Homework3

这是 Homework3.pdf的标准答案。

Q1

思路：大端存储指的是数据的低位存在高地址，小端存储指的是数据的低位存在低地址。但是请注意，二者在读写时都是从低地址开始的。

因此该题目答案应该是：

	little	big
1	33	14
2	3302	140A
3	33020A14	140A0233

Q2

fraction	binary representation	decimal representation
1/8	0.001	0.125
3/4	0.11	0.75
43/16	10.1011	2.6875
25/16	1.1001	1.5625
51/16	11.0011	3.1875

Q3

通用表示

- **指数 E**：$E = e - b$，其中 e 是存储的指数值，b 是偏移量（bias），通常为 $2^{(k-1)} - 1$。
- **有效数 M**：$M = 1 + f$，其中 f 是尾数部分的值，表示为二进制分数。
- **尾数 f**：如果尾数为 n 位，尾数 f 的计算公式为：$f = \frac{b_0}{2^1} + \frac{b_1}{2^2} + \ldots + \frac{b_{n-1}}{2^n}$，其中 b_i 是尾数的二进制位。
- **值 V**：浮点数的值可以表示为：$V = (-1)^s \times M \times 2^E$，其中 s 是符号位。

5.0

- **符号位 s**：0（正数）
- **数值表示**：5.0 可以表示为 1.25×2^2。

- **指数 E**：E=2 → $e = 2 + b$。
- **有效数 M**：$M = 1 + 0.25 = 1.25$。
- **尾数 f**：$f = 0.01_2 = 0.25$。

最大奇数整数表示

$V = (-1)^s \times M \times 2^E$ 最大的奇数V的二进制表示应该是: 1111111 因此这要求M=1+f的小数部分f的前m位是1，m=E。这样才能保证整数部分全1，而小数部分全0。

从而我们需要分类讨论小数位数n和指数位数k的大小: $$ E_{\text{max}} = e_{\text{max}} - b = (2^k - 2) - (2^{(k-1)} - 1) = 2^{(k-1)} - 1 $$ $$ E_{\text{max}} = 2^{(k-1)} - 1 $$

1. 如果E_{max} < n 那么m=E_{max} 时，取得最大的正奇数 s=0 ,E = E_{max} , M = 1.1...1000(E_{max} 个1)
2. 如果$E_{max}>=n$ 那么E=n时，取得最大的整数 s=0 , E=n , M = 1.1111..1

最小规范化正数的倒数

首先表示 最小的规范化正数： s =0 , E = 2^{1-b}, f = 0 M = 1

变成倒数时 只需要调整指数为 即可： s = 0, E'= ${b-1}$,e = E'+b = 2b-1 = $2^k -3$ f =0 , M =1

题目4

Format A		**Format B**	
Bits	**Value**	**Bits**	**Value**
1 01110 001	$-\frac{9}{16}$	1 0110 0010	$-\frac{9}{16}$
0 10110 101	208	0 1110 1010	208
1 00111 110	$-\frac{7}{1024}$	1 0000 0111	$-\frac{7}{1024}$
0 00000 101	5 $\times$ 2^{-17}	0 0000 0001	2^{-10} 向正无穷舍入
1 11011 000	-4096	1 1111 0000	-inf （溢出）
0 11000 100	768	0 1111 0000	inf (溢出）

关于float double int之间类型转换的结论

1. float 转 double没有精度损失 因为float的E和M的位数都小于double,所以E和M都可以精确转换。
2. int转double不会有精度损失，但是转float可能会有 Tmax = $2^{32}-1$需要32-1位小数，所以可以被double表示，不能被float表示。

Q5

1. !(x+1)
2. !x
3. !(~x&0xff)
4. !(x>>24+1)

Q5（2）

A. 正确，这是由于左右近似的方法是一样的，即使int转float/double会有精度损失 B. 错误，当x=0且y=Tmin时不符合该等式 C. 正确，由于dx，dy，dz都是由int32转成double类型，在这个过程中不会出现精度损失，因此符合等式 D. 错误，dx，dy，dz相乘的数可以很大，进而导致乘积的精度损失，且浮点数的运算不满足结合律

反例：$$\text{设 } x=2^{27}+1, y=2^{27}+1, z=2^{27}+2$$，请同学们思考一下double的精度与舍入问题

E. 错误，可以代入dx=0.0

Q6

```
float_bits float_absval(float_bits f)
{
  float_bits sign = f & 0x80000000;
  float_bits exp = f & 0x7f800000;
  float_bits frac = f & 0x007fffff;

  if (exp == 0x7f800000 && frac != 0)
    return f;

  return f & 0x7fffffff;
}
```

Q7

```
float_bits float_twice(float_bits f) {
    unsigned sign = uf & 0x80000000;
    unsigned exp = (uf >> 23) & 0xFF;
    unsigned frac = uf & 0x7FFFFF;
    if (exp == 0xFF) {
        return uf;
    }
    if (exp == 0) {
        frac = frac << 1;
        return sign | frac;
    }
    exp = exp + 1;
    if (exp == 0xFF) {
        return sign | 0x7F800000;
    }
    return sign | (exp << 23) | frac;
}
```