

- **机器数**是多少位就要写多少位，前后的0都不能省略！也不能随便增加0！
- 机器数里的小数点可写可不写
- 不同进制的后缀要写清楚！（十进制可省略）
- **double**是64位
- 补码的+0，-0是一样的
- 真值一定注意正负！
- 浮点数换算时不要忘记符号位（正负）！
- 存储地址要计算正确
- 细心。。。

# 第一章

3

$$(1) (25.8125)_{10} = (1\ 1001.1101)_2 = (31.64)_8 = (19.D)_{16}$$

$$(2) (101101.011)_2 = (45.375)_{10} = (55.3)_8 = (2D.6)_{16} = (0100\ 0101.0011\ 0111\ 0101)_{8421}$$

$$(3) (0101\ 1001\ 0110.0011)_{8421} = (596.3)_{10} = (1001010100.010011...)_{2} = (254.4...)_{16}$$

$$(4) (4E.C)_{16} = (78.75)_{10} = (100\ 1110.11)_2$$

4

数值	原码
+0.1001	0.1001000
-0.1001	1.1001000
+1.0	溢出
-1.0	溢出
+0.010100	0.0101000
-0.010100	1.0101000
+0	0.0000000
-0	1.0000000

5

数值	补码	移码 (偏置常数=1 0000000)
+1001	0 0001001	1 0001001
-1001	1 1110111	0 1110111
+1	0 0000001	1 0000001
-1	1 1111111	0 1111111
+10100	0 0010100	1 0010100
-10100	1 1101100	0 1101100
+0	0 0000000	1 0000000
-0	0 0000000	1 0000000

6

$[x]_{\text{补}} = 1110\ 0111$	$x = -001\ 1001\text{B} = -25$
$[x]_{\text{补}} = 1000\ 0000$	$x = -1000\ 0000\text{B} = -128$
$[x]_{\text{补}} = 0101\ 0010$	$x = +101\ 0010\text{B} = 82$
$[x]_{\text{补}} = 1101\ 0011$	$x = -010\ 1101\text{B} = -45$

9

- 在32位计算机中运行一个C语言程序，在该程序中出现了一些变量，已知这些变量在某一时刻的机器数（用十六进制表示）如下，请写出它们对应的真值。

(1) int x: FFFF0006H

=1...1 0000 0000 0000 0110B,

故 $x = -1111\ 1111\ 1111\ 1010B = -(65535-5) = -65530$ 。

(2) short y: DFFCH

=1101 1111 1111 1100B=-010 0000 0000 0100B, 故 $y = -(8192+4) = -8196$ 。

(3) unsigned z: FFFFFFFAH

=1...1 1010B, 故 $z = 2^{32}-6$ 。

(4) char c: 2AH

=0010 1010B, 故 $c = 42$ , c表示字符, 则c为字符'\*' (ASCII码)

(5) float a: C4480000H

=1100 0100 0100 1000 0...0B, 阶码为10001000, 阶为 $136-127=9$ , 尾数为-1.1001B, 故 $a = -1.1001B \times 2^9 = -11\ 0010\ 0000B = -800$ 。

(6) double b: C024800000000000H

=1100 0000 0010 0100 1000 0 0...0B, 阶码为100 0000 0010, 阶为 $1026-1023=3$ , 尾数为1.01001B, 故 $b = -1.01001B \times 2^3 = -1010.01B = -10.25$ 。

12 (5) )

下述格式的浮点数（基为2，移码的偏置常数为128）

符号s（1位）

阶码e（8位移码）

尾数f（7位原码数值部分）

没有明确说明的情况下，就没有规格化，没有隐藏位，阶码也无需专门留给0、无穷大、非数等。

e=1111 1111 对应指数为127

e=1000 0000 对应指数为0

e=0000 0000 对应指数为-128

f=000 0000 对应0.000 0000，此时无论阶码是多少，都表示0

f=000 0001 对应0.000 0001

f=111 1111 对应0.111 1111

最大正数： $+0.111\ 1111\text{B} \times 2^{127}$ 最小非0正数： $+0.000\ 0001\text{B} \times 2^{-128}$ 最大非0负数： $-0.000\ 0001\text{B} \times 2^{-128}$ 最小负数： $-0.111111\text{B} \times 2^{127}$ 

另外，可以表示+0和-0

0	1111 1111	111 1111
0	0000 0000	000 0001
1	0000 0000	000 0001
1	1111 1111	111 1111

x	x	000 0000
---	---	----------

14 设一个变量的值为4098，要求分别用32位补码整数和IEEE 754单精度浮点格式表示该变量（结果用十六进制表示），并说明哪段二进制序列在两种表示中完全相同，为什么会相同？

$$\begin{aligned} 4098 &= 0001\ 0000\ 0000\ 0010\text{B} \\ &= +1.0000\ 0000\ 001\ \text{B} \times 2^{12} \end{aligned}$$

用32位补码整数表示为0000 0000 0000 0000 0001 0000 0000 0010  
用十六进制形式表示为 0000 1002H；

用IEEE 754单精度浮点数格式表示时，  
符号位s=0，

阶码e=12+127=10001011B，

尾数的小数部分f为0000 0000 001B

因此，4098用IEEE 754单精度浮点数格式表示为

0 100 0101 1 000 0000 0001 0000 0000 0000，用十六进制形式表示为4580 1000H。

在上述两种表示中，存在相同的二进制序列0000 0000 001。因为float编码中，有效数值部分中最前面的1被隐藏，其余数值部分为0000 0000 001，而32位补码整数表示中保留了完整的有效数值部分，但最前面的1没有被隐藏，所以除了开头一个1之外，后面的二进制序列是相同的。

17 假定在一个程序中定义了变量x、y和i，其中，x和y是float型变量，i是16位short型变量（用补码表示）。程序执行到某一时刻，x=-0.125、y=7.5、i=100，它们都被写到了主存（按字节编址），其地址分别是100，108和112。请分别画出在大端机器和小端机器上变量x、y和i中每个字节在主存的存放位置。

$x = -0.125 = -0.001B = -1.0B \times 2^{-3}$ ，阶码 $e = 127 - 3 = 01111100B$ ，Float x的机器数为：  
 $101111100\ 000000000000000000000000 = BE000000H$

$y = 7.5 = 111.1B = +1.111B \times 2^2$ ，阶码 $e = 127 + 2 = 1000\ 0001$ ，Float y的机器数为：  
 $0100000011110000\ 0000000000000000 = 40F00000H$

$i = 100 = 0000000001100100B$ ，  
 用16位补码表示为0064H

数据在大端和小端机器中的存放位置（16进制表示）

地址	大端机器	小端机器
&x (100)	BE	00
&x+1 (101)	00	00
&x+2 (102)	00	00
&x+3 (103)	00	BE
&y (108)	40	00
&y+1 (109)	F0	00
&y+2 (110)	00	F0
&y+3 (111)	00	40
&i (112)	00	64
&i+1 (113)	64	00