Homework5

Homework5

- 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;

Homework5

- 题目1
  - 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);`		

Homework5

- 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

Homework5

- 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

Homework5

- 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.

Homework5

- 题目4
  - 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** $+\infty$.
  - 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).

Homework5

- 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			