HW for Part 1

> 计64 翁家翌 2016011446

1

原始数字为 `0b10001110111011111100000000000000`，转换成16进制为 `0x8eefc000`

a. 10进制补码，表示为 `-1896890368`

b. 无符号10进制，表示为 `2398076928`

c. 单精度浮点数，表示为 `-1 × 2^(29-127) × 1.873046875 = -5.91029381333555×10^-30`

2

代码如下所示，使用一个union类型即可。

```c
#include <stdio.h>
union tmp {
    float f;
    int i;
} a;
int main() {
    scanf("%f", &a.f);
    printf("0x%X\n", a.i);
}
```

3

假设有 r 位检验码，则需满足 $2^r \geq k + r + 1 = 2^n - 1 + 1 = 2^n$，因此 $r \geq n$

因此 r 最小为 n，最小比例为 $\frac{n}{2^n-1}$。

n	r	Ratio
3	3	3/7 = 42.86%
4	4	4/15 = 26.67%
5	5	5/31 = 16.13%
6	6	6/63 = 9.52%

4

825 = 0x0339 = 0b0000001100111001

5

-125 = 0x10000 - 0x7d = 0xff83 = 0b1111111110000011

6

x = 12，y = -7，假设使用8位存储。

(1) 转换成原码为：x = 0x0c = 0b00001100，y = 0xf9 = 0b11111001

相乘得到 0b1111100100 + 0b11111001000 = 0b11110101100，溢出部分截取之后为 0b10101100 = 0xac，转换成补码为 -84

(2)

步骤	操作	部分积	附加位
0	初始值	0000000011111001	0
1	-X	1111010011111001	0
	>>	1111101001111100	1
2	+X	0000011001111100	1
	>>	0000001100111110	0
3	>>	0000000110011111	0
4	-X	1111010110011111	0
	>>	1111101011001111	1
5-8	>>	1111111110101100	1

结果为 `0b10101100`，与(1)一致。

(3) 将其转换为绝对值之后进行除法，结果为 `0b11011`，加上符号位为 `0b111011`。