

Almost every mark lost in 0478 goes to imprecision rather than ignorance: a term used loosely, a conversion done in the wrong direction, a trace table filled in from the answer instead of from the code. Read this before a mock, and again the night before. 失分几乎都源于表述不精确，而不是不会 – 考前请再读一遍。

Definitions and terminology

Using ‘validation’ and ‘verification’ interchangeably. Validation is an automatic check that data is SENSIBLE; verification checks it was entered ACCURATELY. A range check is validation; double entry is verification.

Saying an IP address identifies the device permanently. That is the MAC address; an IP address is assigned by the network and can change.

Describing RAM as ‘where files are stored’. RAM holds what is currently in use and is volatile; files live in secondary storage.

Calling a compiler and an interpreter ‘the same but faster’. A compiler translates the whole program once and produces an executable; an interpreter translates and runs line by line and stops at the first error.

Answering ‘what is a protocol’ with an example. TCP/IP is a protocol; the definition is a set of rules governing transmission.

Data representation

Converting denary to binary by repeated division and then writing the remainders in the order they were produced. They must be read BOTTOM to TOP.

Forgetting that one hexadecimal digit is exactly 4 bits, and converting via denary instead – slower and where the arithmetic errors creep in.

Giving a file-size answer in bits when the question asked for bytes, or vice versa. Divide by 8, and state the unit.

Confusing sampling RATE with sampling RESOLUTION: rate is samples per second, resolution is bits per sample. Both raise file size, for different reasons.

Saying lossy compression ‘makes the file smaller without losing anything’. That is lossless; lossy discards data permanently.

Algorithms, trace tables and pseudocode

Filling a trace table from what the algorithm is SUPPOSED to do. Execute it literally, line by line – the question is usually testing a deliberate off-by-one.

Leaving a trace-table cell blank when a variable does not change. Every row needs its values, or the examiner cannot follow the trace.

Writing pseudocode with no indentation and no closing keyword. ENDIF, NEXT and ENDWHILE are part of the syntax being marked.

Confusing = with $\leftarrow$ in Cambridge pseudocode: $\leftarrow$ assigns, = compares.

Off-by-one in an array loop. Cambridge pseudocode arrays are 1-INDEXED, so an array declared [1:10] has no element 0.

Describing a bubble sort as finished after one pass. It repeats until a pass makes no swaps, and saying so is the mark.

Programming

Using a variable before it has been given a value, and losing the initialisation mark – totals start at 0, counters at 0 or 1, and flags at FALSE.

Reading input inside a loop when the question wanted it once, or outside when it wanted it each iteration. Re-read the requirement before writing.

Comparing a string to a number without converting, or comparing with = instead of the language's equality operator.

Writing a validation routine that rejects bad input but does not ask again. A validation loop must re-prompt until the input is acceptable.

Answering 'give a test' with the value only. A test needs the data, the TYPE of test (normal, boundary, erroneous) and the expected result.

Databases and logic

Choosing a primary key that is not unique – a surname or a date will not do; it must identify one record only.

Writing SQL without the correct clause order: SELECT, FROM, WHERE, ORDER BY.

Omitting the quotation marks around a text value in a WHERE clause.

Filling a truth table with only the combinations that seem interesting. Every combination of inputs must appear – 2^n rows for n inputs.

Reading a logic circuit right to left. Work from the inputs forwards, labelling the output of each gate as you go.