1. Format

What are the effect of decreasing the number of bits allocated to Mantissa and increasing the exponent

1. Reduction in precision
    - As the number of bits in mantissa has decreased.
2. Increasing in range
    - As the number of bits in exponent has increased.

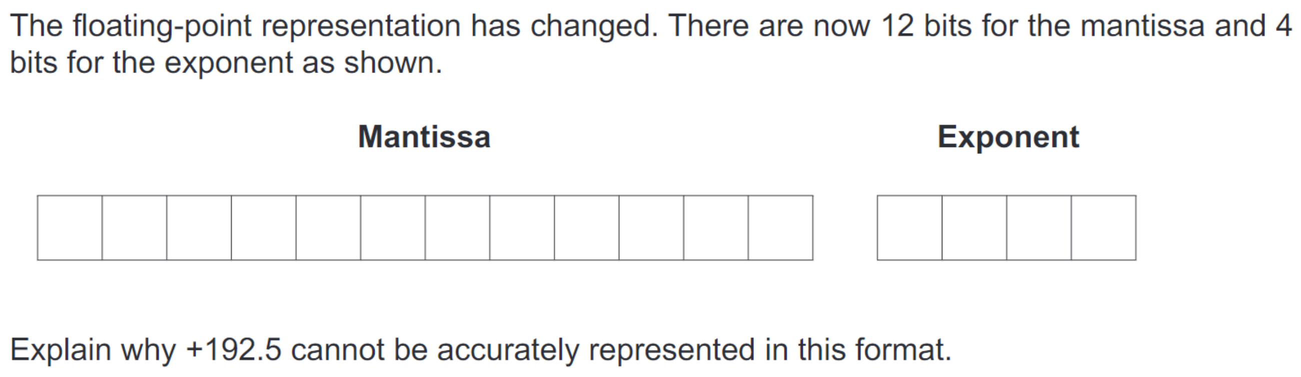
The denary number 513 cannot be stored accurately as normalised floating-point number in this computer system: (10 bits for mantissa, 6 bits for exponent)

Explain reasons for this:

- Require more than 10 bits/11bits to store; the maximum number that can be stored is 511
- The denary 513 in binary is 1000000001 // Normalised: 0.1000000001
- Results in **overflow**

Describe an alteration to the way floating-point numbers are stored to enable this number to be stored accurately with the same total number of bits:

- The number of bits for mantissa must be increased
- 11 bits for mantissa and 5 bits for exponent



The floating-point representation has changed. There are now 12 bits for the mantissa and 4 bits for the exponent as shown.

Mantissa
Exponent

Explain why +192.5 cannot be accurately represented in this format.

- Exponent too large to fit 4 bits as two’s complement number
- Exponent will turn negative
- … therefore the binary point moves the wrong way
- Value will be approximately +0.029(296875)

Explain the trade-off between either using a large number of bits for the mantissa, or a large number of bits for the exponent

- The trade-off is between range and precision
- Any increase in the number of bits for the mantissa means fewer number of bits for the exponent
- More bits used in mantissa would result in better precision
- More bits used in exponent would result in larger range