Homework3

- 题目 1
  - Consider the following function
  - typedef unsigned char * byte_pointer;
  - void show_bytes(byte_pointer start, int len) {
    - int i;
    - for (i=0; i<len; i++)
      - printf(“%.2x”, start[i]);
    - }
  - int val = 0x140A0233;
  - byte_pointer valp = (byte_pointer) & val;
  - What is the output of the following call to show_bytes on big-endian and little-endian machines respectively?

	little-endian	big-endian
show_bytes(valp, 1);		
show_bytes(valp, 2);		
show_bytes(valp, 4);		

- 题目 2
  - Fill in the missing information in the following table:

Fractional value	Binary representation	Decimal representation
1/8		
3/4		
	10.1011	
25/16		
		3.1875

- 题目 3
  - Given a floating-point format with a k-bit exponent and an n-bit fraction, write formulas for the exponent E, significand M, the fraction f, and the value V for the quantities that follow. In addition, describe the bit representation.
  - A. The number 5.0
  - B. The largest odd integer that can be represented exactly
  - C. The reciprocal of the smallest positive normalized value

- 题目 4
  - Consider the following two 9-bit floating-point representations based on the IEEE floating-point format.
  - Format A
    - There is one sign bit.
    - There are k = 5 exponent bits. The exponent bias is 15.
    - There are n = 3 fraction bits.
  - Format B
    - There is one sign bit.
    - There are k = 4 exponent bits. The exponent bias is 7.
    - There are n = 4 fraction bits.

  - Below, you are given some bit patterns in Format A, and your task is to convert them to the closest value in Format B. If rounding is necessary, you should **round toward positive infinity**
  - In addition, give the values of numbers given by the Format A and Format B bit patterns. Given these as whole numbers(eg.,17) or as fractions(eg.,17/64 or 17/26).

- 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			
1 00111 110			
0 00000 101			
1 11011 000			
0 11000 100			

题目 5

- int 为 32 位， float 和 double 分别是 32 位和 64 位 IEEE 格式
  - Int x =random();
  - Int y = random();
  - Int z = random();
  - Double dx = (double)x;
  - Double dy = (double)y;
  - Double dz = (double)z;
- 对于下面的每个 C 表达式，判断是否恒为 1 。如果是请说明原理，如果不是请举出反例。
  - A. (float)x == (float)dx
  - B. dx-dy == (double)(x-y)
  - C. (dx+dy)+dz == dx+(dy+dz)
  - D. (dx*dy)*dz == dx*(dy*dz)
  - E. dx/dx == dz/dz

题目 6

- 编写如下函数，求浮点数 f 的绝对值 |f| 。如果 f 是 NaN ，那么应该直接返回 f （注意 NaN 不要对 f 做任何修改）。
- 其中 float_bits 等价于 unsigned ，是 float 数字的二进制形式
  - typedef unsigned float_bits;
- /* Compute |f|. If f is NaN, then return f. */
- float_bits float_absval (float_bits f);

题目 7

- 实现如下函数，对于浮点数 f ，计算 2.0*f 。如果 f 是 NaN ，你的函数应该简单返回 f 。
- /* Compute 2*f. If f is NaN, return f. */
- float_bits float_twice(float_bits f);