

# 基本运算

■ ■ DATA TYPES AND EXPRESSIONS  
2 数据类型与表达式

## 何谓运算符和操作数?

Example:

$W + Z$

操作数  
(Operand)

运算符  
(Operator)

操作数  
(Operand)



## 2.4 基本运算

算术运算

关系运算

逻辑运算

地址运算

按位运算

## 2.4.1 算术运算

### 1. 双目运算

加	$a+b$	$10+5=15$	
减	$a-b$	$10-5=5$	
乘	$a*b$	$10*5=50$	
除	$a/b$	$30/4=7;$	$30.0/4=7.5$
取余	$a\%b$	$14\%3=2;$	$25\%8=1$

## 2.4.1 算术运算

### 2. 单目运算

自增             $a++$  或  $++a$         等效于         $a=a+1;$

自减             $a--$  或  $--a$         等效于         $a=a-1;$

取负             $-a$

**注意：**单目运算的“前置”和“后置”的区别

“前置”        变量先增值(或先减值),后被引用

“后置”        变量先被引用,后再增值(或后减值)



➤ 已知

```
int x, y, i=25;
```

则 `x=--i;` /\* 相当于: `i=i-1; x=i; */`

```
y=i;
```

结果是: `x=24, y=24;`

➤ 已知 `int x, y, i=25;`

则 `x=i--;` /\* 相当于: `x=i; i=i-1; */`

```
y=i;
```

结果是: `x=25, y=24;`

## 2.4.2 赋值运算

表示形式:

**<变量> = <表达式> ;**

表达式是指由各种运算符把常量、变量、函数等运算对象连接起来的具有实际意义并符合C语法规则的式子。赋值是指将表达式的值赋给一个变量。

例如：

`x=5*2+8 ; x=18 ; v=volume(x , y , z ) ;`

赋值表达式可以嵌套

`x=(y=(z=100));` 相当于 `x=100; y=100; z=100;`

自动将右边表达式的数据类型转换成左边变量的类型

可以构成复合运算符，例如：

加赋值	$a += b$	等效于	$a = a + b$
-----	----------	-----	-------------

减赋值	$a -= b$	等效于	$a = a - b$
-----	----------	-----	-------------

乘赋值	$a *= b$	等效于	$a = a * b$
-----	----------	-----	-------------

除赋值	$a /= b$	等效于	$a = a / b$
-----	----------	-----	-------------

取余赋值	$a \% = b$	等效于	$a = a \% b$
------	------------	-----	--------------

注意：设  $x = 20$ ；

$x * = 10 + 2$ ； $x = ?$

等效于  $x = x * (10 + 2)$

算术运算符优先级高于算术赋值运算符





可以构成复合运算符，例如：

设  $x=20$  ;  $x*=10+2$  ;  $x=?$

等效于  $x=x*(10+2)$

等效于  $x=x*10+2$

算术运算符优先级高于算术赋值运算符

## 2.4.3 关系运算与逻辑运算

关系运算符：

大于	$a > b$	大于等于	$a \geq b$
小于	$a < b$	小于等于	$a \leq b$
等于	$a == b$	不等于	$a \neq b$

关系表达式的值为真，用 1 或非零数值表示，  
关系表达式的值为假用零表示。

编程：输入2个整数，求这2个数的最大数

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("输入2个整数：");
    scanf_s("%d%d", &a, &b);
    if (a > b) printf("%d与%d的最大数是%d\n", a, b, a);
    else printf("%d与%d的最大数是%d\n", a, b, b);
}
```

### 三种逻辑运算:

逻辑与	$a \& b$
逻辑或	$a \parallel b$
逻辑非	$!a$

- 同样, 逻辑真为非零, 逻辑假为零。
- 关系和逻辑运算用于比较两个数间特定关系的一种运算。常用于C的控制结构中。
- 关系和逻辑构成复合的逻辑关系。

编程：输入1个代表年份的正整数，判断是否闰年

闰年判断条件：

能够被400整除，或者能够被4整除但不能被100整除

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



## 编程：输入1个代

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int year;
    printf("input the year:");
    scanf_s("%d", &year);
    if (year % 400 == 0 || (year % 4 == 0 && year
    % 100 != 0))
    {
        printf("%d is leap year!", year);
    }
    else printf("%d is not leap year!", year);
    system("pause");
}
```



## 2.2.4 位运算

特有的位运算功能,可以完成汇编语言的一些工作.

**位逻辑运算:**

按位与  $a \& b$

按位或  $a | b$

按位异或  $a \wedge b$

按位反  $\sim a$

**移位运算:**

右移位  $a >> 4$

左移位  $a << 2$

逻辑运算：&&，||，！



## 2.2.4 位运算

特有的位运算功能,可以完成汇编语言的一些工作.

### 位逻辑运算:

按位与       $a \& b$

$a=01, b=10$

按位或       $a | b$

按位异或    $a \wedge b$

按位反       $\sim a$





设  $x=0x55(01010101)$   
 $y=0x5a(01011010)$

### 1. 按位与运算

规则:两个运算量中对应位都是1,则该位的结果值为1,否则为零.

### 2. 按位或运算

规则:两个运算量中对应位只要有一个为1,则该位的结果值为1,否则为零.

$x=01010101$   
 $y=01011010$   
 $\& \text{—————}$   
 $01010000 \quad (0x50)$

$x=01010101$   
 $y=01011010$   
 $| \text{—————}$   
 $01011111 \quad (0x5f)$



### 3. 按位异或运算

规则:两个运算量中对应位不同,则该位的结果值为1,否则为0。

### 4. 按位反运算

规则:一个运算量的每一位都取反。

$$\begin{array}{r} x=01010101 \\ y=01011010 \\ \wedge \text{-----} \\ 00001111 \text{ (0x0f)} \end{array}$$

$$\begin{array}{r} x=01010101 \\ \sim \text{-----} \\ 10101010 \text{ (0xaa)} \end{array}$$

## 5. 右移位运算

变量名  $\gg$  右移位数

每右移1位相当于除以2

每右移n位相当于除以2的n次幂

## 6. 左移位运算

变量名  $\ll$  左移位数

每左移1位相当于乘以2,

每左移n位相当于乘以2的n次幂

注意: 在移位时,变量一端被移出弃,另一端空出的位一般则以0填充

编程：

李丽设计了一个扫地机器人，可以完成扫地，拖地，吸尘等功能。已知扫地机器人控制变量 $x=01101111$ ，其三个功能是分别对第2,4,6位置做如下操作：清0，置1，取反。

请为李丽实现程序控制代码。

扫地：第2位清0

$$\begin{array}{r} 01101111 \\ \& 00000000 \\ \hline 00000000 \end{array}$$



编程：

李丽设计了一个扫地机器人，可以完成扫地，拖地，吸尘等功能。已知扫地机器人控制变量 $x=01101111$ ，其三个功能是分别对第2,4,6位置做如下操作：清0，置1，取反。

请为李丽实现程序控制代码。

扫地：第2位清0

~~$$\begin{array}{r} 01101111 \\ \& 00000000 \\ \hline 00000000 \end{array}$$~~

编程：

李丽设计了一个扫地机器人，可以完成扫地，拖地，吸尘等功能。已知扫地机器人控制变量 $x=01101111$ ，其三个功能是分别对第2,4,6位置做如下操作：清0，置1，取反。

请为李丽实现程序控制代码。

扫地：第2位清0

$$\begin{array}{r} 01101111 \\ \& 11111101 \\ \hline 01101101 \end{array}$$

编程：

李丽设计了一个扫地机器人，可以完成扫地，拖地，吸尘等功能。已知扫地机器人控制变量 $x=01101111$ ，其三个功能是分别对第2,4,6位置做如下操作：清0，置1，取反。

请为李丽实现程序控制代码。

```
x = x & ~2; //第2位清0(11111101)
```

```
x = x | 8; //第四位置1(00001000)
```

```
x = x ^ 32; //第6位取反
```





```
x = x & ~2; //第2位清0(11111101)
```

```
x = x | 8; //第四位置1(00001000)
```

```
x = x ^ 32; //第6位取反
```

操作之后的数据x怎么输出？

输出10进制？八进制？十六进制？二进制？





数字按照二进制输出

$X \& 01$  得到最低位

$(x >> 1) \& 01$  就把次低位  
移到最低位并取出最低位

.....

$(x >> 7) \& 01$  就把最高位  
移到最低位并取出

从最高位先输出



```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf("当前状态为 : ");
    char x=111 ,y;//x=01101111
    printf("开始扫地...\n");
    y = (x >= 1) & 01;
    printf("%d",y);
    x = x & ~2;//第2位清0(11111101)
    printf("结束扫地，开始拖地...\n");
    x = x | 8; //第四位置1(00001000)
    printf("结束拖地，开始吸尘...\n");
    x = x ^ 32; //第6位取反
```



## 2.4.5 其他运算

### 1. 三目条件运算

基本形式:  $e1 ? e2 : e3$

规则: 若 $e1$ 为真,则整个表达式结果值取 $e2$ ,否则整个表达式结果值取 $e3$ 。

例如 :

$a > b ? a : b$

取 $a, b$ 中比较大的数

$x > 0 ? 1 : (x < 0 ? -1 : 0)$

用 $1, -1, 0$ 表示变量 $x$ 是正数,负数,还是0  
实际上它是条件语句的简化形式,见第3章。

## 2.逗号结合运算

一般形式:

**表达式1,表达式2,...表达式n**

求解过程是: 先求解表达式1,再求解表达式2,  
最后求解表达式n,整个逗号表达式的值即为表达式n的值。

例如:  $a=2*5$ ,  $a/10$ ,  $a-2$  表达式值为8

$x=(a=10,b=100,c=1000);$  x的值为1000

逗号表达式常用于 循环语句中,可以分别对多个变量进行操作.  
详见第3章。

### 3. 长度运算

sizeof是取数据类型或变量的存储长度的运算符(以字节为单位)。

格式是：

sizeof(数据类型说明符或变量名)

例如: 定义 `double dx;`

则, `sizeof(dx)=8`                  `sizeof(double)=8`

可以了解不同编译程序中为不同类型的数据所分配的内存字节数.