Question 1

a) To represent the fraction, we'll use two integers (8 bytes) and 2 characters (2 bytes); that is 10 bytes in total.
   1) The first integer will represent the numerator.
   2) The char '/' (ASCII code)
   3) The second integer will represent the dominator, the sign of fraction will be presented in this number. // if the number after the '/' is zero, it will be treated as an error.
   4) The null character to indicate the end of the fraction.

b) an example, the fraction 240/4 stored at address / little endian

Adresse	byte (value)
00001000	11110000
00001001	00000000
00001010	00000000
00001011	00000000
00001100	00101111
00001101	00000100
00001110	00000000
00001111	00000000
00010000	00000000
00010001	00000000

c) A fraction takes about 10 bytes in total, so ten fractions will take 10 * 10 = 100 bytes.

d) This representation have an impact in arithmetic; for addition and subtraction, three multiplication operations happen before an addition operation. As for multiplication and division, only two multiplication operations occur.

As for the range it is the same for a 32 bit integer.

Question 2:

a)

1) $154CD_{16}$ to binary:

$D_{16} = 1101_{(2)}$, $C_{16} = 1100$, $4_{(10)} = 0100$, $5_{16} = 0101$

$1_{(16)} = 0001$.

So $154CD_{16} = 0001\,0101\,0100\,1100\,1101$

$154CD_{16}$ to decimal:

~~$1_{16} = A_{10}$, $5_{16} = 5_{10}$, $4_{16} = 4_{10}$, $C_{16} = 12_{10}$, $D_{16} = 13_{10}$.~~

$154CD_{16} = (13 \times 16^0) + (12 \times 16) + (4 \times 16^2) + (5 \times 16^3) + (1 \times 16^4)$

$= 87245_{(10)}$

2) $1001101111111_{(2)}$.

. to decimal

$1001101111111_{(2)} = 1\times2^0 + 1\times2 + 1\times2^2 + 1\times2^3 + 1\times2^4 + 1\times2^5 + 0\times2^6 + 1\times2^7 + 1\times2^8 + 0\times2^9 + 0\times2^{10} + 1\times2^{11}$

$= 2495_{(10)}$.

. to Hex:

1001 1011 1111

9 B F

$\rightarrow 1001\,1011\,1111_{(2)} = 9BF_{(16)}$.

3) $1025_{(10)}$:

we know that $2^{10} = 1024$.

$1024_{(10)} = 10000000000_{(2)}$.

So $1025_{(10)} = 10000000001_{(2)}$

to Hex;

we'll use the binary representation.

0100 0000 0001

4 0 1

$\Rightarrow 1025_{(10)} = 401_{(16)}$.

B) The memory taken up by The C string “Hello World” is:

Address	Value	
0b1001010101	0b01001000	H
0b1001010110	0b01100101	e
0b1001010111	0b01101100	l
0b1001011000	0b01101100	l
0b1001011001	0b01101111	o
0b1001011010	0b00100000	
0b1001011011	0b01010111	W
0b1001011100	0b01101111	o
0b1001011101	0b01110010	r
0b1001011110	0b01101100	l
0b1001011111	0b01100100	d
0b1001100000	0b00101110	.
0b1001100001	0b00000000	

Question 3

a)

```
for ( int i = 0; i < 100; i++){
      LOAD R1, #200 + i;
      SAVE R1, #0F ;
      }
```

b)

The first step is reading the byte;
Following this line: LOAD R1, #200 + i
The CPU will send a read request, The data section of the bus will be emptied and the address section will be filled with the address of the byte that is intended to be read. The write/read bit is set to read. After that, the RAM will respond to the read request by filling the data section of the bus with the value stored at that address. The CPU will copy the data into the registry then, R1 in our example

The second step is saving the byte// writing the byte.
After this line : SAVE R1, #0F ;
The CPU will send a write request to the system bus, It will copy the data section of the bus with the data in the registry and the address section will be set to where we want to save it, #0F is this case. The write/read bit is set to write. The data section of the bus will be written in the address (#0F), the data will go then from the slot to the device.