

# 郭锦沛 520021910746

## Homework 1

### Thinking Questions

1.

The formula is:

$$\frac{1}{1 - p + \frac{p}{n}} = 7$$

Let  $p = 0.98$ , so  $n = \lceil \frac{0.98}{\frac{1}{7} - 0.98} \rceil + 1 = 8$

2.

$x = 4x - x = 3x$ , so  $M = 3$ .

$y = y/16$ , so  $N = 16$ .

3.

- False. When  $x = 0x80000000$ .
- False. When  $x = 123456$ .
- True. Each positive integer has its corresponding negative integer.
- False. When  $x = 0x80000000$ .
- True. int will be transformed into unsigned.
- True.  $y = -y - 1$ , so  $eq = -xy - x + ux * uy$ , and  $xy = ux * uy$ , so  $eq = -x$ .
- True.  $x * 4 = x \ll 2$  and  $y * 8 = y \ll 3$ .
- True. Let  $eq = (x \gg 2) \ll 2$   
When the lowest 2 bytes of  $x$  contain 1,  $eq < x$ , otherwise,  $eq = x$ .

4.

- N.  $x = 0x80000000$ .
- Y.
- Y.  $\sim x = -x - 1$ ,  $\sim y = -y - 1$ ,  $\sim(x + y) = -x - y - 1$ .

- Y.
- Y. unsigned type has a wider positive range than int type.
- Y.
- N. (float)x might lose the accuracy of x while (double)x won't.
- Y.
- N. When  $dx = -dy$  and  $dx$  is much larger than  $dz$ .
- Y.

## Programming

```
int allOddBits(int x) {
//your codes Here
int tool = 0xaa + (0xaa << 8) + (0xaa << 16) + (0xaa << 24);

return !((x & tool) ^ tool);
}
```

```
int isLessOrEqual(int x, int y) {
int sig_x = (x >> 31);
int sig_y = (y >> 31);
int isSame = !(sig_x ^ sig_y);
int res = !((y + ~x + 1) >> 31 & 1);

return (isSame & res) | ((!isSame) & sig_x);
}
```

```
int logicalNeg(int x) {
//your codes here
int neg = ~x + 1;
return ((~x & ~neg) >> 31) & 1;
}
```

```

unsigned floatScale2(unsigned uf) {
//your codes here
    int sig = uf & (1 << 31);
    int exp = (uf & 0x7f800000) >> 23;

    switch (exp)
    {
    case 255:
        return uf;
    case 0:
        return uf << 1 | sig;
    default:
        exp++;
        if(exp == 255)return 0x7f800000 | sig;
        return (exp << 23) | (uf & 0x807fffff);
    }
}

int floatFloat2Int(unsigned uf) {
//your codes here
    int s = uf >> 31;
    int exp = ((uf & 0x7f800000) >> 23) - 127;
    int frac = (0x0007ffff & uf) | 0x000800000;

    if(exp > 31)return 0x80000000;
    if(exp < 0)return 0;

    if(exp > 23)
        frac <<= (exp - 23);
    else
        frac >>= (23-exp);

    if(frac >> 31)return 0x80000000;
    if(s)return ~frac + 1;
    return frac;
}

```

Score	Rating	Errors	Function
2	2	0	allOddBits
4	4	0	isLessOrEqual
4	4	0	logicalNeg
5	5	0	floatScale2
5	5	0	floatFloat2Int
Total points: 20/20			