Fixed-Point Number (EXERCISE)

Examples of 2's complement I1.Q15 fractions

0.100 0000 0000 0000 has the value 1/2

0.000 0000 0000 0001 has the value 1/(2^15)

0.111 1111 1111 1111 has the value = 1-1/(2^15)

1.000 0000 0000 0000 has the value -1

1.100 0000 0000 0000 has the value -1/2

1.111 1111 1111 1111 has the value -1/(2^15)

More examples of 2's complement fractions

0100 0.100 0000 0000 has the value

0000 0000 00.00 0001 has the value

0111 1111 111.1 1111 has the value

1111 0000 0.010 0000 has the value

1111 1111 1111 111.1 has the value

1000 00.00 0000 0000 has the value

Two's complement for fixed point numbers

e.g. 0110.1000 which is 6.5 in decimal

How do we represent -6.5 in fixed point ?

```
  0110.1000
  1001.0111   <----- invert bits
+ 0000.0001   <----- add .0001
  0000.0000
```

Thus,

```
  1001.0111   <----- invert bits
+ 0000.0001   <----- add .0001
  1001.1000   <-----  answer: -6.5 in (signed) fixed point
```

Overflow

What happens if we add two 2's complement numbers (I1.Q5) 0.75 and 0.5?
The sum is 1.25.

		-0.75+0.5 =	-0.75+-.5
1 1	0 1	0 0	1 0
0.11000	0.11000	1.01000	1.01000
+1.10000	+0.10000	+0.10000	+1.10000
10.01000	01.01000	01.11000	10.11000
1 xor 1 = 0, ans = 00.01000	0 xor 1 = 1, ans = 01.01000	0 xor 0 = 0, ans = 11.11000	1 xor 0 = 1, ans = 10.11000

- How do you deal with this problem?
  - We can use more bits, i.e., I2.Q5 representation

C(n) xor C(n-1) = 0, No overflow, discard cout bit and sign extend
C(n) xor C(n-1) = 1, Overflow, keep cout

Consider two fixed-point numbers A and B where, A = 3/4 and B = -1/4. **Use 2's complement number, I1.Q3 format.**

$I1.Q3 \times I1.Q3 = I(1+1-1).Q(3+3+1)$
$= I1.Q7$

$\frac{1}{4} = 0.25 = 0.010 \Rightarrow -\frac{1}{4} = \begin{array}{r} 1.101 \\ +\quad 1 \\ \hline 1.110 \end{array}$

```
   3/4        0.110
x -1/4        1.110
-------------------
        000 0000
        00 0110
        0 0110
        1010
-------------------
- 3/16  11101000
```

negative

Sign extension

$2's(0110) = \begin{array}{r} 1001 \\ +\quad 1 \\ \hline 1010 \end{array}$

discard carry-out

I1.Q7 ⇒ 8-bits ⇒ $\boxed{1.1101000}$

$= -\frac{3}{16}$

$\begin{array}{r} 0.0010111 \\ +\qquad\quad 1 \\ \hline 0.0011000 \end{array} = 2^{-3} + 2^{-4} = \frac{3}{16}$

Consider two fixed-number numbers A and B where, A = ~~3/4~~ -1/4 and B = -1/4. **Use 2's complement number, I1.Q3 format.**

Consider two fixed-point numbers A and B where, A = 19.25 and B = 27.50.

Use 2's complement number, I8.Q4 format, for both A and B.

1. What is your **I8.Q4 format** result for A?
2. What is your **I8.Q4 format** result for B?
3. Calculate A + B and find your final **I9.Q4 format** result. Note: show your bit operation.
4. What is your **I8.Q4 format** result for -A?
5. What is your **I8.Q4 format** result for -B?
6. Calculate A – B and find your final **I9.Q4 format** result. Note: show your bit operation.
7. Calculate B – A and find your final **I9.Q4 format** result. Note: show your bit operation.

1. A=0001 0011.0100
2. B=0001 1011.1000

3.
```
  0001 0011.0100
+ 0001 1011.1000
0 0010 1110.1100
 0 xor 0 = 0
 ans = 0 0010 1110.1100
```

4. -A=1110 1100.1100
5. -B=1110 0100.1000

6.
```
  0001 0011.0100
+ 1110 0100.1000
0 1111 0111.1100
 0 xor 0 = 0
 ans = 1 1111 0111.1100
```

7. 11
```
  0001 1011.1000
+ 1110 1100.1100
1 0000 1000.0100
 1 xor 1 = 0
 ans = 0 0000 1000.0100
```

Consider two fixed-number numbers A and B where, A = 19.25 and B = 27.50.

Use 2's complement number, I8.Q4 format, for both A and B.

8. Calculate A x B and find your final **I15.Q9 format** result. Note: show your bit operation. 1)
9. Calculate (B) x (-A) and find your final **I15.Q9 format** result. Note: show your bit operation.
10. Calculate (-B) x (-A) and find your final **I15.Q9 format** result. Note: show your bit operation.

A=0001 0011.0100 -A=1110 1100.1100

B=0001 1011.1000 -B=1110 0100.0111

```
8)               0001 0011.0100
                x0001 1011.1000

                  000000000000
                 000000000000_
                000000000000__
               000100110100___
              000100110100____
             000100110100_____
            000000000000______
           000100110100_______
          000100110100________
         000000000000_________
        000000000000__________
      +000000000000___________
       00000011023223221100000

       00000100001000101100000

0000 0100 0010 001.0 1100 0000
```

```
9)               1110 1100.1100
                x0001 1011.1000

                  000000000000
                 000000000000_
                000000000000__
       11111111111011001100___
       1111111111011001100____
       111111111011001100_____
            000000000000______
       1111111011001100_______
       111111011001100________
         000000000000_________
        000000000000__________
      +000000000000___________
       55555544532342122100000

       11111011110111010100000

1111 1011 1101 110.1 0100 0000
```

```
10)              1110 1100.1100
               x 1110 0100.0111

       11111111111111011001100
       1111111111111011001100_
       111111111111011001100__
               000000000000___
              000000000000____
             000000000000_____
       11111111011001100______
           000000000000_______
          000000000000________
       11111011001100_________
       1111011001100__________
      +111011001100___________
```