Computer Systems I

Homework 1

# Problem 1

1. 0101 0011+0011 0111=1000 1001

2. 0010 1100+1010 0010=1100 1110

3.1111 1100+1011 0101=1011 0001(overflow)

4. 0111 1010+0011 1011=1011 0101

5. 0110 0101−0010 1010=0011 1011

6. 1100 0010−1111 1100=1100 0110

7. 0110 1111−1111 0101=0111 1010

8. 0111 0010−1000 0010=1111 0000

# Problem 2

1. 错误。x=-2^31时会发生溢出，产生的相反数仍为负数
2. 正确。∼*x*+∼*y*简化为-x-1-y-1,=-(x+y)-2,−(*x*+*y*)−2<−(*x*+*y*)−1 总是为真。所以这个语句对所有*x*和*y*都是真。
3. 正确。右移6位后再左移6位，相当于是将低6位置为0，结果必定比原数小。
4. 正确。如果选择的 x 和 y 使得 x + y 溢出，则结果可能不同。但在非溢出情况下，这个表达式成立。

# Problem 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | For | mat A | Format B |  |
| No. | Bits | Value | Bits | Value |
| 1 | 0 10010 011 | 1.375\*2^3 = 11D | 0 110 01100 | 1536D |
| 2 | 1 00011 010 | 1.25\*2^-12= -5/2^15 | 符号位仅有3位，无法表示-12次方 | NULL |
| 3 | 0 00011 010 | 1.25\*2^-12 = 5/2^15 | 符号位仅有3位，无法表示-12次方 | NULL |
| 4 | 1 11000 000 | -512 | 符号位无法表示2^9 | NULL |
| 5 | 0 10011 100 | 24 | 0 111 10000 | 32768 |

# Problem 4

unsigned srl(unsigned x, int k) {

unsigned xsra = (int) x >> k;

int w = sizeof(int) << 3;

unsigned m = ((1 << (w - k)) - 1);

return xsra & m;

}

int sra(int x, int k) {

unsigned xsrl = (unsigned)x >> k;

int w = sizeof(int) << 3;

int sign = x & INT\_MIN;

int m = sign >> (w - k);

return (int)(xsrl | m);

}

# Problem 5

float\_bits float\_denorm\_zero(float\_bits f) {

unsigned sign = f >> 31;

unsigned exp = (f >> 23) & 0xFF;

unsigned frac = f & 0x7FFFFF;

if (exp == 0) {

frac = 0;

}

return (sign << 31) | (exp << 23) | frac;

}

# Problem 6

Code 1:由于i为unsigned类型，故其永远不是负数，所以循环条件一直成立，死循环。

Code 2:各个平台的sizeof(int)可能不一样，故运行出来的结果也会不一样。而且sizeof(int)的值可能不是1，故无法实现累加的功能。