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 $Y = B2U_k(y)$, 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:

Y = m/(2^k – 1),m is the integer represented in binary form by y(for example, when it comes to 1/5, y = 0011, m = 3, k = 4, Y = 1/5 = 3/(2^4-1)

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

(a) 001

1/7

(b) 1001

0.6

(c) 000111

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^(-61-2^14)	3.645*10^-4951
Smallest positive normalized	2^(2-2^14)	3.362*10^-4932
Largest normalized	2^(2^14-1)*(2-2^-63)	1.1897*10^4932

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* $+\infty$. 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	-7/1024	1 0000 0111	-7/1024
0 00000 101	5/131072	0 0000 0000	0
1 11011 000	-4096	1 1111 0000	$-\infty$
0 11000 100	768	0 1111 0000	$+\infty$