

2.59:

将 x 与 $0x000000FF$ 按位与，将 y 与 $0xFFFFFFFF00$ 按位与，再将两次得到的值按位或，具体可以表达成下面的式子：

$res = (0x000000FF \& x) | (0xFFFFFFFF00 \& y)$

2.67:

A. 程序运行会得到警告 `left shift count >= width of type`，在代码的第六行“`int beyond_msb = 1<<32;`”中，`int` 类型的大小是 32 位，如果将 1 左移 32 位，会丢失所有的数据，违背 C 语言标准。

B.

```
#include <stdio.h>
int int_size_is_32();
int main() {
    int res = int_size_is_32();
    printf("int 是否为 32 位:%d\n", res);
}
int int_size_is_32() {
    int set_msb = 1<<31;
    return (set_msb<0)?1:0;
}
```

C.

```
#include <stdio.h>
int int_size_is_32();
int main() {
    int res = int_size_is_32();
    printf("The result is:%d\n", res);
}
int int_size_is_32() {
    int set_msb = 1<<15;
    int count=0;
    //用来计算 set_msb 在一开始左移 15 位后还能移多少

    while(set_msb>0) {
        set_msb = set_msb<<1;
        count++;
    }

    return (count==16)?1:0;
}
```

2.71:

A. 不能处理截断的字节是负数的情况

B.

```
typedef unsigned packed_t;
```

```
int xbyte(packed_t word, int bytenum)
{
    return (word << ((3 - bytenum) << 3)) >> 24;
}
```

2.83:

A.

设这个无穷串的值为 X ,

由题可知,

$$X \cdot 2^k = Y + X$$

$$X = \frac{Y}{2^k - 1}$$

B.

(a) $\frac{5}{7}$

(b) $\frac{6}{15} = \frac{2}{5}$

(c) $\frac{19}{63}$

2.87:

描述	Hex	M	E	V	D
-0	8000	0	-14	-0	-0.0
最小的>2 的值	4001	1025/1024	1	$1025 \cdot 2^{-9}$	2.001953
512	6000	1	9	512	512.0
最大的非规格化数	03FF	1023/1024	-14	$1023 \cdot 2^{-24}$	0.000061
$-\infty$	FC00	——	——	$-\infty$	$-\infty$
十六进制表示为 3BB0 的数	3BB0	123/64	-1	$123 \cdot 2^{-7}$	0.960938