



西北工业大学  
NORTHWESTERN POLYTECHNICAL UNIVERSITY

---

# C程序设计 Programming in C



**1011014**

---

主讲：姜学锋，计算机学院

## 赋值和类型转换

◆ 3、混合运算的风险控制

◆ 4、显式类型转换

## 2.5.1 隐式类型转换

---

- ▶ 4. 赋值运算中的隐式类型转换
- ▶ 如果赋值运算符左右两边的类型不一致，且都是数值型数据时，在赋值时需要进行隐式类型转换。
- ▶ 即先计算等号右侧表达式的值，转换成与左侧变量相同的类型，再赋值。

### 2.5.1 隐式类型转换

---

- ▶ (1) 将浮点型数据赋给整型变量时，舍弃浮点数的小数部分。如n是整型变量，n=6.18的结果是n的值为6。

### 2.5.1 隐式类型转换

---

- ▶ (2) 将整型数据赋给浮点型变量时，数值不变，但以浮点数形式存储到变量中。如78按78.0处理（根据浮点类型分别有7位或16位有效数字）。

### 2.5.1 隐式类型转换

---

- ▶ (3) 将double型数据赋给float变量时，截取前面7位有效数字存储到float变量。将float型数据赋给double变量时，数值不变，有效数字扩展到16位。

### 2.5.1 隐式类型转换

---

- ▶ (4) 将char型数据赋给short、int变量时，数据存储到变量的低8位，高位补0。将short、int型数据赋给字符型变量时，只将数据的低8位存储到字符型变量中。

### 2.5.1 隐式类型转换

---

- ▶ (5) 将unsigned short型数据赋给int、long变量时，数据存入低位，高位补0。将short型数据赋给int、long变量时，数据存入低位，高位补0或者补1。



## 2.5.1 隐式类型转换

---

- ▶ (6) 将存储空间长度大的整型赋值给小的整型时，低字节复制，高字节“丢弃”。将长度相同的无符号和有符号整型相互赋值时，符号位与数值位同时复制。

## 2.5.1 隐式类型转换

- ▶ Ariane 5
- ▶ 阿丽亚娜5型火箭是欧洲研制的一次性运载火箭，主要作用是将人造卫星发射到地球同步轨道或低地轨道。
- ▶ 阿丽亚娜5号运载火箭501首次测试发射在1996年6月4日，结果失败，整台火箭在发射后37秒自行毁灭，原因是控制火箭飞行的软件故障，这件事可以说是历史上损失最惨重的软件故障事件之一。



## 2.5.1 隐式类型转换

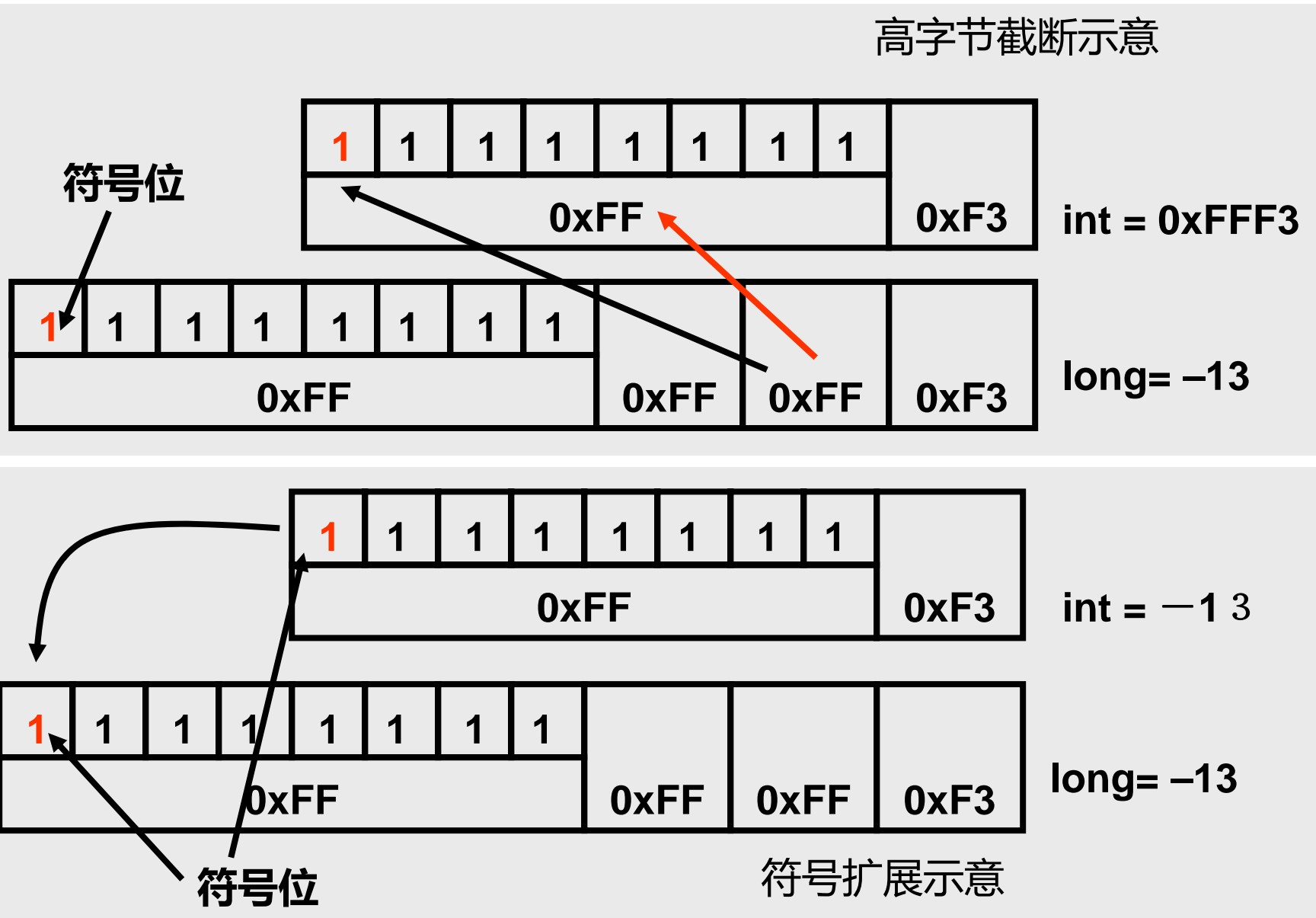
- ▶ 资料显示其64位浮点数转换16位整型时处理有误，造成处理器异常陷阱

（processor trap）。浮点数对于16位整型来讲会超出很多，所以无法运算。考虑到效能的设计（转换为整型）而使处理器无法负荷。



# 2.5.1 隐式类型转换

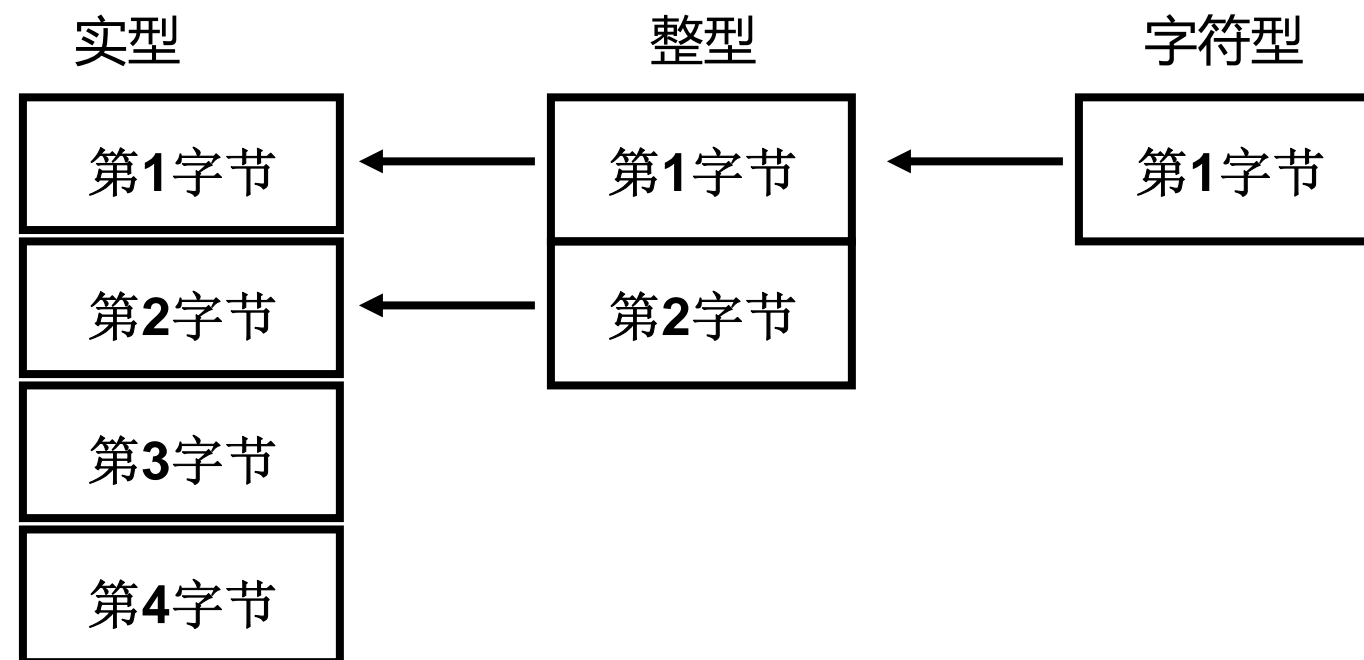
- ▶ 混合运算的
- ▶ 风险控制



## 2.5.1 隐式类型转换

- ▶ 混合运算的
- ▶ 风险控制

具体的类型转换：



- (1) **char**  $\Rightarrow$  **short**、**int**、**long** : “符号扩展” , 转换结果相同。
- (2) **unsigned char**  $\Rightarrow$  **short**、**int**、**long** : 直接字节传递 , 转换结果相同。
- (3) **int**  $\Rightarrow$  **long** : “符号扩展” , 转换结果相同。

## 2.5.1 隐式类型转换

- ▶ 混合运算的
- ▶ 风险控制

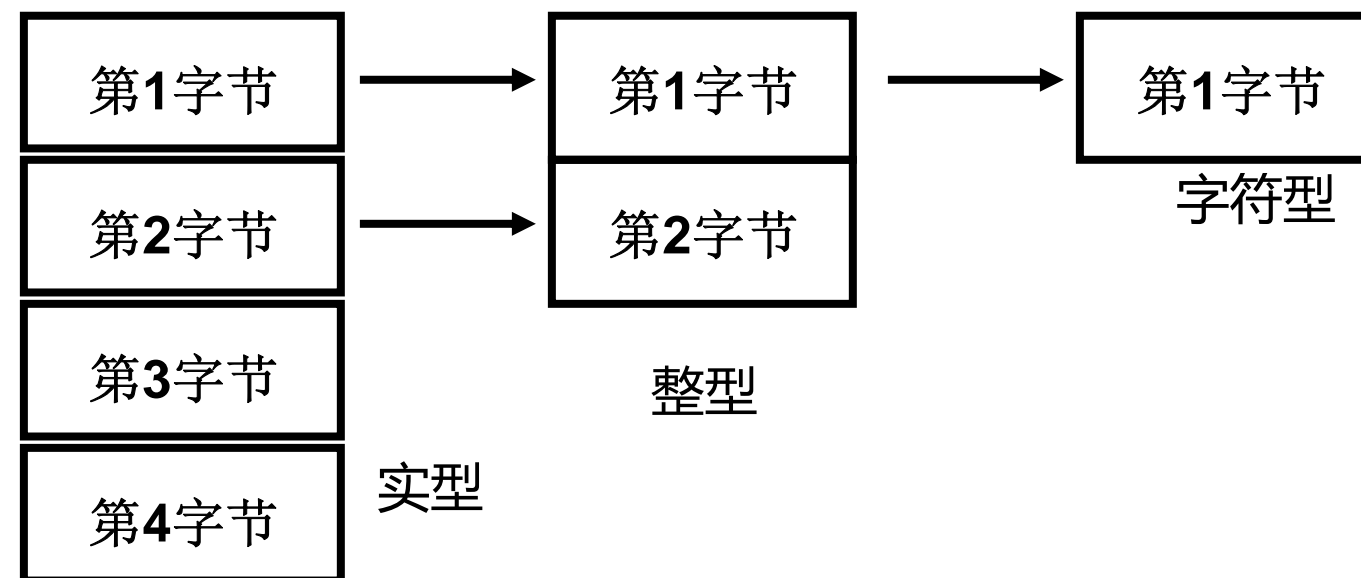
(4) **unsigned int**  $\Rightarrow$  **long** : 直接字节传递, 转换结果相同。

(5) **char**  $\Rightarrow$  **float**、**int**  $\Rightarrow$  **float**、**long**  $\Rightarrow$  **float**、**unsigned char**  $\Rightarrow$  **float**、**unsigned int**  $\Rightarrow$  **float**、**unsigned long**  $\Rightarrow$  **float** :

直接字节传递, “符号扩展”, 转换结果相同。

(6) **float**  $\Rightarrow$  **double**、**double**  $\Rightarrow$  **long double** :

直接字节传递, “符号扩展”, 转换结果相同。



## 2.5.1 隐式类型转换

---

- ▶ 混合运算的
- ▶ 风险控制

( 7 ) **short**、**int**、**long**  $\Rightarrow$  **char** : “截断高字节”。

( 8 ) **short**、**int**、**long**  $\Rightarrow$  **unsigned char** : “截断高字节”。

( 9 ) **long**  $\Rightarrow$  **int** : “截断高字节”。

( 10 ) **unsigned int**  $\Rightarrow$  **long** : “截断高字节”。

( 11 ) **float**  $\Rightarrow$  **char** 、 **float**  $\Rightarrow$  **int**、 **float**  $\Rightarrow$  **long**、 **float**  $\Rightarrow$  **unsigned char** 、 **float**  $\Rightarrow$  **unsigned int**、 **float**  $\Rightarrow$  **unsigned long** :

“截断高字节”，保留整数部分，如果超出数值范围则出错。

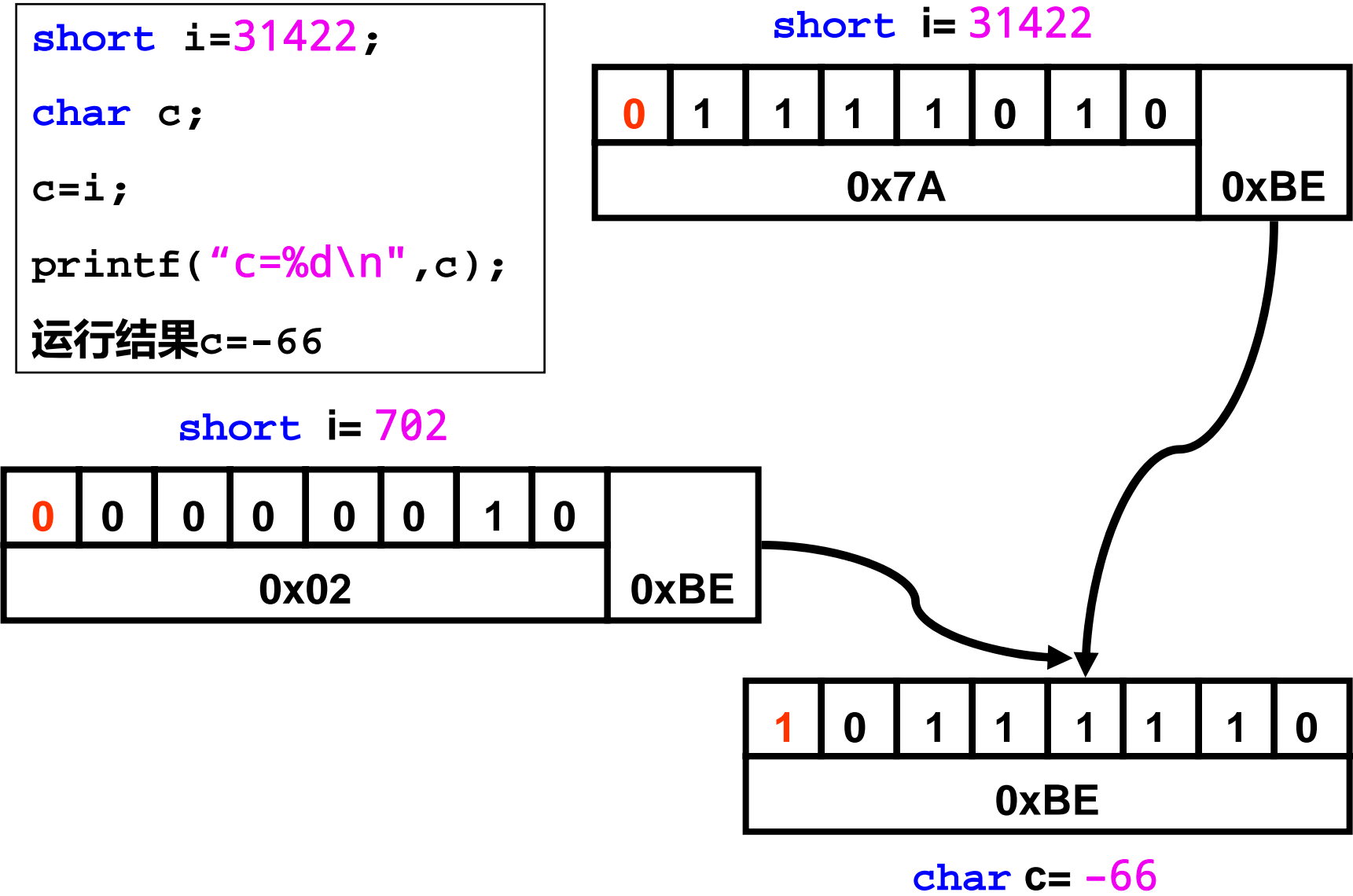
( 12 ) **double**  $\Rightarrow$  **float** 、 **long double**  $\Rightarrow$  **double** :

“截断高字节”，如果超出数值范围则出错。

# 2.5.1 隐式类型转换

- ▶ 混合运算的
- ▶ 风险控制

```
short i=31422;  
char c;  
c=i;  
printf("c=%d\n",c);  
运行结果c=-66
```





## 2.5.2 显式类型转换

---

- ▶ 不能进行自动类型转换时，或在程序中要指定数据类型时，就要利用类型转换运算符进行强制类型转换，称为**显式类型转换**（explicit cast）。

## 2.5.2 显式类型转换

表2-15 显式类型转换运算符

运算符	功能	目	结合性	用法
(type)	显式类型转换	单目	自右向左	(type)expr

## 2.5.2 显式类型转换

---

- ▶ (1) 显式类型转换运算符的运算对象可以是任意类型的常量、变量或表达式。如果是表达式一般用圆括号将其明确，否则容易发生歧义。
- ▶ 示例

```
(int)x+y    //将x转换成整型  
(int)(x+y) //将x+y转换成整型
```

## 2.5.2 显式类型转换

---

- ▶ (2) 显式类型转换的目的是人为进行类型转换，使不同类型数据之间的运算进行下去。显式类型转换后会产生一个指定类型的临时数据对象继续参与运算，但expr中原有类型和数据值不会改变。

- ▶ 示例

```
(int)x%3 //x的类型和数据值不变，表达式引用转换成int后的x值
```

## 2.5.2 显式类型转换

---

- ▶ 设计程序时还是尽量设计好数据类型及其表达式，以减少不必要的类型转换。

## 2.5.2 显式类型转换



### 【例2.11】

将一个浮点型变量d保留两位小数（四舍五入）。

解：

```
(int)(d*100+0.5)/100.0  
// (1)步: d=1.2356  
// (2)步: d*100=123.56  
// (3)步: 123.56+0.5=124.06  
// (4)步: (int)124.06=124  
// (5)步: 124/100.0=1.24
```

## 2.5.2 显式类型转换

表2-16 数据的内存形式

变量类型	-32768	-128	-1	168	32767	65535	2147483647
int							
unsigned int							
short							
unsigned short							
char							
unsigned char							

**CP 程序设计**