$-12.5 \equiv -1.25\times10^{1}$ or -25×2^{-1}

$\equiv -11001_2 \times 2^{-1}$

$\equiv -1.1001_2 \times 2^{3}$ when normalised.

$0.1 \equiv 1.0\times10^{-1}$ or $\frac{1}{10} \equiv \frac{1}{1010_2} \equiv \frac{0.1_2}{101_2}$

so now we need to do some binary long division...

```
       0.0001100
101 / 0.1000000000000000
         101
          110
          101
           1000
            101     at this point
                    we see that the
                    fraction recurs
```

$= \frac{0.1_2}{101_2} = 0.000\dot{1}10\dot{0}_2$

$\underline{1.\dot{1}00\dot{1}100 \times 2^{-4}}$ answer.

So how do we use the above results to represent these numbers in the computer? Well, firstly we reserve 4 bytes of storage for each real in the following format:

```
[sign]   [normalised mantissa]   [exponent]   [data]
 23       22                0     7        0   bit
 \________________/\___________________________/
    H          L          E          D         register
```

sign: the sign of the mantissa: 1 = negative, 0 = positive.

normalised mantissa: the mantissa normalised to the form 1.xxxxxx with the top bit (bit 22) always 1 except when representing zero (HL=0, DE=0).

exponent: the exponent in binary 2's complement form.

Thus:

2	≡	0	1000000	00000000	00000000	00000001	(£40,£00,£00,£01)
1	≡	0	1000000	00000000	00000000	00000000	(£40,£00,£00,£00)
-12.5	≡	1	1100100	00000000	00000000	00000011	(£E4,£00,£00,£03)
0.1	≡	0	1100110	01100110	01100110	11111100	(£66,£66,£66,£FC)