COMP 273 Assignment 1
Chelsea Jin-Yeong Ma
260515648

Question 1:

(a) Description of fraction data structure in binary:
Suppose that size of integer is 16 bits. Since fraction consists of two integers, I would store two signed integers using 2's complement.

(b) Example:
3/5 is stored as 00000000 00000011 00000000 00000101

(c) If I want to create 10 fraction numbers, each fraction has two integers so I would store 20 signed integers. Suppose size of integer is 16 bits. Then I would need 20*16=320 bits of memory to use.

(d) Discuss representation and operations:
My fraction has limited range since the range of integer is also limited.
In addition operation, multiply a fraction (numerator and denominator) with other fraction's denominators and repeat this for the other fractions. Then add numerators only.
In subtraction operation, the negative sign will be handled by 2's complement and follow the same procedure as addition operation.
In multiplication, I would simply multiply numerators with numerators and denominators with denominators.
In division, I would reverse the positions of the numerator and denominator of second fraction and do multiplication.
My fraction would be as efficient as single precision of fixed point because their sizes are both 32 bits.

Question 2:

(a) $92A5F_{16}$
Decimal: 9*16^4 + 2*16^3 + A*16^2+ 5*16^1 + F*16^0
= 9*16^4 + 2*16^3 + 10*16^2+ 5*16^1 + 15*1
= 600671_{10}
Binary: 600671_{10} = 600671/2 = 300335 R 1
= 300335/2 = 150167 R 1
= 150167/2 = 75083 R 1
= 75083/2 = 37541 R 1
= 37541/2 = 18770 R 1
= 18770/2 = 9385 R 0
= 9385/2 = 4692 R 1
= 4692/2 = 2346 R 0
= 2346/2 = 1173 R 0
= 1173/2 = 586 R 1
= 586/2 = 293 R 0
= 293/2 = 146 R 1
= 146/2 = 73 R 0
= 73/2 = 36 R 1

= 36/2 =18 R 0
= 18/2 = 9 R 0
= 9/2 = 4 R 1
= 4/2 = 2 R 0
= 2/2 = 1 R 0
= 1/2 = 0 R 1
$=1001001010100101111_2$

100010110101_2

Decimal: 1*2^11 + 1*2^7 + 1*2^5 + 1*2^4 + 1*2^2 + 1* 2^0

$= 2229_{10}$

Hex: 2229_{10} = 2229/16 = 139 R 5
= 139/16 = 8 R B
= 8/16 = 0 R 8
$= 8B5_{16}$

339_{10}

Binary: 339/2 =169 R 1
=169/2 = 84 R 1
= 84/2 = 42 R 0
= 42/2 = 21 R 0
= 21/2 = 10 R 1
= 10/2 = 5 R 0
= 5/2 = 2 R 1
= 2/2 = 1 R 0
= 1/2 = 0 R 1
$=101010011_2$

Hex: 339/16 = 21 R 3
=21/16 = 1 R 5
=1/16 = 0 R 1
$=153_{16}$

(b) 110_{16} = 1*16^2 + 1*16^1
$= 272_{10}$

272_{10} = 272/2 = 136 R 0
= 136/2 = 68 R 0
= 68/2 = 34 R 0
= 34/2 = 17 R 0
= 17/2 = 8 R 1
= 8/2 = 4 R 0
= 4/2 = 2 R 0
= 2/2 = 1 R 0
= 1/2 = 0 R 1
$=100010000_2$

I am Mary. = 73 32 97 109 32 77 97 114 121 46

73 = 73/2 = 36 R 1
= 36/2 = 18 R 0
= 18/2 = 9 R 0
= 9/2 = 4 R 1
= 4/2 = 2 R 0
= 2/2 = 1 R 0

= 1/2 = 0 R 1
=1001001_2
32 = 32/2 = 16 R 0
= 16/2 = 8 R 0
= 8/2 = 4 R 0
= 4/2 = 2 R 0
= 2/2 = 1 R 0
= 1/2 = 0 R 1
= 100000_2
97 = 97/2 = 48 R 1
= 48/2 = 24 R 0
= 24/2 = 12 R 0
= 12/2 = 6 R 0
= 6/2 = 3 R 0
= 3/2 = 1 R 1
= 1/2 = 0 R 1
=1100001_2
109 = 109/2 = 54 R 1
= 54/2 = 27 R 0
= 27/2 = 13 R 1
= 13/2 = 6 R 1
= 6/2 = 3 R 0
= 3/2 = 1 R 1
= 1/2 = 0 R 1
=1101101_2
77= 77/2 = 38 R 1
= 38/2 = 19 R 0
= 19/2 = 9 R 1
= 9/2 = 4 R 1
= 4/2 = 2 R 0
= 2/2 = 1 R 0
= 1/2 = 0 R 1
= 1001101_2
114=114/2 = 57 R 0
= 57/2 = 28 R 1
= 28/2 = 14 R 0
= 14/2 = 7 R 0
= 7/2 = 3 R 1
= 3/2 = 1 R 1
=1/2 = 0 R 1
=1110010_2
121 = 121/2 = 60 R 1
= 60/2 = 30 R 0
= 30/2 = 15 R 0
= 15/2 = 7 R 1
= 7/2 = 3 R 1
= 3/2 = 1 R 1
= 1/2 = 0 R 1
=1111001_2
46= 46/2 = 23 R 0
= 23/2 = 11 R 1
= 11/2 = 5 R 1
= 5/2 = 2 R 1
= 2/2 = 1 R 0
= 1/2 = 0 R 1
=101110_2

I am Mary. (with null):

RAM

ADDRESS	DATA

100010000	01001001	00100000	01100001	01101101	00100000
	01001101	01100001	01110010	01111001	00101110
	00000000				

Question 3:

(a) Pseudo algorithm that moves 100 bytes from RAM to slot:

```
// Creating starting address
void *B= 0x200
void *C = 0x0F
//Transferring 100 bytes
for i=0 to 100
LOAD A, B
SAVE A, C
B++;
```

(b) Tracing path of bytes:

Bytes will start its path from RAM (at address 200_{16}) to Data Register, then to System bus, to MBR, to CPU bus, to Register 1, to CPU bus, to MBR, to System bus, to Data Register, to RAM (at address $0F_{16}$) and at last, the slot.