计算机系统第一次作业

软件2203班——丁海桐——202226010304

2.61 ： 写一个 C 表达式，在下列描述的条件下产生 1, 而在其他情况下得到 0 。假设 x 是 int 类型。

- A.x 的任何位都等于 1 。`~x == 0`
- B.x 的任何位都等于 0 。`x & x == 0`
- C.x 的最高有效字节中的位都等于 1 。`x & 0xFF000000 == 0xFF000000`
- D.x 的最低有效字节中的位都等于 0 。`x & 0x000000FF == 0x000000FF`

2.71 ： 你刚刚开始在一家公司工作，他们要实现一组过程来操作一个数据结构，要将 4 个有符号字节封装 成一个 32 位 unsigned 。一个字中的字节从 0 (最低有效字节）编号到 3 (最高有效字节）。分配给你的任务是：为一个使用补码运算和算术右移的机器编写一个具有如下原型的函数：

```
/* Declaration of data type where 4 bytes are packed into an unsigned */
typedef unsigned packed_t;
/* Extract byte from word. Return as signed integer */
int xbyte(packed_t word, int bytenum);
```

也就是说，函数会抽取出指定的字节，再把它符号扩展为一个 32 位 in 己 你的前任（因为水平不够高而被解雇了）编写了下面的代码：

```
/* Failed attempt at xbyte */
int xbyte(packed_t word, int bytenum) { return (word»(bytenum«3)) & OxFF; }
```

A. 这段代码错在哪里？ B. 给出函数的正确实现，只能使用左右移位和一个减法。

这段代码错误在没有进行符号扩展，无法处理负数

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

2.87 考虑下面两个基于 IEEE 浮点格式的 9 位浮点表示。

1. ***格式 A***
   - ***有一个符号位。***
   - ***有 k=5 个阶码位。阶码偏置量是 15 。***
   - ***有 n=3 个小数位。***
2. ***格式 B***
   - ***有一个符号位。***
   - ***有 k=4 个阶码位。阶码偏置量是 7 。***
   - ***有 n=4 个小数位。***

下面给出了一些格式 A 表示的位模式，你的任务是把它们转换成最接近的格式 B 表示的值。如果 需要舍入，你要向+ 00 舍入。另外，给出用格式 A 和格式 B 表示的位模式对应的值。要么是整数 （例如， 17), 要么是小数（例如， 17/64 或 17/26) 。

格式A（位）	格式A（值）	格式B（位）	格式B（值）
1 01110 001	$\frac{-9}{16}$	1 0110 0010	$\frac{-9}{16}$
0 10110 101	26	0 1011 1010	26
1 00111 110	$\frac{-7}{2^{10}}$	1 0000 0111	$\frac{-7}{2^{10}}$
0 00000 101	$\frac{5}{2^{17}}$	0 0000 0001	$\frac{1}{2^{10}}$
1 11011 000	-2^{12}	1 1110 1111	$-31 * 2^3$
0 11000 100	$3 * 2^8$	0 1111 0000	$+\infty$

2.88：我们在一个 int 类型为 32 位补码表示的机器上运行程序。 float 类型的值使用 32 位 IEEE 格式， 而 double 类型的值使用 64 位 IEEE 格式。 我们产生随机整数 x 、 y 和 z, 并且把它们转换成 double 类型的值：

```
/* Create some arbitrary values */
int x = random() ;
```

```
int y = random() ;
int z = random();
/* Convert to double */
double dx = (double) x;
double dy = (double) y;
double dz= (double) z;
```

对于下列的每个 C 表达式，你要指出表达式是否总是为 1 。如果它总是为 1, 描述其中的数学原理。 否则，列举出使它为 0 的参数的例子。请注意，不能使用 IA32 机器运行 GCC 来测试你的答案，因 为对于 float 和 double, 它使用的都是 80 位的扩展精度表示。

`(double) (float) x == dx` 否。 `x = 0x7FFFFFFF`

`dx + dy == (double) (x+y)` 否。 `x = y = 0x7FFFFFFF`

`dx +dy + dz == dz + dy + dx` 是。加法具有交换律

`dx * dy * dz == dz * dy * dx` 是。乘法具有交换性

`dx / dx == dy / dy` 否。 `x = 0, y = 1`