question 1

1. `(x > 0) || (x - 1 < 0)` ×

   example:

   ```
   x = 0x80000000
   ```

2. `(x & 7) != 7 || (x << 29 < 0)` ✓

   ```
       (x & 7 != 7) || (x << 29 < 0)
   =>  !((x & 7 == 7) && (x << 29 >= 0))
   ```

   当 `x & 7 == 7` 为假时，`(x & 7 == 7) && (x << 29 >= 0)` 为假，原式为真

   当 `x & 7 == 7` 为真时，有 `x == 7` ，此时 `x << 29 = 0xe0000000 < 0` ，故 `(x & 7 == 7) && (x << 29 >= 0)` 为假，原式为真

   终上，原式为永真式

3. `(x * x) >= 0` ×

   example:

   ```
   x = 0xc000
   ```

4. `x < 0 || -x <= 0` ✓

   如果 $x < 0$ 不成立，则 $x >= 0$ ，即 $x \in [0, 2^{31} - 1]$

   所以 $-x \in [-(2^{31} - 1), 0] \subset [INT\_MIN, 0]$

5. `x > 0 || -x >= 0` ×

   example:

   ```
   x = 0x80000000
   ```

6. `x + y == uy + ux` ✓

   有符号加法和无符号加法的位级表示相同

7. `x * ~y + ux * uy == -x` ✓

   ```
       x * ~y + ux * uy == -x
   =>  ux * uy == -(~y + 1) * x
   =>  ux * uy == x * y
   ```

   有符号乘法和无符号乘法的位级表示相同

question 2

1. exponent bias = 2^3 - 1 = 7
2.

Value	Floating point bits	Rounded value
-53/4	1 1010 1010	-13
1/16	0 0011 0000	1/16
-256	1 1111 0000	$-\infty$
31/2048	0 0001 0000	1/64

1. -53/4 = -1101.01 = $(-1)^1 * 1.10101 * 2^3$

   s = 1

   M = 1.10101 => frac = 1010

   E = 3 => exp = E + bias = 10 = $1010_{(2)}$

   rounded_value = $-1.1010 * 2^3 = -13$

2. 1/16 = $(-1)^0 * 1 * 2^{-4}$

   s = 0

   M = 1 => frac = 0000

   E = -4 => exp = E + bias = 3 = $0011_{(2)}$

   rounded_value = value

3. -256 = $(-1)^1 * 1 * 2^8$

   s = 1

   M = 1 => frac = 0000

   E = 8 => exp = E + bias = 15 = $1111_{(2)}$

   exp 全 1，为负无穷

4. 31/2048 = $(-1)^0 * 1.1111 * 2^{-7}$

   最大的非规格化值为 $0.1111 * 2^{-6} = 11110 * 2^{-11} = 30 / 2048$

   所以原式应该进位为 $2.0 * 2^{-7} = 1.0 * 2^{-6}$

   s = 0

   M = 1.0 => frac = 0000

   E = -6 => exp = E + bias = 1 = $0001_{(2)}$

   rounded_value = $1.0 * 2^{-6}$ = 1/64

question 3

1. ```
   int get_LSB(int x) {
       return x & 0xff;
   }
   ```

2. ```
   int complemented_not_LSB(int x) {
       return ~x & ~0xff | (x & 0xff);
   }
   ```

3.
```
int set_LSB_one(int x) {
    return x | 0xff;
}
```

question 4

```
float fpwr2(int x) {
    unsigned exp, frac;
    unsigned u;

    if (x <= -150) {
        exp = 0;
        frac = 0;
    }
    else if (x <= -127) {
        exp = 0;
        frac = 1 << (x + 150 - 1);
    }
    else if (x <= 127) {
        exp = x + 127;
        frac = 0;
    }
    else {
        exp = 255;
        frac = 0;
    }
    u = exp << 23 | frac;
    return u2f(u);
}
```

question 5

Operand	Value
%eax	0x100
0x104	0xAB
$0x108	0x108
(%eax)	0xFF
4(%eax)	0xAB
9(%eax, %edx)	0x11
260(%ecx, %edx)	0x13
0xFC(, %ecx, 4)	0xFF
(%eax, %edx, 4)	0x11

question 6

```
int arith(int x, int y, int z) {
    int t1 = 3 * z;
    int t2 = z << 4;
    int t3 = x + y + 65535;
    int t4 = t2 * t3;
    return t4;
}
```

question 7

```
movl 8(%ebp), %eax
sall 4, %eax
movl 12(%ebp), %ecx
sarl %ecx, %eax
```

question 8

src_t	dest_t	Instruction
int	int	movl %eax, (%edx)
char	int	movsbl %al, (%edx)
char	unsigned	movsbl %al, (%edx)
unsigned char	int	movzbl %al, (%edx)
int	char	movb %al, (%edx)
unsigned	unsigned char	movb %al, (%edx)
unsigned	int	movl %eax, (%edx)