1. **Finish your homework independently**
2. **Convert this docx to pdf: “stuID_name_csapp2.pdf”**
   Example: ”2017010000_zhangsan_csapp2.pdf”
3. **Submit this pdf: learn.tsinghua.edu.cn**

2.82 ◆◆

Consider numbers having a binary representation consisting of an infinite string of the form 0.*y y y y y y* . . ., where *y* is a *k*-bit sequence. For example, the binary representation of $\frac{1}{3}$ is 0.01010101 . . . (*y* = 01), while the representation of $\frac{1}{5}$ is 0.001100110011 . . . (*y* = 0011).

A. Let $\mathrm{Y} = \mathrm{B2U_k(y)}$, that is, the number having binary representation *y*. Give a formula in terms of *Y* and *k* for the value represented by the infinite string.

Hint: Consider the effect of shifting the binary point *k* positions to the right.

Formula:

Let value=0. y y y y y. . . , Y = unsigned(y).

Shift the binary point *k* positions to the right, y. y y y y y. . .=value*2^k.

Also, value + Y = y. y y y. . . = value*2^k.

It gives :

$$value = \frac{Y}{2^k - 1}$$

B. What is the numeric value of the string for the following values of *y*?

(a) 001

$$value = \frac{1}{2^3 - 1} = \frac{1}{7}$$

(b) 1001

$$value = \frac{9}{2^4 - 1} = \frac{9}{15} = \frac{3}{5}$$

(c) 000111

$$value = \frac{7}{2^6 - 1} = \frac{7}{63} = \frac{1}{9}$$

2.85 ◆

Intel-compatible processors also support an "extended precision" floating-point format with an 80-bit word divided into a sign bit, k = 15 exponent bits, a single *integer* bit, and n = 63 fraction bits. The integer bit is an explicit copy of the implied bit in the IEEE floating-point representation. That is, it equals 1 for normalized values and 0 for denormalized values. Fill in the following table giving the approximate values of some "interesting" numbers in this format:

	Extended precision	
Description	Value	Decimal
Smallest positive denormalized	$2^{-63} \times 2^{-16382}$	3.645×10^{-4951}
Smallest positive normalized	2^{-16382}	3.362×10^{-4932}
Largest normalized	$(2 - 2^{-63}) \times 2^{16383}$	1.190×10^{4932}

bias=2^{14} -1=16383

Smallest positive denormalized:

E=1-16383=-16382, M=0.00···1=2^{-63}

Smallest positive normalized:

Exp=000···1=1, E=exp-16383=-16382, M=1.000···0=1

Largest normalized:

Exp=11···10, E=exp-16383=0111···11=2^{14}-1=16383, M=1.1111···111=2-2^{-63}

2.87 ◆◆

Consider the following two 9-bit floating-point representations based on the IEEE floating-point format.

1. Format A

There is one sign bit.

There are k = 5 exponent bits. The exponent bias is 15.

There are n = 3 fraction bits.

2. 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*** **+∞**. In addition, give the values of numbers given by the Format A and Format B bit patterns. Give these as whole numbers (e.g., 17) or as fractions (e.g., $\frac{17}{64}$ or $\frac{17}{2^6}$).

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	$\frac{5}{2^{17}}$	0 0000 0001	$\frac{1}{1024}$
1 11011 000	$-2^{12} = -4096$	1 1110 1111	-248
0 11000 100	$3 \times 2^8 = 768$	0 1111 0000	$+\infty$

Format A

bias=2^{4}-1=15,

1 01110 001, s=1, E=14-15=-1, M=1.001=9/8

0 10110 101, s=0, E=22-15=7, M=1.101=13/8

1 00111 110, s=1, E=7-15=-8, M=1.110=14/8

0 00000 101, s=0, E=1-15=-14, M=0.101=5/8

1 11011 000, s=1, E=27-15=12, M=1.000=8/8

0 11000 100, s=0, E=24-15=9, M=1.100=12/8=3/2

Format B

bias=2^{3}-1=7

1 0110 0010, s=1, E=6-7=-1, M=1.0010=9/8

0 1110 1010, s=0, E=14-7=7, M=1.1010=13/8

1 0000 0111, s=1, E=1-7=-6, M=0.0111=7/16

0 0000 0001, s=0, E=1-7=-6, M=0.101=1/16

1 1110 1111, s=1, E=14-7=7, M=1.1111=31/16
0 1111 0000, inf