题目 1

(1) $!(\sim x)$

(2) $!x$

(3) $!((x\&0xFF)\hat{}\,0xFF)$

(4) $!(\sim((x\&0xFF000000)\,|\,0xFFFFFF))$

题目 2

(a) 恒为 1。dx 是 x 的精确表示，因此 float(x) 与 float(dx) 均为将 x 表示为 float 格式，且舍入方式相同。

(b) 不恒为 1。x 为 INT_MAX，y=-1 时 x-y 会溢出，结果为 INT_MIN。而 dx-dy 为 INT_MAX+1。

(c) 恒为 1。int 转为 double 可以实现精确表示，而且 double 的 frac 位数大，满足结合律。

(d) 不恒为 1。double 的 frac 位数为 53 位，而三个 32 位整型的乘积可能会造成舍入，因此不满足结合律。

(e) 不恒为 1。x=0 的时候，dx/dx 结果为 NaN，该表达式恒为 0。

题目 3

```
float_bits float_absval (float_bits f) {
    float_bits exp = (f >> 23) & 0xFF;
    float_bits frac = f & 0x7FFFFF;
    if (exp == 0xFF && frac != 0) return f;
    else return (f & 0x7FFFFFFF);
}
```

题目 4

```
float_bits float_twice(float_bits f) {
    float_bits s = (f >> 31) << 31;
    float_bits exp = (f >> 23) & 0xFF;
    float_bits frac = f & 0x7FFFFF;
    if (exp == 0xFF) return f;
    if (exp != 0){ // 规格化数
        exp = exp + 1;
        if (exp == 0xFF) // 溢出为无穷
            frac = 0;
    }
    else { // 非规格化数
        frac = frac << 1;
        if (frac & 100000){
            exp = 0;
            frac = frac & 0x7FFFFF;
        }
    }
    return (s | (exp << 23) | frac);
}
```