

Homework 7

Floating Point

1. Fill the table:

Fractional value	Binary representation	Decimal representation
5/8	0.101	0.625
		3.34375
		0.90625
3/16		
	1.010101	

2. Consider a 16-bit floating-point representation based on the IEEE floating-point format, with 1 sign bit, 6 exp bits, 9 frac bits, called **Float16**.

(1) Fill in the following table, please represent M in the form **x** or **x/y** where x is an integer and y is an integral power of 2, and represent Value in the form **a** or **a * 2^b** where a and b are integers.

Description	Hex	M	E	Value
-0				--
Largest negtive Normalized value				
$+\infty$		--	--	--
Largest Denormalized value				
Number with hex representation 0x4BF7	0x4BF7			
Sample	0xC4A0	21/16	3	$-21 \cdot 2^{-1}$

(2) Assume we use **IEEE round-to-even** mode to do the approximation, let a = 0x33CD, b = 0x2F89, compute a + b and a * b. The answer should also be represented in **Float16**.

3. Translate the following assembly into C codes. You can name the arguments and variables freely as you like.

```
pushq %rbp
movq %rsp, %rbp
movl %edi, -4(%rbp)
vmovss %xmm0, -8(%rbp)
movq %rsi, -16(%rbp)
vmovsd %xmm1, -24(%rbp)
vxorpd %xmm0, %xmm0, %xmm0
vcvttsi2sd -4(%rbp), %xmm0, %xmm0
movq -16(%rbp), %rax
vmovsd (%rax), %xmm1
vaddsd %xmm1, %xmm0, %xmm1
```

```
vcvtss2sd -8(%rbp), %xmm0, %xmm0
vmovsd -24(%rbp), %xmm2
vsubsd %xmm0, %xmm2, %xmm0
vmulsd %xmm0, %xmm1, %xmm0
popq    %rbp
ret
```