序实例分析，使读者理解 C 语言的基本数据类型的表示形式、存储格式及相关的运算，掌握变量定义的方法及意义，能够灵活运用各种表达式，为 C 语言编程打下基础。本章学习的主要内容为:

- 基本的数据类型
- 常量和变量
- C 语言各种运算符和表达式的应用

2.1　关于数据类型及表达式的 C 程序实例

数据在内存中是以二进制形式存放的，不同的数据类型在内存中存储的形式各不相同。下面介绍两个 C 程序，说明 C 语言数据类型及表达式的相关知识。

【例 2.1】关于常量、变量及数据类型说明的程序实例。

```
/*程序名为 l2_1.cpp*/
#include"stdio.h"
void main()
{   int a=3;                          /*定义整型变量 a 并赋予常量值 3 */
    short r=6;                        /*定义短整型变量 r 并赋予常量值 6*/
    char c='a';                       /*定义字符变量 c 并赋予字符常量值'a' */
    float pi=3.14;                    /*定义单精度实型变量 pi 并赋予实型常量值 3.14 */
    double s;                         /*定义双精度实型变量 s */
    s=pi*r*r;                         /*将 pi*r*r 值赋给 s */
    printf("int:%d,short:%d,char:%d,float:%d,double:%d\n",sizeof(a),
    sizeof(r),sizeof(c),sizeof(pi),sizeof(s));
                                      /*输出各变量占内存的字节数  */
    printf("a=%d,r=%d,c=%c,pi=%f,s=%lf\n",a,r,c,pi,s);
                                      /*输出各变量的值  */
}
```

【例 2.2】关于运算符和表达式的程序实例。

```
/*程序名为 l2_2.cpp*/
#include"stdio.h"
void main()
{   int i=3,a=5,b=7,c,d;                  /*定义变量并赋初值  */
    printf("i=%d\n",i++);                 /*输出 i++的值  */
```

```
    printf("i=%d\n",i);                    /*输出 i 的值 */
    c=a>b?a:b;                             /*计算表达式 c=a>b?a:b */
    printf("c=%d\n",c);                    /*输出 c 的值 */
    d=a<=b&&i;                             /*计算表达式 d=a<=b&&i */
    printf("d=%d\n",d);                    /*输出 d 的值 */
    d=(c,d=c);                             /*计算表达式 d=(c,d=c) */
    printf("d=%d\n",d);                    /*输出 d 的值 */
}
```

【例 2.1】程序实例运行结果如图 2-1 所示。

图 2-1 【例 2.1】程序输出窗口

可见变量可以被定义成不同的数据类型并赋值，不同类型的数据在内存中所占的字节数不同，如短整型、整型、字符型、单精度实型和双精度实型在内存中分别占 2 字节、4 字节、1 字节、4 字节和 8 字节。同时，这些数据在输出时所使用的输出格式控制符也不相同，分别为%d、%d、%c、%f 和%lf 形式。由此可见，数据表现为常量或变量，数据又具有不同的数据类型，不同数据类型的数据存储格式不同，数据的输出格式控制也不相同。

注意：*在 Turbo C 环境下，整型数据存储时在内存中占两个字节。*

【例 2.2】程序实例运行结果如图 2-2 所示。

图 2-2 【例 2.2】程序输出窗口

由结果 i=3 i=4 可知 printf("i=%d\n",i++);中 i++的值仍然为 3，而 i 的值由 3 增 1 后变为 4，理解自增运算符++的运算方式。

c=a>b?a:b;语句中包括关系运算符“>”、条件运算符“?:”、赋值运算符“=”，经过各种运算处理后 c 的值为 7。

d=a<=b&&i; 语句中包括关系运算符“<=”、逻辑运算符“&&”、赋值运算符“=”，经过各种运算处理后 d 的值为 1。

d=(c,d=c); 语句中包括赋值运算符“=”、逗号运算符“,”，经过各种运算处理后 d 的值为 7。

2.2　C 语言的数据类型

根据数据的取值范围、运算属性及存储方式等，将数据分成不同的数据类型。C 语言中的数据类型十分丰富，可分为基本类型、构造类型、指针类型和空类型，如图 2-3 所示。本节重点介绍最常用的整型、字符型和实型等基本数据类型，其他数据类型在以后的章节中介绍。

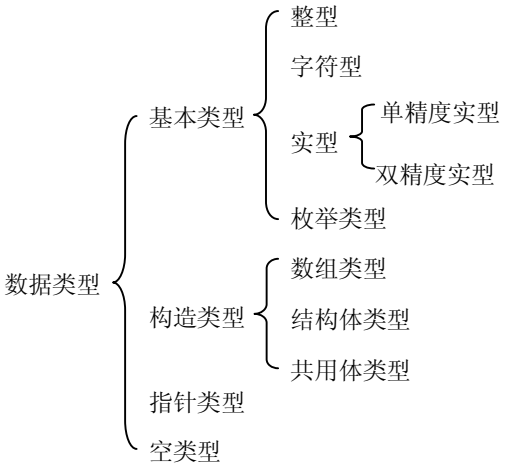

图 2-3　C 语言的数据类型

在 C 语言程序中所用到的数据都必须指明一定的数据类型，以便系统为其分配相应的存储空间，才能使用该数据进行相关运算。

2.2.1　基本数据类型

基本数据类型是语言系统预定义的数据类型，用户可以直接引用。在 C 语言中常用的基本数据类型有整型、实型和字符型。数据在内存中是以二进制形式存放的，不同类型数据存储时所占的内容空间和存储格式不同，取值范围也各不相同。

1. 整型数据

整型数据分为一般整型（int）、短整型（short）和长整型（long），并且每一种类型又分为带符号（signed）和无符号（unsigned）两种类型。具体情况如表 2-1 所示。

表 2-1　整型数据的特性

数据类型名	说明	Visual C++ 6.0 环境		Turbo C 2.0 环境	
		字节	取值范围	字节	取值范围
short [int]	短整型	2	-32768~32767	2	-32768~32767
signed short [int]	带符号短整型	2	-32768~32767	2	-32768~32767
unsigned short [int]	无符号短整型	2	0~65535	2	0~65535
int	整型	4	-2147483648~2147483647	2	-32768~32767
signed [int]	带符号整型	4	-2147483648~2147483647	2	-32768~32767
unsigned [int]	无符号整型	4	0~4294967295	2	0~65535

续表

数据类型名	说明	Visual C++ 6.0 环境		Turbo C 2.0 环境	
		字节	取值范围	字节	取值范围
long [int]	长整型	4	-2147483648~2147483647	4	-2147483648~2147483647
signed long [int]	带符号长整型	4	-2147483648~2147483647	4	-2147483648~2147483647
unsigned long [int]	无符号长整型	4	0~4294967295	4	0~4294967295

注意：用[]括起来的关键字内容在程序中可以省略。

在【例 2.1】中，由于“int a=3; short r=6;”定义 a 和 r 分别为整型和短整型，则“printf ("int:%d,short:%d,char:%d,float:%d,double:%d\n",sizeof(a), sizeof(r),sizeof(c), sizeof(pi),sizeof(s));”中输出 a 和 r 的类型长度分别为 4 和 2。

2. 实型数据

实型数据分为单精度实型（float）、双精度实型（double）和长双精度实型（long double）。具体情况如表 2-2 所示。

表 2-2 实型数据的特性

数据类型名	说明	Visual C++ 6.0 环境			Turbo C 2.0 环境		
		字节	取值范围	有效数字	字节	取值范围	有效数字
float	单精度实型	4	-3.4×10^{38}~3.4×10^{38}	6~7	4	-3.4×10^{38}~3.4×10^{38}	6~7
double	双精度实型	8	-1.7×10^{308}~1.7×10^{308}	15~16	8	-1.7×10^{308}~1.7×10^{308}	15~16
long double	长双精度实型	16	-1.2×10^{4932}~1.2×10^{4932}	18~19	10	-1.2×10^{4932}~1.2×10^{4932}	18~19

在【例 2.1】中，由于“float pi=3.14; double s;”定义 pi 和 s 分别为单精度实型和双精度实型，则“printf("int:%d,short:%d,char:%d,float:%d,double:%d\n",sizeof(a), sizeof(r), sizeof(c), sizeof(pi),sizeof(s));”中输出 pi 和 s 的类型长度分别为 4 和 8。

3. 字符型数据

字符型数据分为一般字符型（char）、带符号字符型（signed char）和无符号字符型（unsigned char）。具体情况如表 2-3 所示。

在【例 2.1】中，由于“char c='a';”定义 c 为字符型，则“printf("int:%d,short:%d, char:%d, float:%d,double:%d\n",sizeof(a), sizeof(r), sizeof(c), sizeof(pi),sizeof(s));”中输出 c 的类型长度为 1。

表 2-3 字符型数据的特性

数据类型名	说明	字节	取值范围
char	字符型	1	-128~127
signed char	带符号字符型	1	-128~127
unsigned char	无符号字符型	1	0~255

2.2.2 常量

常量是指在程序运行过程中其值不能被改变的量。常量往往由书写形式决定其类型和值。

如 12、-3.4、'a'等。在 C 语言中，常量主要有整型、实型、字符型和字符串常量四种。为增加程序的可读性，C 语言还使用了另一种形式的常量，用一个标识符代表一个常量，称为符号常量，有关内容将在第 5 章中介绍。

1. 整型常量

在 C 语言中，整型常量有十进制、八进制、十六进制三种进制表示方法，并且各种数制均可有正（+）负（-）之分，正数的“+”可省略。

（1）十进制整型常量：以数字 1~9 开头，其他位以数字 0~9 构成十进制整型常量。如 12，-38 等。

（2）八进制整型常量：以数字 0 开头，其他位以数字 0~7 构成八进制整型常量。如 012，-037 等。

（3）十六进制整型常量：以 0X 或 0x 开头（数字 0 和大写或小写字母 x），其他位以数字 0~9 或字母 a~f 或 A~F 构成十六进制整型常量。如 0x12，-0Xa9 等。

如果在整型常量加上后缀 L 或 l 表示该常量为长整型常量，加上后缀 U 或 u 表示无符号整型常量。

2. 实型常量

实型常量又称浮型常量。实型常量由整数部分和小数部分组成，有两种表示形式：小数表示法和科学计数法。它只能用十进制表示。

（1）小数表示法。它是由数的符号、数字和小数点组成的实型常量（注意：必须有小数点）。如-2.5、3.0、4.、.34 等都是合法的实型小数形式。

（2）科学计数法。科学计数法也称指数法。它是由数的符号、尾数（整数或小数）、阶码标示（E 或 e）、阶符和整数阶码组成的实型常量。尾数不可缺省，阶码必须为整数。如-2.5E-3、3e5、34E-3 等都是合法的指数形式。如-2.5E-3 表示-2.5×10^{-3}。

实型常量分为单精度、双精度和长双精度三种类型。实型常量如果没有任何说明，则表示为双精度常量，实型常量后加上 F 或 f 则表示单精度常量，实型常量后加上 L 或 l 则表示长双精度常量。

3. 字符型常量

字符型常量是由一对单引号括起来的一个字符。它分为一般字符常量和转义字符。一个字符常量在计算机的存储中占据 1 个字节。

（1）一般字符常量。一般字符常量是用单引号括起来的一个普通字符，其值为该字符的 ASCII 代码值。ASCII 编码表见附录 1。如'a'、'A'、'0'、'?'等都是一般字符常量，但是'a'和'A'是不同的字符常量，'a'的值为 97，而'A'的值为 65。

（2）转义字符。C 语言允许用一种特殊形式的字符常量，它是以反斜杠（\）开头的特定字符序列，表示 ASCII 字符集中控制字符、某些用于功能定义的字符和其他字符。如“\n”表示回车换行符，“\\”表示字符“\”。常用的转义字符参见 1.4.1 节的内容。

4. 字符串常量

字符串常量也称字符串。它是由一对双引号（""）括起来的字符序列。字符序列中的字符个数称字符串长度，没有字符的字符串称为空串。如"a"、"This is a C program."、"12+3"等都是合法的字符串常量。

字符串常量中的字符是连续存储的，并在最后自动加上字符'\0'（空字符，该字符的 ASCII 码值为 0，也称 NULL 字符）作为字符串结束标志。如字符串"a"在计算机内存中占两个连续

单元，存储内容为字符'a'和'\0'。可见，字符串"a"与字符'a'的存储形式完全不同。

在【例 1.1】中 printf("This is a C program.");就是输出字符串"This is a C program."。

对于字符串常量和字符常量的区别是十分显著的，主要表现在以下方面：

（1）两者表示形式不同。字符常量以单引号表示，而字符串常量以双引号表示。

（2）两者存储时所占的内存空间不同。字符常量在内存中只占 1 个字节，也就是用 1 个字节存放该字符的 ASCII 码值。而字符串常量在内存中，除了存储串中的有效字符，每个字符占 1 个字节存放其 ASCII 码值外，系统还自动在串后加上 1 个字节，存放字符串结束标志'\0'。

（3）两者允许的操作不同。字符常量允许在一定范围内与整数进行加法或减法运算，如'a'-32 运算是合法的。而字符串常量不允许上述运算，如"a"-32 运算是非法的。

（4）存放两者的变量不同。字符常量可存放在字符变量或整型变量中，而字符串常量不能存放在字符变量或整型变量中，而需要存放在字符数组中。字符变量和字符数组将在后面介绍。例如：

```
char c='a';                /*将字符'a'存放到字符变量 c 中，是合法的*/
char c="a"                 /*将字符串"a"存放到字符变量 c 中，是不合法的*/
```

2.2.3 变量及其类型定义

变量是指在程序执行过程中其值可以被改变的量。变量有 3 个基本要素：变量名、变量数据类型和变量的值。在 C 语言中，任何一个变量在使用之前都必须首先定义它的名字，并说明它的数据类型。也就是说，变量使用前必须先定义，即指定变量名，说明变量数据类型。变量定义的实质是按照变量说明的数据类型为变量分配相应空间的存储单元，在该存储单元中存放变量的值。

C 语言中，变量使用时遵循“先定义，后使用”的原则。

1. 变量的定义

【例 2.1】程序实例中以下语句：

```
int a=3;                   /*定义整型变量 a 并赋予常量值 3 */
short r=6;                 /*定义短整型变量 r 并赋予常量值 6*/
char c='a';                /*定义字符变量 c 并赋予字符常量值'a' */
float pi=3.14;             /*定义单精度实型变量 pi 并赋予实型常量值 3.14 */
double s;                  /*定义双精度实型变量 s */
```

都是对变量的定义语句，从中可以总结出变量定义的一般格式为：

```
数据类型 变量名表;
```

说明：

（1）数据类型：是 C 语言的合法数据类型。如例中的 int、short、char、float、double 等。

（2）变量名表：变量名是 C 语言合法的标识符。变量名表可以包含多个变量名，彼此之间用逗号分开，表示同时定义若干个具有相同数据类型的变量，如：

```
char c1,c2;
```

定义了字符类型的变量 c1 和 c2。变量定义时还可以同时赋给变量一定的值，如【例 2.1】中“int a=3;”表示定义了整型变量 a，同时赋给变量 a 初值 3。

（3）变量定义语句可以出现在变量使用之前的任何位置。程序设计时不违背“先定义，后使用”的原则即可。

2. 变量初始化及赋值

在【例 2.1】中“int a=3;short r=6; char c='a'; float pi=3.14; ”的“=”是赋值运算符，用来给变量赋值。所以，以上 4 条语句都是在定义变量的同时给变量赋了值，称为变量的初始化。即变量初始化是指定义变量同时，给变量一个初始值。

在【例 2.1】中“s=pi*r*r;”语句是通过赋值运算符“=”将“pi*r*r”的值赋给变量 s，即该语句实现了给变量 s 的赋值。所以，变量值可以通过初始化取得，也可以在定义后，通过给变量赋值的方法取得。

2.3　运算符及表达式

2.3.1　运算符和表达式

1. 运算符

C 语言的运算符十分丰富，使用十分灵活，功能十分强大，除了程序控制语句和输入输出语句以外的几乎所有操作都可以由运算符处理实现。C 语言提供了 13 类，共计 34 种运算符，本节介绍几类运算符，其余的将在以后章节介绍。具体分类情况如表 2-4 所示。

表 2-4　运算符分类表

分类名称	运算符
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	*、&
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	函数运算符()

根据运算符的运算对象的个数，C 语言的运算符分为单目运算符、双目运算符和三目运算符。如++、--是单目运算符，+、-、*、/是双目运算符，？:是 C 语言中唯一的三目运算符。

2. 表达式

用运算符将操作对象连接起来、符合 C 语法规则的式子称为表达式。表达式因运算符种类也可分为各种表达式，如算术表达式、关系表达式、赋值表达式等，每一个表达式也都具有一定的值。

当表达式中有若干个运算符进行混合运算操作时，必须考虑各个运算符的优先级和结合性。

3. 运算符的优先级

运算符的优先级是指不同的运算符在表达式中进行运算的先后次序。各种运算符的优先级见本章小结。如，算术运算符*、/的优先级高于+、-的优先级。在【例 2.2】中“c=a>b?a:b;”语句中运算符按优先级由高到低排列顺序为“>” 、“?:”、“=”。因此，计算次序如图 2-4 所示。

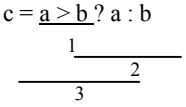

图 2-4 运算符优先级

4. 运算符的结合性

当一个运算对象两侧的运算符的优先级相同时，运算的结合方向称为结合性。运算符的结合性分为左结合和右结合两种。在 C 语言中，运算对象先与左面的运算符结合称为左结合，如+、-、*、/的结合方向为自左向右；运算对先与右面的运算符结合称为右结合，如单目运算符++、--的结合方向是自右向左。

2.3.2 算术运算符和算术表达式

1. 基本的算术运算符

基本算术运算符包括加法（+）、减法（-）、乘法（*）、除法（/）和求余（%）。后 3 种运算符优先级高于前两种，它们都是左结合性。

+ 加法运算符或正值运算符。如 3+7、+3。

- 减法运算符或负值运算符。如 3-7、-3。

* 乘法运算符。如 3*7。

/ 除法运算符。如 13/7。

% 求余运算符，也称取模运算符，运算对象均须为整型数据。如 13%7 的值为 6。

对于运算符/和%有如下说明：

（1）若除法运算符的运算对象均为整型数据，则结果为其商的整数部分，舍去小数部分。如 13/7 的结果为 1。若运算对象中有一个为负值，则舍入的方向是不固定的。如，-13/7 在有的机器上得到结果是-1，有的机器上得到结果是-2，但多数机器采取“向零取整”方法，即 13/7=1，-13/7=-1，取整后向零靠拢。

（2）求余运算符的运算对象必须是整型数据，运算结果的符号与被除数的符号相同。如：-13%7 运算结果为-6；13%(-7)运算结果为 6；-13%(-7)运算结果为-6。

2. 自增、自减运算符

自增（++）和自减（--）运算符是单目运算符，其功能是使运算对象（变量）的值增 1 或减 1。它们既可以作前缀运算符（位于运算对象的前面），如++i，--i；也可以作后缀运算符（位于运算对象的后面），如 i ++，i --。前缀和后缀运算的数据处理方法有明显的差异。对于前缀形式表示在用该表达式之前先使变量值增（减）1；对于后缀形式表示在用该表达式的值之后使 n 的值增（减）1。

++j,--j 在使用 j 之前，先使 j 的值加（减）1。

j++,j-- 在使用 j 之后，使 j 的值加（减）1。

在【例 2.2】中“int i=3;printf("i=%d\n",i++);”，i++是后缀形式，则先计算表达式的值，其值为当前 i 值，因此，输出 i=3，然后使 i 自增 1，i 值为 4。所以，语句“printf("i=%d\n",i);”输出的结果是 i=4。

再如，若 i 的原值等于 3，则执行下面的赋值语句，其结果不同。

（1）k=++i;　i 的值先增 1 变成 4，再赋给 k，k 的值为 4。

（2）k=i++;　先将 i 的值 3 赋给 k，k 的值为 3，再将 i 的值增 1 变成 4。

自增或自减运算符在使用时，需要注意以下几点：

（1）运算符的操作对象，只能是变量，而不能是常量或表达式。

（2）运算符的优先级高于基本算术运算符，结合性是“自右向左”右结合。如有-i++，遵照右结合原则，它相当于-(i ++)，而不相当于(-i)++形式。

它们的运算对象只能是整型变量、字符型变量和指针变量，而不能是其他类型的量。

2.3.3　关系运算符和关系表达式

关系运算符是对两个操作对象进行大小比较的运算符，是逻辑运算的一种简单形式。用关系运算符将两个表达式连接起来的符合 C 语法规则的式子称为关系表达式。关系表达式的运算结果是一个逻辑值，即“真”或“假”。在 C 语言中关系运算结果为真，以整数“1”表示，结果为假，以整数“0”表示。

C 语言中的关系运算符共有 6 种，它们是：

<　小于

<=　小于等于

>　大于

>=　大于等于

==　等于

!=　不等于

关系运算符的优先级低于算术运算符的优先级，且等于（==）和不等于（!=）的优先级低于另外 4 种运算符的优先级。关系运算符的结合性是左结合性。

在【例 2.2】实例“int a=5,b=7;”和“c=a>b?a:b; d=a<=b&&i;”中，a>b 值为 0，a<=b 值为 1。

关系运算符在应用中应特别注意关系运算符“==”和赋值运算符“=”的区别，前者是判断运算两边表达式的值是否相等，而后者是将“=”左边表达式的值赋给右边的变量。例如：

```
a=12==3;
```

先进行“==”比较运算，结果值为“假”，即为“0”，再进行“=”赋值运算，将“0”赋给变量 a，使 a 的值为 0。

2.3.4　逻辑运算符和逻辑表达式

逻辑运算符是对逻辑量进行操作的运算符，运算结果是逻辑值。用逻辑运算符将两个表达式连接起来的符合 C 语法规则的式子称为逻辑表达式。参与逻辑运算的逻辑量“真”或“假”的判断原则是：以 0 代表“假”，以非 0 代表“真”。即将一个非零的数值认作“真”，将零值认作“假”。逻辑运算的结果逻辑值只有两个：“真”和“假”，它们分别用“1”和“0”表示。

C 语言中提供的 3 种逻辑运算符及运算法则如表 2-5 所示。

表 2-5 逻辑运算符及运算法则

运算符	运算名称	运算法则	结合性
&&	逻辑与	当两个操作对象都为“真”时，运算结果为“真”，其他情况运算结果都为“假”	左结合
||	逻辑或	只有当两个操作对象都为“假”，运算结果才为“假”，其他情况运算结果都为“真”	左结合
!	逻辑非	当操作对象为“真”时，运算结果为“假”；当操作对象为“假”时，运算结果为“真”	右结合

逻辑运算符中“&&”和“||”低于关系运算符，“!”高于算术运算符。

在【例 2.2】程序实例中“d=a<=b&&i;”先计算“a<=b”为真，值为 1，再计算“1&&i”，两个操作对象均非零为“真”，故结果为“真”，值为 1。再将该值 1 赋给变量 d。

在处理逻辑表达式时应注意以下几点：

（1）C 语言编译系统在给出逻辑运算结果时，以 0 代表“假”，以 1 代表“真”。但在判断一个逻辑量“真假”时，以非 0 表示“真”，以 0 表示“假”。例如：

当 a=5.4,b=5,c='a'时，!a，!b，!c 的值均为“假”，即为 0。a&&c 的值为 1，因为 a 和 c 均为非 0。

（2）在进行逻辑运算时，逻辑表达式运算到其值完全确定时为止。例如：运算表达式(a=3)&&(a==5)&&(a=6)时，由于 a=3 之后运算 a==5 的值为假，所以就不再进行 a=6 的运算了，因此 a 的值仍为 3，而整个逻辑表达式的值为 0。

2.3.5 条件运算符和条件表达式

条件运算符是“?:”是 C 语言中唯一的三目运算符，用条件运算符将两个表达式连接起来的符合 C 语法规则的式子称为条件表达式。条件表达式的一般形式如下：

表达式 1？表达式 2：表达式 3

操作过程是：先计算表达式 1 的值，若为“真”，则计算表达式 2 的值，整个条件表达式的值就是表达式 2 的值；若表达式 1 的值为“假”，则计算表达式 3，整个条件表达式的值就是表达式 3 的值。

条件运算符优先级低于逻辑运算符，其结合性是右结合。

在【例 2.2】程序实例中“a=5,b=7;c=a>b?a:b;”，先计算 a>b 值为假，则条件表达式值取 b 的值为 7。因此，c 值为 7。

使用条件表达式时，应注意以下几点：

（1）条件运算符优先级高于赋值运算符。如 a=b>0?b:-b;相当于 a=(b>0?b:-b)，功能是将 b 的绝对值赋给 a。

（2）条件运算符结合性是右结合。如 b>0?1:b<0?-1:0;相当于 b>0?1:(b<0?-1:0)。

2.3.6 赋值运算符和赋值表达式

为了精炼程序，提高编译效率，C 语言的赋值运算符包括简单赋值运算符和复合赋值运算符，复合赋值运算符又包括算术复合赋值运算符和位复合赋值运算符（位运算在第 8 章介绍）。由赋值运算符将操作对象连接起来符合 C 语法规则的式子称为赋值表达式。

1. 简单赋值运算符及其表达式

赋值运算符是“=”，其作用是将赋值运算符右侧的表达式的值赋给其左侧的变量。在【例 2.2】程序实例中“i=3,a=5,b=7; c=a>b?a:b; d=a<=b&&I; d=(c,d=c);”都是将“=”右侧的数值或表达式的值赋给“=”左侧的变量。

赋值运算符的优先级低于条件运算符，其结合性是右结合。如：a=b=c=a>b?a:b;相当于a=(b=(c=(a>b?a:b)));。

值得注意的是：赋值运算符运算对象中的左侧对象一定是变量。如：a=b-c=5;相当于a=((b-c)=5);，由于表达式中出现将数值 5 赋给 b-c 表达式，因此，该表达式是非法的。

2. 复合赋值运算符及其表达式

C 语言允许在赋值运算符“=”之前加上其他运算符构成复合运算符。在“=”之前加上算术运算符则构成算术复合赋值运算符，如+=、-=、*=等；在“=”之前加上位运算符则构成位复合赋值运算符，算术复合赋值运算符的用法是：

+=　加赋值。如：a+=b-c 等价于 a=a+(b-c)。

-=　减赋值。如：a-=b-c 等价于 a=a-(b-c)。

=　乘赋值。如：a=b-c 等价于 a=a*(b-c)。

/=　除赋值。如：a/=b-c 等价于 a=a/(b-c)。

%=　取余赋值。如：a%=b-c 等价于 a=a%(b-c)。

位复合赋值运算符有<<=、>>=、&=、^=、!=五种，用法与算术复合赋值运算符类似。

在赋值运算中，当赋值运算符两侧的类型不一致，且都是数值型或字符型数据时，在赋值时要进行数据类型转换。

（1）“整型变量=实型数据”时，舍弃实数的小数部分。如 a 为整型变量，执行“a＝45.89;”的结果是使 a 的值为 45。

（2）“实型变量＝整型数据”时，数据值不变，但以浮点数形式存储到变量中。

（3）“单精实型变量=双精实型数据”时，截取其前面 7 位有效数字，存放到单精实型变量中。

（4）“整型变量=字符型数据”时，将字符数据（8 位）放到整型变量的低 8 位中。

（5）“字符型变量=整型数据”时，只截取其低 8 位原样送到 char 型变量中。

还有一些其他情况，如带符号整型数据赋给长整型变量及无符号整型数据赋给长整型变量等，这里不再赘述。但不同类型的整型数据间的赋值都是按存储单元中的存储形式直接传送。

2.3.7　逗号运算符和逗号表达式

逗号运算符是“,”，其优先级低于赋值运算符，是左结合性。用逗号运算符将若干个表达式连接成一个逗号表达式，它的一般形式如下：

表达式 1,表达式 2,……,表达式 n

逗号表达式的操作过程是：先计算表达式 1，再计算表达式 2，……，最后计算表达式 n，而逗号表达式的值为最右边表达式 n 的值。如：

a=4.5,b=6.4,34.5-20.1,a-b

这是一个逗号运算表达式，它由 4 个表达式结合而成，从左向右依次计算，逗号表达式的值为 a-b 的值，即-1.9。

在【例 2.2】程序实例中“d=(c,d=c);”是将逗号表达式 c,d=c 的值赋给变量 d。

值得注意的是，逗号运算符在 C 语言所有运算符中优先级最低，在进行运算时应给予注意。如：a=10,20;不同于 a=(10,20);，前者 a 的值为 10，表达式的值为 20，后者 a 的值为 20，表达式的值也为 20。

2.3.8 求字节运算符

求字节运算符是 sizeof，用于计算变量或某种类型的量在内存中所占的字节数。用法有两种：

1. sizeof 表达式

功能是计算出表达式计算结果所占用内存的字节数。在【例 2.1】程序实例中“printf("int:%d, short:%d,char:%d,float:%d,double:%d\n",sizeof(a),sizeof(r),sizeof(c),sizeof(pi),sizeof(s)); ”就是输入变量 a,r,c,pi,s 所占的内存字节数。

2. sizeof（类型名）

功能是计算出某种类型的量存储时所占用内存的字节数。如：sizeof(float)计算单精度实型数据在内存中所占的字节数，结果为 4。

2.3.9 类型转换

在 C 语言中，整型、单精度、双精度及字符型数据可以进行混合运算。当表达式中的数据类型不一致时，首先转换为同一类型，然后再进行运算。C 语言有两种方法实现类型转换：自动类型转换和强制类型转换。

1. 自动类型转换

在表达式中，双目运算符两侧的操作数要求类型一致，不同类型的数据要进行类型转换，这是由 C 编译系统自动完成的，转换的原则是低类型转换为高类型。如：

```
int a=9;
float b=9.7;
```

若计算机 a*b 时，则系统自动将低类型的 int 型变量 a 转换成高类型变量 b 的 float 类型，再进行计算。各种数据类型的高低及自动转换顺序如图 2-5 所示。

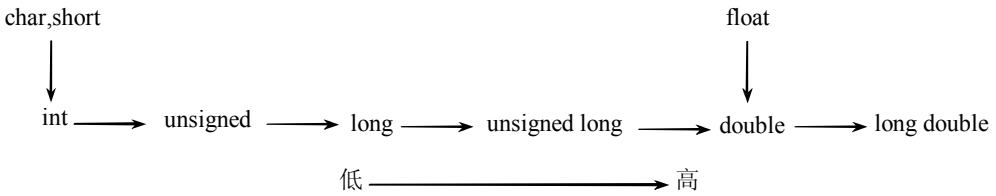

图 2-5 类型自动转换规则

2. 强制类型转换

强制类型转换是指通过强制类型转换运算符，将表达式的类型强制转换为所指定的类型。强制类型转换的一般形式为：

(数据类型)操作数

或

数据类型(操作数)

其中，数据类型是要转换成的数据类型，操作数是要转换的对象，可以是变量名和表达

式。功能是将操作数的值强制转换成指定的数据类型。如：

int(a)是将变量 a 转换成 int 型数据。

int(120.4*7)是将 120.4*7 值转换成 int 型数据，表达式的值为 842。

float(a+b)是将 a+b 的值转换成 float 类型。

double(120%7)是将 120%7 的值转换成 double 类型，表达式的值为 1.000000。

值得注意的是，数据类型转换是对操作数的值进行转换，并不改变操作数中变量本身的数据类型。如：

```
int a;          /*定义变量 a 为整型数据*/
float c;        /*定义变量 c 为单精度实型*/
c=float(a);     /*将变量 a 的值转换为单精度并赋值给变量 c，而变量 a 仍为整型*/
```

在两种类型转换过程中，当数据类型由低向高转换时，数据精度不会受到损失；而数据类型由高到低转换时，数据精度会受到损失。

本章小结

本章通过 C 程序实例分析，介绍了 C 语言的数据类型和丰富的运算符及表达式，是后续学习程序的基础。

C 语言的数据类型分为四大类，即基本类型、构造类型、指针类型和空类型。本节重点介绍了基本类型，其他类型将在后续章节中介绍。基本数据类型包括整型、实型、字符型和枚举类型。整型数据包括短整型（short int，2 字节）、一般整型（int，4 字节，Turbo C 下占 2 字节）和长整型（long int，4 字节），每一种整型数据又包括带符号（signed）和无符号（unsigned）两类数据。实型数据包括单精度实型（float，4 字节）、双精度实型（double，8 字节）、长双精度实型（long double，16 字节，Turbo C 下占 10 字节）。字符型数据（char）占 1 个字节，存储时，内存中存放该字符的 ASCII 值。数据又以常量和变量的形式存在。

变量使用时必须遵循“先定义，后使用”的原则。变量定义的实质是按照变量说明的数据类型为变量分配相应空间的存储单元，在该存储单元中存放变量的值。

变量定义的一般格式为：

```
数据类型 变量名表;
```

C 语言的运算符十分丰富，提供了 13 类，共计 34 种运算符。各种运算符优先级及结合性总结如表 2-6 所示。

表 2-6　运算符的优先级和结合性

优先级	运算符	名称	结合方向
1	() [ ] -> .	圆括号 下标运算符 指向结构成员运算符 结构成员运算符	自左向右
2	! ~ ++	逻辑非运算符 按位取反运算符 自增 1 运算符	自右向左

续表

优先级	运算符	名称	结合方向
2	-- - （类型） * & sizeof	自减 1 运算符 负号运算符 类型转换运算符 间接访问运算符 取地址运算符 求字节运算符	自右向左
3	* / %	乘法运算符 除法运算符 取模运算符	自左向右
4	+ -	加法运算符 减法运算符	自左向右
5	<< >>	左移运算符 右移运算符	自左向右
6	<、<=、>、>=	关系运算符	自左向右
7	== !=	等于运算符 不等于运算符	自左向右
8	&	按位与运算符	自左向右
9	^	按位异或运算符	自左向右
10	|	按位或运算符	自左向右
11	&&	逻辑与运算符	自左向右
12	||	逻辑或运算符	自左向右
13	? :	条件运算符	自右向左
14	=、+=、-、-=、*=、/=、%=、>>=、<<=、&=、^=、|=	赋值运算符	自右向左
15	,	逗号运算符	自左向右

习题二

一、思考题

1．C 语言为什么对变量使用时规定“先定义，后使用”的原则？

2．C 语言中自动数据类型转换的规则是什么？

3．指出下列合法的常量，并说明其类型。

123　3.　.45　1.4　8L　123U　‘0’　‘\0’　‘00’　“00”　057　059　0xff　0X12A　3.2E-5　3.45e3　3e4.5　‘\xab’　‘\018’　‘\016’　‘\\’

4．指出下列不合法的 C 语言表达式，并说明原因。

++7　x+y,a,b　a!=b=c+1　a<b<c　k--　k=++k　++k=a　b=c+d=a　a=b,b=c,c=a

二、选择题

1．在 C 语言中，Turbo C 下一个整型数据占（　）字节，Visual C++下一个整型数据占（　）字节。

A）2　2　　B）2　4　　C）4　2　　D）4　4

2．以下运算符中，运算优先级最高的是（　）。

A）! =　　B）=　　C）+.　　D）++

3．以下运算符中，运算优先级最低的是（　）。

A）？：　　B）!=　　C）&&　　D）+=

4．以下运算符中，其操作对象只能是整型数据的是（　）。

A）/　　B）++　　C）%　　D）&&

5．在 C 语言中，关系表达式的值是（　）。

A）0　　B）1　　C）0 或 1　　D）以上都不对

6．在 C 语言中，逻辑值“真”用（　）表示。

A）0　　B）1　　C）非零值　　D）非零整数

7．若有“int a;float b;char c;double d;”说明后，表达式 a+b*c/d 的数据类型是（　）。

A）int　　B）float　　C）char　　D）double

8．sizeof(double)的值是（　）。

A）0　　B）2　　C）4　　D）8

9．不能表示 x 为偶数的表达式是（　）。

A）x%2==0　　B）!(x%2)　　C）x==x/2*2　　D）!x%2

10．若有“int a=5,b=4,c=2;”说明后，则表达式 a>b>c 的值为（　）。

A）1　　B）0　　C）编译出错　　D）2

11．若有“int x=1,y=1,z=1||z++”语句，执行后 x，y，z 的值分别为（　）。

A）2　0　2　　B）1　0　1　　C）2　0　1　　D）1　0　2

12．不能正确表示数学关系 0<a<10 的 C 语言表达式是（　）。

A）a>0&&a<10　　B）!(a<=0)&&!(a>=10)

C）0<a<10　　D）0<a&&a<10

13．设 a=5，b=15，c=7，d=17，x=20，y=2，执行(x=a>b)&&(y=c>d)后 y 的值为（　）。

A）1　　B）2　　C）0　　D）7

14．设有“int x=10;x*=3+x%(-4);”语句，执行该语句后 x 的值是（　）。

A）10　　B）32　　C）28　　D）50

15．设有“int c=3;int a;a=2+(c+=c++,c+15,++c);”，执行语句后 a=（　）。

A）6　　B）10　　C）25　　D）12

三、填空题

1．设 a=10，b=20，则表达式!a<b 的值是_____。

2．在<=、==、!、?:、&&、*运算符中优先级最高的是_____，优先级最低的是_____，右结合性的是_____和_____。

3．设有“int x=2,y=3,z;z=(++x>=y--)?((--y==1)?x--:--y):y--;”，则 z 的值是_____。

项目实训

一、项目描述

运用本章所学内容，领会 C 语言基本数据类型、运算符与表达式的概念，掌握相关知识，进一步熟练掌握 C 程序的上机调试运行过程。

编程计算(float)(++a+b)/2+(x+y)%2/4+b++的值。其中，a=3,b=4,x=7,y=8。

二、项目要求

本项目实训目标是了解 C 语言数据类型、运算符与表达式的相关知识。警示学生运算符的优先级和结合性的重要性。要求能够独立完成并输出正确的程序结果。

1．编写相应程序。

2．根据程序运行的结果分析程序的正确性。