

# 计算机系统 HW1

SA25011049 李宇哲

## T1 2.58

```
1 #include <stdio.h>
2 #include <string.h>
3
4
5 void show_short(short x) {
6     // 输出 x 的二进制表示
7     for (int i = sizeof(short) * 8 - 1; i >= 0; i--) {
8         putchar((x & (1 << i)) ? '1' : '0');
9     }
10    putchar('\n');
11}
12
13 void show_long(long x) {
14     for (int i = sizeof(long) * 8 - 1; i >= 0; i--) {
15         putchar((x & (1L << i)) ? '1' : '0');
16     }
17    putchar('\n');
18}
19
20 void show_double(double x) {
21     unsigned long long bits;
22     memcpy(&bits, &x, sizeof(double));
23     for (int i = sizeof(double) * 8 - 1; i >= 0; i--) {
24         putchar((bits & (1ULL << i)) ? '1' : '0');
25     }
26    putchar('\n');
27}
28 int main() {
29     short s = -12345;
30     long l = 123456789012345;
31     double d = -12345.6789;
32     printf("Short: %d\nBinary: ", s);
33     show_short(s);
34     printf("Long: %ld\nBinary: ", l);
35     show_long(l);
36     printf("Double: %f\nBinary: ", d);
37     show_double(d);
38     return 0;
39 }
```

由于我手头只有一种机器，在当前主机上结果如下

```
(base) [?] base ➤ innerpeace@innerpeace ➤ ~ / csapp / hw ➤ uname -a
Linux innerpeace 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun 5 18:30:46 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
```

```

● (base) [2] base innerpeace@innerpeace ~/csapp/hw clang hw1.c          /0.2s
● (base) [2] base innerpeace@innerpeace ~/csapp/hw ./a.out

Short: -12345
Binary: 1100111111000111
Long: 123456789012345
Binary: 0000000000000000111000010010001000011000001101110111101111001
Double: -12345.678900
Binary: 110000001100100000111001101011100110001111100010100001           /0.0s

```

## T2 2.61

```

1 int get_result(int x) {
2     return !(~x) | !x | !((x & 0xFF) ^ 0xFF) | !(((unsigned)x >> 24) & 0xFF);
3 }

```

## T3 2.77

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void) {
5     int x = 5;
6
7     // A
8     int a = (x << 4) + x;
9     printf("A: %d\n", a);
10
11    // B
12    int b = x - (x << 3);
13    printf("B: %d\n", b);
14
15    // C
16    int c = (x << 6) - (x << 2);
17    printf("C: %d\n", c);
18
19    // D
20    int d = (x << 4) - (x << 7);
21    printf("D: %d\n", d);
22
23    return 0;
24 }

```

## T4 2.84

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 unsigned f2u(float f) {
4     return *((unsigned*)&f);
5 }
6
7 int float_le(float x, float y) {
8     unsigned ux = f2u(x);
9     unsigned uy = f2u(y);

```

```

10
11     unsigned sx = ux >> 31;
12     unsigned sy = uy >> 31;
13
14
15     return (sx != sy) ? (sx > sy) : ((sx == 0) ? (ux <= uy) : (ux >= uy));
16 }
17 int main(void) {
18
19     // test
20     float x = -1.5f;
21     float y = 2.5f;
22     int result = float_le(x, y);
23     printf("Result of float_le(%f, %f): %d\n", x, y, result);
24     return 0;
25 }
```

## T5 2.89

**A**

总是成立，都是将同一个整数值转换为float

**B**

反例：x = 2147483647, y = -1

**C**

总是为1， dx, dy和hz都是从int精确转换，整数的double表示满足结合律

**D**

反例：

```

1 int x = 2000000000;
2 int y = 2000000000;
3 int z = 2;
```

## T6 2.91

A.

11.0010010000111111011011

B.

11.001001001001...<sub>2</sub> (循环)

C.

小数点后8位

