1. Convert each of the following decimal numbers to a 10-bit binary number using **signed magnitude** form. Clearly indicate the signed bit followed by the 9 “magnitude” bits.

a) -16

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	0	0	1	0	0	0	0

b) -78

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	1	0	0	1	1	1	0

c) 35

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	0	0	0	1	1

2. Convert each of the following decimal numbers to a 10-bit binary number using **one’s compliment** form. Clearly indicate the signed bit followed by the 9 “magnitude” bits.

a) -16

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	1	1	0	1	1	1	1

b) -78

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	1	0	0	0	1

c) 35

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	0	0	0	1	1

3. Convert each of the following decimal numbers to a 10-bit binary number using **two's compliment** form. Clearly indicate the signed bit followed by the 9 "magnitude" bits.

a) -16

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	1	1	1	0	0	0	0

b) -78

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	1	0	0	1	0

c) 35

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	0	0	0	1	1

d) 90

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	0	1	1	0	1	0

e) -389

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	0	0	0

f) 480

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	1	1	1	0	0	0	0	0

g) -123

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	0	0	0	1	0	1

h) -205

Sign	Magnitude								
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	0	1	1	0	0	1	1

4. Convert each of the following binary numbers in floating point format to the corresponding decimal numbers.

a)

S	Exponent								Mantissa																						
0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Format: S Mantissa * 2^{Exponent}

= +11100 * 2^{101}

= +28 * 2^5

= +28 * 32

= +896

b)

S	Exponent								Mantissa																						
1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Format: S Mantissa * 2^{Exponent}

= -1111 * 2^{11}

= -15 * 2^3

= -15 * 8

= -120

c)

S	Exponent								Mantissa																						
1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0

Format: S Mantissa * 2^{Exponent}

= $-110101010 * 2^{11111101}$

= $-426 * 2^{-3}$

= -15 * 8

= **-53.25**

<u>Alternate way to convert a floating point value:</u>

Exponent: $11111101_2 = -3$

Mantissa: 110101010**.** => 110101**.**010 = $1*2^5 + 1*2^4 + 1*2^2 + 1*2^0 + 1*2^{-2}$ = 32 + 16 + 4 + 1 + ¼ = 53.25

After moving binary point
to the left 3 spaces

Finally apply the sign: **-53.25**

d)

S	Exponent								Mantissa																						
0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0

Format: S Mantissa * 2^{Exponent}

= $+1101100 * 2^{11101110}$

= $+108 * 2^{18}$

= +108 * 262,144

= +28,311,552

5. Convert each of the following decimal numbers to a floating point number using the format discussed in class.

a) 52.0

Whole number 52 is 110100_2

Fractional part 0 is 0_2

Therefore the binary value is just 110100 (there is no fractional part)

We can move the binary point left 2 spaces to the first bit that's a 1. This would give us a mantissa of 1101 (or 1101.00 if we show the binary point)

This would also give an exponent of +2 since we must move the binary point 2 spaces to the right to restore the original number of 110100

S	Exponent								Mantissa																						
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

b) -829.0

Whole number 829 is 1100111101_2

Fractional part 0 is 0_2

Therefore the binary value is just 1100111101 (there is no fractional part)

We don't really have to move the binary point, we can just leave the exponent as 0 and the mantissa can be 1100111101

Don't forget to make the sign a 1 since we are dealing with a negative number

S	Exponent								Mantissa																						
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	1

c) 0.5

Whole number 0 is 0_2

Fractional part 0.5 is:

Value	Times	Two	Equals	Product	Left part	Right part
.5	*	2	=	1.0	1	.0

So fractional part is 1_2

We then get 0.1 as the binary representation

Moving the binary point to the right one spot gives us 01 (or just 1). This implies the exponent is -1 since we must move it to the left one spot to restore 01 to 0.1

S	Exponent								Mantissa																						
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

d) -0.625

Whole number 0 is 0_2

Fractional part 0.625 is:

Value	Times	Two	Equals	Product	Left part	Right part
.625	*	2	=	1.25	1	.25
.25	*	2	=	.5	0	.5
.5	*	2	=	1.0	1	0

So fractional part is 101_2

We then get 0.101 as the binary representation

Moving the binary point to the right three spots gives us 101. This implies the exponent is -3 since we must move it to the left three spots to restore 101 to 0.101

S	Exponent								Mantissa																						
1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

e) 0.923

Whole number 0 is 0_2
Fractional part 0.923 is:

Value	Times	Two	Equals	Product	Left part	Right part
.923	*	2	=	1.846	1	.846
.846	*	2	=	1.692	1	.692
.692	*	2	=	1.384	1	.384
.384	*	2	=	0.768	0	.768
.768	*	2	=	1.536	1	.536
.536	*	2	=	1.072	1	.072
.072	*	2	=	0.144	0	.144

We can see how sometimes we can only approximate a value, but not fully represent it. Let's stop here as a close enough approximation. Obviously, to be as accurate as possible, we should use all 23 bits of the mantissa (i.e. 23 rows in the table), but I think you get the idea of the process above, so we can stop here.

So fractional part is 1110110_2

We then get 0.1110110 as the binary representation

Moving the binary point to the right seven spots gives us 1110110. This implies the exponent is -7 since we must move it to the left seven spots to restore 1110110 to 0.1110110

S	Exponent								Mantissa																						
0	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0

f) 0.8515625

Whole number 0 is 0_2
Fractional part 0.8515625 is:

Value	Times	Two	Equals	Product	Left part	Right part
.8515625	*	2	=	1.703125	1	.703125
.703125	*	2	=	1.40625	1	.40625
.40625	*	2	=	0.8125	0	.8125
.8125	*	2	=	1.625	1	.625
.625	*	2	=	1.25	1	.25
.25	*	2	=	0.5	0	.5
.5	*	2	=	1.0	1	0

So fractional part is 1101101_2

We then get 0.1101101 as the binary representation

Moving the binary point to the right seven spots gives us 1101101. This implies the exponent is -7 since we must move it to the left seven spots to restore 1101101 to 0.1101101

S	Exponent								Mantissa																						
0	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1

g) -92.578125

Whole number 92 is 1011100_2

Fractional part 0.578125 is:

Value	Times	Two	Equals	Product	Left part	Right part
.578125	*	2	=	1.15625	1	.15625
.15625	*	2	=	0.3125	0	.3125
.3125	*	2	=	0.625	0	.625
.625	*	2	=	1.25	1	.25
.25	*	2	=	0.5	0	.5
.5	*	2	=	1.0	1	0

So fractional part is 100101_2

We then get 1011100.100101 as the binary representation

Moving the binary point to the right six spots gives us 1011100100101. This implies the exponent is -6 since we must move it to the left six spots to restore 1011100100101 to 1011100.100101

S	Exponent								Mantissa																						
1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1	0	0	1	0	1