

# 1

$X = -101$  的 16 进制表示的补码为 `FFFFFF9B`。

$X + Y$  的 16 进制补码为 `FFFFFF05`。

$X - Y$  的 16 进制补码为 `00000031`

# 2

1. 若  $x \geq 0, y \geq 0$ , 由于正数的补码和原码一致且  $x + y > 0$ , 于是有:

$$[x]_{\text{补}} + [y]_{\text{补}} = x + y \equiv [x + y]_{\text{补}} \pmod{2^w}$$

2. 若  $x \geq 0, y < 0, x + y \geq 0$ , 则  $[x]_{\text{补}} = x, [y]_{\text{补}} = y + 2^w$ , 且  $x + y < 2^{w-1}$ , 于是有:

$$\begin{aligned} [x]_{\text{补}} + [y]_{\text{补}} &= x + y + 2^w \equiv x + y \pmod{2^w} \\ x + y &= [x + y]_{\text{补}} \\ \Rightarrow [x]_{\text{补}} + [y]_{\text{补}} &\equiv [x + y]_{\text{补}} \pmod{2^w} \end{aligned}$$

3. 若  $x < 0, y \geq 0, x + y \geq 0$ , 同 2 交换  $x, y$  可得。

4. 若  $x \geq 0, y < 0, x + y < 0$ , 则  $[x]_{\text{补}} = x, [y]_{\text{补}} = y + 2^w$ , 且  $x + y \geq -2^{w-1}$ , 于是有:

$$[x]_{\text{补}} + [y]_{\text{补}} = x + y + 2^w = [x + y]_{\text{补}}$$

5. 若  $x < 0, y \geq 0, x + y < 0$ , 同 4 交换  $x, y$  可得。

6. 若  $x < 0, y < 0$ , 则  $[x]_{\text{补}} = x + 2^w, [y]_{\text{补}} = y + 2^w$ , 且  $x + y \geq -2^w$  于是有:

$$\begin{aligned} [x]_{\text{补}} + [y]_{\text{补}} &= x + y + 2^{w+1} \equiv x + y + 2^w \pmod{2^w} \\ x + y + 2^w &= [x + y]_{\text{补}} \\ \Rightarrow [x]_{\text{补}} + [y]_{\text{补}} &\equiv [x + y]_{\text{补}} \pmod{2^w} \end{aligned}$$

综上所述:

$$[x]_{\text{补}} + [y]_{\text{补}} \equiv [x + y]_{\text{补}} \pmod{2^w}$$

# 3

$$129 = [10000001]_2 = [1.0000001]_2 \times 2^7$$

舍入到后三位:

$$[1.0000001]_2 \approx [1.000]_2$$

因此:

$$\begin{aligned} \text{Exp} &= 14 = [1110]_2 \\ \text{Frac} &= 0 = [000]_2 \end{aligned}$$

## 4

1. `(x < y) == (-x > -y)`, N。
2. `((x + y) << 4) + y - x == 17 * y + 15 * x`, Y。
3. `~x + ~y + 1 == ~(x + y)`, Y。
4. `ux - uy == -(y - x)`, Y。
5. `(x >= 0) || (x < ux)`, N。
6. `((x >> 1) << 1) <= x`, Y。
7. `(double)(float)x == (double)x`, N。
8. `dx + dy == (double)(y + x)`, N。
9. `dx + dy + dz == dz + dy + dx`, Y。

## 5

1. f。
2. b。
3. a。
4. c。
5. e。
6. h。

## 6

`%rax`

## 7

由于条件跳转指令对于现代流水线处理器的效率有很大负面影响，所以在 `__builtin_expect()` 提示下，会选择生成尽量避免条件跳转指令的指令序列。

当 `unlikely(a == 2)` 时，表示 `a` 很有可能不为 2，则会选择优先考虑 `a` 未知的情况，将跳转指令用于进只行 `a == 2` 情况下特判的部分（即跳过 `.L3` 部分）。

当 `likely(a == 2)` 时，表示 `a` 很有可能为 2，则默认进行 `a == 2` 的特判处理，对于 `a != 2` 则选择两次跳转处理（即将 `.L2` 部分置于默认情况的下方）。