Homework1 Answer

T1

略

T2

$$
\begin{gathered}
98 \to 01100010 \\
-105 \to 11101001 \to 10010110 \to 10010111 \\
01000010 \to 66 \\
11101111 \to 10010000 \to 10010001 \to -17
\end{gathered}
$$

T3

$a.\ 01\ +\ 11\_0011\ =\ 00\_0001\ +\ 11\_0011\ =\ 11\_0100\ =\ (-12)_D$

$b.\ 111\ +\ 010\_0110\ =\ 111\_1111\ +\ 010\_0110\ =\ 010\_0101\ =\ (37)_D$

$c.\ 1010\ +\ 1101\ =\ 0111 = (7)_D$

$d.\ 0001\ +\ 1110 = (-1)_D$

T4

a. 1110_1011

b. 0001_1110

c. 1111_0000

d. 0000_0001

T5

$4.3 = 100.01\_0011\_0011\cdots = (1.0001\ 0011\ 0011\cdots) \times 2^2$

0 10000001 00010011001100110011010（注意末尾两位）（注：由于题目没说清楚，所以只写10位的也算对）

示例代码：

```c
#include <stdio.h>

union my_union {
    int   a;
    float b;
};

int main() {
    union my_union t;
    t.b = 4.3;
    for (int loop = 31; loop >= 0; loop--) {
        putchar((t.a & (1 << loop)) == 0 ? '0' : '1');
    }
    return 0;
}
```

阶码全为1，当尾数全为1时表示无穷，正负号由符号位决定。阶码全为1且尾数不全为0则表示NaN

T6

10001001 = 137

$(1.11110011010010000000000) \times 2^{137-127} = 111\_1110\_0110.1001 = 2022.5625 = 2022\frac{9}{16}$

T7

a. 1010_0101 AND 1101_0101 = 1000_0101

b. 1000_1110 OR 1111_0101 = 1111_1111

c. NOT(1111_0001) OR NOT(0101_1010) = 0000_1110 OR 1010_0101 = 1010_1111

d. (x1234 AND X5678) OR (xABCD AND X99EF) = x1230 OR x89CD = x9BFD

e. x6A12 XOR x3A15 = x5007

T8

A	B	C	Q_1	Q_2
0	0	0	1	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

$Q_2 = A\ AND\ B\ AND\ C$

T9

1. 两种解答
   - 转义字符：000010 010000 101000 001101 $\rightarrow$ CQoN
   - 非转义字符：010111 000111 010001 011100 101011 100101 110001 110010 $\rightarrow$ XHRcblxy
2. 规避格式符号、仅允许字符的情况下传递信息等。

T10

$$(\sum_{i=0}^{23} 2^{-i}) \times 2^{254-127} = (2 - 2^{-23}) \times 2^{127} = 2^{128} - 2^{104}$$

T11

随意舍入，不管它

1. Mult
   - $A[23:31] + B[23:31] + 10000001 \rightarrow EXP[8:0]$
   - $\{1, A[0:23]\} * \{1, B[0:23]\} = FRAC[0:48]$

- $(FRAC[47]?\{0, EXP, FRAC[23:46]\}:\{0, EXP+00000001, FRAC[24:47]\}) \rightarrow C[0:32]$

2. Add
   - $A \geq B \Rightarrow A[23:31] \geq B[23:31]$
   - $A[23:31] - B[23:31] \rightarrow SHIFT$
   - $\{01, A[0:23]\} + (\{01, B[0:23]\} >> SHIFT) \rightarrow FRAC[0:25]$
   - $(FRAC[24]?\{0, A[23:31]+00000001, FRAC[1:24]\}:\{0, A[23:31], FRAC[0:23]\}) \rightarrow C[0:32]$

以上是大致流程，可自行画图。