1.4 DATA TYPES & STRUCTURES · 1.4.1 A-LEVEL

Floating point numbers

Representation, normalisation & arithmetic. Spec 1.4.1(g)(h).

01 Representation

Value	**mantissa × 2^exponent**.
Mantissa	Two's complement; point after the sign bit. → precision.
Exponent	Two's complement signed integer. → range.
More bits	Mantissa = precision; exponent = range.

02 FP → denary

Step 1	Read exponent (signed).
Step 2	Move point: + → right, – → left.
Step 3	Evaluate the binary (two's comp if negative).
Example	`0.1011100` e `0011` = 101.11 = 5.75

03 Normalisation

Why	**Maximum precision** for the bits available.
Positive	Mantissa starts `0.1…`
Negative	Mantissa starts `1.0…`
Rule	First two bits **differ**; adjust exponent to keep value.
13.25	`0.1101010` e `0100`

04 Addition

Step 1	Make **exponents equal** (shift smaller one's mantissa).
Step 2	Add mantissas (two's complement).
Step 3	**Renormalise** the result; watch for overflow.
Mantra	Align → add → renormalise.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on floating point questions.

01

The exponent is **two's complement** — a leading 1 means move the point **left**.

02

Normalised: positive `0.1…` , negative `1.0…` — first two bits differ.

03

Shift the mantissa? **Adjust the exponent** to keep the value.

04

Adding: align exponents **first**, then add mantissas.

05

Precision = **mantissa** bits; range = **exponent** bits.

06

Use the mantissa/exponent **sizes the question gives** — they vary.