

第三章-数据类型、运算符与表达式

计算机数据的存储与表示

在计算机内部存储数据常用单位有位和字节。其中，位（bit）是计算机存储数据的最小单位，计算机中最基本最直接的操作就是对二进制位的操作。字节（byte），简写为B，通常一个字节由8个二进制位组成，即1 byte = 8 bit，一个字节可以存放一个ASCII码。

整数的二进制表示

整数，也称为整型，分为无符号和有符号两种。有符号的整数包含正数和负数，正负用最高位的0和1来区分。

一般用4个字节来表示整数，能表示的有符号数的范围是：-2147483648 ~ 2147483647。无符号数比有符号数扩大一倍。

浮点型数据的二进制表示

浮点型数据在计算机中存储时首先将十进制数转换成二进制数之后，将整数部分与小数部分分开保存。

单精度浮点型

4个字节表示

双精度浮点型

8个字节表示

C语言的数据类型与取值范围

C语言的数据类型包括：

基本类型、构造类型、指针类型、空类型void、定义类型typedef

不同数据类型的取值范围：

基本数据类型

- (1) 整型：用int表示，包括短整型（short或short int）、整型（int）、长整型（long 或 long int）
- (2) 浮点型：包括单精度浮点型（float）和双精度浮点型（double，即long float）

float)

(3) 字符型：用char表示。

各数据类型所占长度

int (4) 、 short (2) 、 long (4) 、 float (4) 、 double (8) 、 char (1)

常量与变量

常量和符号常量

1. 标识符：用来标识变量名、标号、函数名以及其他各种用户定义名等等的字符序列。

标识符定义规则：

- (1) 只能有数字、字母、下划线组成
- (2) 不能以数字开头
- (3) 不能使用关键字
- (4) 长度不超过32字符
- (5) 区分大小写

2. 常量：程序运行时其值不能改变的量。

(1) 整型常量：表示的常量是整数。

》》 有以下三种表达形式：

整型常量	进制	十进制数
17	十进制	17
017	八进制	15
0X17	十六进制	23
17 L或17l	十进制	17
17LU 或17lu	十进制	17

注：U (unsigned) 、 L(long)

十进制整数：由数字0-9和正负号表示

八进制整数：由数字0开头，后跟0-7表示

十六进制数：由0X或0x开头，后跟0-9、a~f、A~F表示

八进制与十六进制一般只用于表示无符号数

(2) 实型常量：表示的常量是实数

一般由两种表示方法：浮点计数法和科学计数法

》》 浮点计数法：通常表示成十进制数形式，必须有小数点

// // 浮点常量：通常用小写字母表示，也可以用科学计数法表示

》》科学计数法：通常表示成指数形式：其中e或E前面必须有数字，指数必须是整数

(3) 字符常量：用 (' ') 括起来的单个普通字符或转义字符就是字符常量

注：单引号中的字符不可以只有单引号和反斜杠，反斜杠本身就是一个转义字符

(4) 字符串常量：用 (" ") 括起来的字符序列或一串字符称为字符串常量。

C语言中，字符串常量在内存中存储时，系统自动在字符串的末尾加一个“串结束标志”，即ASCII值为0的字符NULL，用\0表示。因此字符串的存储在空间上实际占用了字符串长度n + 1个字节长度。但实际字符串长度任然是n。

(5) 符号常量：将程序中的常量定义为标识符，称为符号常量。

符号常量应用前应该先定义：

define 标识符 常量

注：后面没有分号

变量：在程序运行过程中其值可变的量称为变量

变量在内存中占据一定的存储单元（占据存储单元的大小由数据类型决定），在该存储单元中存放变量的值。定义格式如下：

数据类型名 变量1[, 变量2, 变量3,, 变量n];

e.g. int i;

double a, b, c;

C语言的运算符

算数运算符和算数表达式

(1) < ++ >、< -- >：自增和自减。在变量之前，先增（减）再执行语句；在变量之后，先执行语句，再增（减）。

(2) < * >、< \ >、< % >：乘、除、取模（取余数）

》》两整数相除，遵循向零取整原则。

》》% 运算符要求两侧必须是整数，不能是浮点数

(3) < + >、< - >：加减

赋值运算符和赋值表达式

赋值运算符用 “ = ” 表示，其左边是一个变量（左值），不能是常数或者变量表达式，其右可以是任意表达式（右值），格式为：

变量标识符 = 表达式;

类型转换

对于赋值运算符，如果类型不匹配，首先会按照编译环境设定的数据类型进行自动转换，无法实现自动转换的时候需要强制转换。如果程序中没有强制转换，则编译器报错。

类型转换规则：

》》 (1) 浮点型-> 整型： 向零取整

》》 (2) 整型 -> 浮点型： 数值不变，在证书后面添0

》》 (3) double -> float： 不出现溢出的前提下，截取前7位数字放到float的存储单元

》》 (4) 将一个int、short、long型数据赋值给char型变量时，把对应的int、short、long型变量的低8位，赋值给char类型就可以了。

》》 (5) 将一个带符号的int赋值给long，对应值不变，存储字节扩展；将一个long赋值给int只赋值低16位即可。int和long型数据在无符号时可以直接赋值。

复合运算符

```
+=、-=、*=、/=、%=、<<=、>>=、&=、^=、|=
```

注：复合运算符两个运算符之间不可以有空格。

关系运算符和关系表达式

<, <=, ==, >=, >, !=

运算结果返回：true false

逻辑运算符和逻辑表达式

! —— 逻辑非

&& —— 逻辑与

|| —— 逻辑或

注：

》》 (1) C语言中，零值表示假，其他任何非零值表示真

》》 (2) 逻辑运算符短路情况

(表达式1) && (表达式2)

一旦表达式1是假，那么即使表达式2不用执行也可以判断上述语句为假，因此C语言在这种情况下不会执行表达式2，此称为短路。

在这种情况下不会执行表达式4，此称为短路与

(表达式3) || (表达式4)

一旦表达式3为真，那么即使表达式4不用执行也可以判断上述语句为真，因此C语言在这种情况下不会执行表达式4，此称为短路或

条件运算符和条件表达式

格式：

(判断语句) ? (表达式1) : (表达式2) ;

如果判断语句为真，则执行表达式1，否则执行表达式2

注：

条件语句可以嵌套使用

所有语句类型可以不相同

位运算符

1. 原码与补码

- 》》 (1) 原码：一个十进制数转换成的二进制数本身就是这个数的原码
- 》》 (2) 补码：整数的补码是原码本身；负数的补码是其正整数原码取反加一
- 》》 (3) 补码转换成原码：
- 》》》 正：原码 = 补码
- 》》》 负：原码 = 补码 - 1 再取反然后加上符号

2. 位运算符

- (1) &：按位与-二进制位进行与运算。原则：全1则1
- (2) |：按位或-二进制位进行或运算。原则：有1则1
- (3) ^：按位异或-二进制位异或运算。原则：同0异1
- (4) ~：按位取反-二进制位取反运算。原则：01互换
- (5) <<：位左移-二进制位左移运算。原则：二进制位左移n位，右补0
- (6) >>：位右移-二进制位右移运算。原则：二进制位右移n位，低位舍弃，无符号数高位补0，负数不确定。

3. 位运算的应用

- (1) 整数各位清零——按位与 (&) 0
- (2) 整数高位屏蔽——按位与 (&) 屏蔽位&0，保留为全部&1
- (3) 整数奇数位置1——按位或 (|) 01010101，使奇数位一定有1
- (4) 整数取反——按位异或 (^) 11111111

数值类型数据间的混合运算

(1) 自动类型转换

》》 不同数据类型运算时，先自动转换成同一类型，较短的数据类型向较长的数据类型

型转换。

(2) 强制类型转换

》》一般格式： （类型名） （表达式或变量）