Chapter 5 · Section 5.3 — Exercises (Mazidi)

Chapter 5 · Section 5.3 — Exercises (Mazidi)

> Problems are paraphrased to respect copyright. I show the **bit layout**, the **bias math**, and hand-conversion sketches, then give the exact encodings (hex and fields).

---

6) Disadvantage of using a general-purpose processor for math operations

Answer: Without dedicated math hardware (e.g., **hardware multiply/divide** or an **FPU**), a GPP must emulate many operations in **software**, making them **much slower** (many more cycles) and often **larger in code size** than on a DSP or an MCU with an FPU.

---

7) Bit assignment of the IEEE-754 single-precision (32-bit) format

- **Sign**: 1 bit (bit31).
- **Exponent**: 8 bits (bits30–23), **bias = 127**.
- **Fraction (mantissa)**: 23 bits (bits22–0).
- Normalized value: `V = (-1)^S × (1.F) × 2^(E - 127)`.

---

8) Convert each real number to single precision (by hand)

I outline the steps and then show the final fields. (Fraction is rounded to 23 bits.)

value	sign S	unbiased exp	biased E (bin)	fraction bits (23)	32-bit hex
15.575	0	3	10000010	11110010011001100110011	**0x41793333**
89.125	0	6	10000101	01100100100000000000000	**0x42B24000**
−1022.543	1	9	10001000	11111111010001011000001	**0xC47FA2C1**
−0.00075	1	−11	01110100	10001001001101110100110	**0xBA449BA6**

Sketch of the first two:

- `15.575 = 1111.10010011…₂ = 1.11110010011… × 2^3`, so `E=3+127=130 (10000010)` and `F=11110010011001100110011`.
- `89.125 = 1011001.001₂ = 1.011001001 × 2^6`, so `E=6+127=133 (10000101)` and `F=011001001000…`

---

9) Bit assignment of the IEEE-754 double-precision (64-bit) format

- **Sign**: 1 bit (bit63).
- **Exponent**: 11 bits (bits62–52), **bias = 1023**.
- **Fraction (mantissa)**: 52 bits (bits51–0).
- Normalized value: `V = (-1)^S × (1.F) × 2^(E - 1023)`.

---

10) Single-precision: the biased exponent is calculated by adding 127 to the exponent portion of the normalized scientific binary number.

11) Double-precision: the biased exponent is calculated by adding 1023 to the exponent portion of the normalized scientific binary number.

---

12) Convert to double precision

value	S	unbiased exp	E (bin)	52-bit fraction F	64-bit hex
12.9375	0	3	10000000010	1001111000000000000000000000000000000000000000000000	**0x4029E00000000000**
98.8125	0	6	10000000101	1000101101000000000000000000000000000000000000000000	**0x4058B40000000000**

Sketch: $12.9375 = 1100.1111_2 = 1.1001111 \times 2^3$ and $98.8125 = 1100010.1101_2 = 1.1000101101 \times 2^6$ → add the bias `1023` and fill the fraction.

Notes for learners

- The **hidden 1** is present for all **normalized** numbers (not for subnormals).
- Rounding mode by default is **round to nearest, ties to even**; that's why some decimal fractions (e.g., 0.00075) get long fraction fields and rounding.
- For quick checks: interpret the hex in a programmer's calculator; confirm `S`, `E`, and `F` by splitting the bit fields.