Hw2

2.83 ◆◆

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

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.

B. What is the numeric value of the string for the following values of y?
   (a) 101
   (b) 0110
   (c) 010011

A.

```
n = 0.yyyyyyy...
Y = str<<k - str;
n=Y/(2^k-1)
```

$$n = Y/(2^k - 1)$$

B.

a. 5/7 = 0.714285714285….

b. 6/15 = 0.4

c. 19/63=0.301587301587…

> 以上为第三版，接下来为作业版本

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

(a) 001

(b) 1001

(c) 000111

B.

a. 1/7

b. 9/15 = 3/5

c. 7/63 = 1/9

2.86 ◆

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	________	________
Smallest positive normalized	________	________
Largest normalized	________	________

This format can be used in C programs compiled for Intel-compatible machines by declaring the data to be of type `long double`. However, it forces the compiler to generate code based on the legacy 8087 floating-point instructions. The resulting program will most likely run much slower than would be the case for data type `float` or `double`.

$Bias = 2^{15-1} - 1$ = 1024*16-1

Description	Extended precision Value	Extended precision Decimal
Smallest positive denormalized	000000…0000….00001	$2^{-Bias+1-63}$
Smallest positive normalized	00000…0000011000…000	2^{1-Bias}
Largest normalized	011111…1111101111111…11111	$2^E \times M = 2^{Bias} \times (2 - 2^{63})$

Format A		Format B	
Bits	Value	Bits	Value
1 01110 001	$-\frac{9}{16}$	1 0110 0010	$-\frac{9}{16}$
0 10110 101	$2^7 \times \frac{(5+8)}{2^3} = 13 \times 2^4$	0 1100 1010	13×2^4
1 00111 110	$-2^{-8} \times \frac{14}{8} =$ -$2^{-10} \times 7$	1 0000 0111	-$2^{-10} \times 7$
0 00000 101	$2^{-14} \times \frac{5}{8} = 2^{-17} \times 5$	0 0000 0001	1×2^{-10}
1 11011 000	$-2^{12} \times 1 = -2^{12}$	1 110 1111	-31×2^3
0 11000 100	$2^9 \times \frac{12}{8} = 3 \times 2^8$	0 1111 0000	$+\infty$