Exercise walk-through

Floating-point numbers, representation and manipulation ▪ 13.3

eXercise

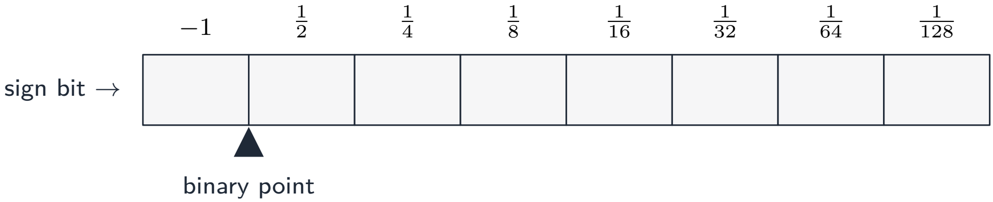
You must be able to

- describe the **mantissa** + **exponent** format
- convert **binary** $\leftrightarrow$ **denary** floating-point
- **normalise** a number and explain **rounding errors**

Method — The format

A floating-point number is mantissa $\times\ 2^{\text{exponent}}$. Both are stored in **two's complement**. The mantissa is a fraction with the binary point just after the **sign bit**:

Method — The format



Method — Binary $\to$ denary

Mantissa 01010000, exponent 00000010:

- sign bit 0 $\to$ positive; fraction .1010000 $= \frac{1}{2} + \frac{1}{8} = 0.625$
- exponent 00000010 $= +2$
- value $= 0.625 \times 2^2 = 0.625 \times 4 = \mathbf{2.5}$

Method — Denary $\rightarrow$ binary (normalised)

Convert `6.0` (8-bit mantissa, 8-bit exponent):

- 6 in binary $= 110.0$
- move the point left until it is just after the first `1`: 0.110×2^3
- mantissa = 01100000, exponent $= 3 =$ 00000011

Method — Normalising

A positive mantissa is **normalised** when it begins `0.1`… (no wasted leading zeros — maximum precision). Shift left and decrease the exponent:

- 00011000 exp 00000101 $= 0.0011 \times 2^5$
- shift mantissa left 2 $\rightarrow$ 01100000, exponent $5 - 2 = 3 =$ 00000011 (same value).

Method — Negative numbers

A **negative** mantissa is in **two's complement** and is normalised when it begins `1.0`… (the first two bits differ). To represent -0.75: take $+0.75 =$ `0.1100000`, then two's-complement it (invert, add 1) $\rightarrow$ `1.0100000`. The sign/integer bit has place value -1, so `1.0100000` $= -1 + \frac{1}{4} = -0.75$.

Method — Rounding error

Some denary reals (like 0.1) are **repeating** binary fractions, so they are stored only **approximately**. Errors build up, so test `ABS(x - 0.3) < 1e-9`, not `x = 0.3`.

Practice 2.1 ▪ 2 marks

State the **two** parts of a floating-point number and what each represents.

Solution 2.1

Method: The format

Worked steps

The **mantissa** holds the **significant digits**; the **exponent** is the **power of 2** to multiply by **B1** **B1**.

Practice 2.2 ▪ 2 marks

A mantissa 01000000 has exponent 00000011. State its **denary** value.

Solution 2.2

Worked steps

$\texttt{0.1000000} = 0.5$; exponent $= 3$; $0.5 \times 2^3 = \mathbf{4}$ **M1** **A1** *read mantissa as* 0.5*,* $= 4$.

Practice 2.3 ▪ 1 mark

State what it means for a number to be **normalised**.

Solution 2.3

Worked steps

The first significant bit is **immediately after the binary point** (`0.1`... for a positive number) — no wasted leading zeros, so precision is maximised **B1**.

Practice 2.4 ▪ 1 mark

State **one** reason some denary real numbers cannot be stored exactly.

Solution 2.4

Method: Rounding error

Worked steps

Some denary fractions (e.g. 0.1) are **repeating/infinite** in binary, so they must be **truncated** to fit the mantissa B1.

Exam-style 3.1 ▪ 3 marks

Convert mantissa 01010000, exponent 00000010 to **denary**. Show your working.

Solution 3.1

Method: Denary $\rightarrow$ binary (normalised)

Worked steps

Sign bit 0 $\rightarrow$ positive M1; $.1010000 = \frac{1}{2} + \frac{1}{8} = 0.625$, exponent $= 2$ M1; $0.625 \times 2^2 = \mathbf{2.5}$ A1.

Exam-style 3.2 ▪ 4 marks

Convert `6.0` to its **normalised** floating-point representation. Show your working.

Solution 3.2

Method: Denary $\rightarrow$ binary (normalised)

Worked steps

6 $= 110.0_2$ **M1**; normalise to 0.110×2^3 **M1**; mantissa 01100000 **A1**; exponent 00000011 **A1**.

Exam-style 3.3 ▪ 3 marks

Normalise the mantissa 00011000 with exponent 00000101. Give the new mantissa and exponent.

Solution 3.3

Worked steps

00011000 $= 0.0011 \times 2^5$; shift mantissa **left 2**, exponent $5 - 2 = 3$ **M1**; new mantissa 01100000 **A1**; new exponent 00000011 **A1**.

Exam-style 3.4 ▪ 2 marks

Explain why `0.1 + 0.2` might **not** equal exactly `0.3` when stored in floating-point.

Solution 3.4

Method: The format

Worked steps

`0.1` and `0.2` cannot be stored **exactly** in binary (they are repeating fractions) **B1**; the small stored errors **add up**, so the result is very close to but **not exactly** `0.3` **B1**.

Exam-style 3.5 ▪ 3 marks

Convert the floating-point number with mantissa 10110000 and exponent 00000010 to **denary**. Show your working.

Solution 3.5

Worked steps

Sign bit 1 $\rightarrow$ negative M1; $1.0110000 = -1 + \frac{1}{4} + \frac{1}{8} = -0.625$, exponent $= 2$ M1; $-0.625 \times 2^2 = -\mathbf{2.5}$ A1.