Two’s Complement

Negative Numbers

Floating Point Notation

Mantissa & Exponent

National 5 Binary

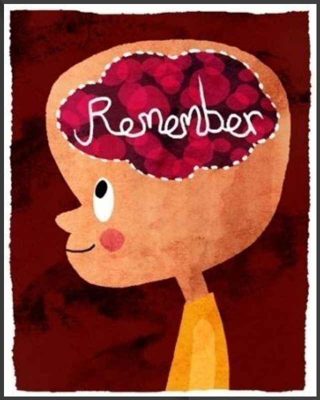
Two's Complement

- This is the name given to binary system we use to represent negative numbers.

- The first bit used in two's complement is -128 instead of 128 we usually have:

-128	64	32	16	8	4	2	1

Note!!



- If you are given a two's complement question which begins with a '1' then your answer will be negative

- If you are given two's complement question which begins with a '0' then your answer will be positive

Negative Numbers

- For example the number -45 us represented by 11010011 In two's complement

-128	64	32	16	8	4	2	1
1	1	0	1	0	0	1	1

= -128 + 64 + 16 + 2 + 1 = -45

Negative Numbers

- For example the number -81 us represented by 10101111 In two's complement

-128	64	32	16	8	4	2	1
1	0	1	0	1	1	1	1

= -128 + 32 + 8 + 4 + 2 + 1 = -81

Floating Point Notation

- This is the name given to the binary system used to represent numbers with a decimal point.

- For example :
  - 33.9
  - 0.0056
  - 1289.1285

Mantissa & Exponent

- A floating point number is made up of:
  - Mantissa : the ‘fraction’ part
  - Exponent : the ‘power of’ part
- For example:

$$100111.10100 = 10011110100 \times 2^{0110}$$

Mantissa

Exponent

Example

11001110.1101

1 1 0 0 1 1 1 0. 1 1 0 1

8 7 6 5 4 3 2 1

Step 1 – count out how many steps until the decimal point is out of the number

$110011101101 \times 2^{8}$

Step 2 – write out your number without the decimal point x 2 to the power of how many steps it took to remove the decimal point

$110011101101 \times 2^{1000}$

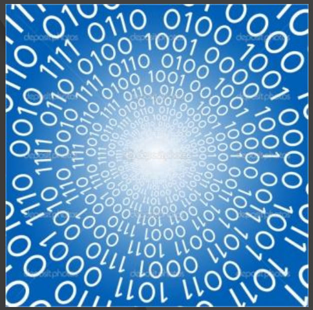
Step 3 – rewrite the power of as a binary number as the computer does not understand an ‘8’

Floating Point Notation

- The accuracy of a floating point number is increased by allocating more bits to the mantissa.

- The range of numbers that can be stored is increased by allocating more bits to the exponent.

Examples



- A floating point number which uses 16 bits for the mantissa and 8 bits for the exponent is less accurate and stores a smaller range of numbers than a floating point number that uses 24 bits for the mantissa and 16 bits for the exponent.

Question

- Jonathan needs to store the floating point numbers accurately in his program.

Which option should he use and why?

Option 1	16-bit exponent	16-bit mantissa
Option 2	8-bit exponent	24-bit mantissa