MT4113 Lecture 4 Computer arithmetic
Solutions to practice questions

1. Write out -15-3=-18 in binary using an 8-bit signed integer system

```
  -15 10001111
-  +3 00000011
= -18 10010010
```

2. What is the largest positive number that can be represented in this system?

   `01111111 = 127`

3. Why is it very fast to multiply numbers represented as integers by 2? What manipulation is required to the bits making up the number to achieve this operation?

   `The computer just needs to shift the bits to the left (apart from the sign bit and the most significant digit - if the latter is already 1 then you get an overflow), and put a 0 in the least significant digit bit`

4. What is the machine epsilon on a 32-bit floating point system where 1 bit is for the sign, 16 bits for the exponent and 16 bits[1] for the fraction?

   `epsilon =` $2^{(1-d)} = 2^{(-15)}$

5. In this system, what would be the result of computing $2 * 2^{-17} + 1$?

   $2 \times 2^{-17} = 1 \times 2^{-}16$. `This is below the machine epsilon so the result would be 1.`

6. What do you get if you compute $512 * 0.25$ in the above floating point system, and then convert it into the above signed integer system?[2]

   512×0.25 `is easily evaluated exactly in the floating point system, giving 128. This is larger than the largest integer, however, so you'd get an overflow`

[1] using the same trick as in the IEEE standard to get one extra bit for free; note that this is not a very sensible allocation as it gives far too much of the space to the exponent

[2] A calculation like this once cost the EU space program about $500 million: https://www.ima.umn.edu/~arnold/disasters/ariane.html