

Feedback Comment Bank

Consistent, specific feedback in seconds. Write the **code** (e.g. B3) on the paper, or paste the full comment into your LMS. Growth-oriented — every comment ends with the fix.

How to use: for handwritten work, jot the short code next to the error; students look codes up on a posted key. For digital feedback, copy the full comment from the plain-text version (`unit1-feedback-bank.txt`) — it's the same bank formatted to paste.

Binary & number systems (B)

Code	Comment
B1	You gave the largest value as 2^N . That's the <i>number of values</i> ; the largest value is $2^N - 1$ because counting starts at 0. Re-check your max-value answers.
B2	Your binary is missing leading zeros — the problem asked for 8-bit numbers, so pad to 8 digits (e.g., 13 = 00001101, not 1101).
B3	Right method, arithmetic slip: re-add your place values (128 64 32 16 8 4 2 1) and you'll catch it. Show which places you turned on next time.
B4	You mixed up bit and byte. 1 byte = 8 bits. Re-read the question with that in mind.
B5	Bits read left-to-right with the largest place value on the left. You reversed the order — flip it and re-convert.

Encoding — text & images (E)

Code	Comment
E1	For color images it's 8 bits per <i>channel</i> (R, G, B) = 24 bits per <i>pixel</i> , not 8 bits per pixel. Recount your total.
E2	Channel values run 0–255 (256 levels), not 1–256. The lowest value is 0.
E3	Your bit string doesn't match your shaded grid. Read the grid row by row, left to right, 1 = black / 0 = white, and rewrite the bits.
E4	ASCII slip: double-check against A = 65. Each later letter adds 1 (B = 66, C = 67...).
E5	Good encoding, but you didn't state the total bit count the question asked for. Add it — that's part of understanding the cost of representation.

Compression, overflow & round-off (C)

Code	Comment
C1	Lossless loses <i>nothing</i> — the original is perfectly recoverable. Your answer implied it drops a little data; that describes lossy.

Code	Comment
C2	Lossy isn't "just worse" — it's a deliberate trade: much smaller files for a little quality loss (JPEG, MP3). Frame it as a trade-off.
C3	Add a concrete example. An answer with no example (a file type or real situation) is capped at half credit.
C4	You've mixed up overflow and round-off. Overflow = value too big for the bits (wraps around). Round-off = value can't be stored exactly. Separate the two.
C5	"The computer breaks" isn't the mechanism. Describe what actually happens: wrap-around (overflow) or a slightly-off stored value (round-off).

Digital Information Dilemma brief (D)

Code	Comment
D1	Your source entries are too vague to credit. Name the title and outlet, and write one specific fact you learned plus why you trust it.
D2	Your local example could apply to anyone. Anchor it to something real from your life, our school, or our town — that's the part that shows real understanding (and can't be faked).
D3	You described the benefit but not the harm (or vice-versa). DAT-2.E requires <i>both</i> — name who benefits and who is harmed.
D4	Strong analysis, but no clear position or policy. State what you think should happen and one concrete rule you'd recommend.
D5	You identified the innovation but not the <i>specific data</i> it collects. Name the actual data and one privacy/security risk it creates.

Positive / growth comments (P) — use these too

Code	Comment
P1	Excellent — you showed your work clearly, which makes your reasoning easy to follow and easy to fix if a step slips. Keep doing this.
P2	Your local example was specific and real. That's exactly the kind of thinking the AP exam rewards on the written responses.
P3	You nailed the trade-off framing (cost vs. benefit). Carry that habit into every "compare the choices" question.
P4	Big improvement in precision — you used the exact terms (bit, byte, lossless) correctly. That vocabulary matters on the MCQ.