A mark scheme prints the answer, not the route. These are the tasks 0478 asks every session, written the way you should write them in the exam: the working visible, the method marks earned before the answer, and a note saying where the marks actually sit. 官方答案只给结果，这里把每一步都写出来。

Number systems and binary arithmetic

Convert the denary number 214 to 8-bit binary, showing your method. [3]

Divide repeatedly by 2 and read the remainders from the bottom up. $214 \div 2 = 107$ r0; $107 \div 2 = 53$ r1; $53 \div 2 = 26$ r1; $26 \div 2 = 13$ r0; $13 \div 2 = 6$ r1; $6 \div 2 = 3$ r0; $3 \div 2 = 1$ r1; $1 \div 2 = 0$ r1. Reading upwards gives '11010110'. Check with place values: $128 + 64 + 16 + 4 + 2 = 214$.

Either method earns the marks – repeated division, or subtracting place values from the left. Show whichever you use; a bare answer scores one mark of three.

Always check by adding the place values back. It costs five seconds and catches a dropped bit, which is the commonest error in this question.

Two 8-bit binary numbers, '10110110' and '01101011', are added. Give the 8-bit result and state whether overflow occurs. [4]

Adding column by column from the right, carrying where a column makes two: $0 + 1 = 1$; $1 + 1 = 10$, write 0 carry 1; $1 + 0 + 1 = 10$, write 0 carry 1; $0 + 1 + 1 = 10$, write 0 carry 1; $1 + 0 + 1 = 10$, write 0 carry 1; $1 + 1 + 1 = 11$, write 1 carry 1; $0 + 1 + 1 = 10$, write 0 carry 1; $1 + 0 + 1 = 10$, write 0 carry 1. The eight stored bits are '00100001' and a 1 is carried out of the eighth column. Overflow HAS occurred: the true answer is 289, which needs nine bits, and an 8-bit register cannot hold it, so the stored value is wrong.

Write the carries above the columns. The marks are for the addition itself and for the overflow judgement, and an examiner cannot award the first without seeing the working.

Overflow is not 'the answer is big'. It is specifically that the result needs more bits than the register has – say that, and say the stored answer is therefore wrong.

A register holds the two's complement value '11100110'. Give its denary value, and explain how you know it is negative. [3]

In two's complement the leftmost bit is worth -128, so the value is $-128 + 64 + 32 + 4 + 2 = -26$. It is negative because the most significant bit is 1, and that bit carries a negative place value.

The quickest reliable route is to treat the MSB as -128 and add the rest normally. Inverting and adding one also works but takes longer and invites a slip.

The explanation mark is for naming the most significant bit, not for saying 'it starts with a 1'.

Data representation and file size

A sound file is 45 seconds long, mono, sampled at 44 100 Hz with a sample resolution of 16 bits. Calculate the file size in mebibytes. [4]

File size in bits = sampling rate × resolution × seconds = $44\,100 \times 16 \times 45 = 31\,752\,000$ bits. Divide by 8 for bytes: $3\,969\,000$ bytes. Divide by 1024 twice: $3\,969\,000 \div 1024 \div 1024 = 3.79$ MiB (3 s.f.).

One mark is for the product, one for dividing by 8, one for the two divisions by 1024, and one for the number. Each is awarded separately, so an arithmetic slip early still earns the later method marks if the working is on the page.

Read the unit the question wants. Answers left in bits or bytes when mebibytes were asked for lose the final mark, and this is the single commonest loss on the question.

An image is 1024 pixels wide and 768 pixels high with a colour depth of 24 bits. Calculate its file size in mebibytes, then state the effect on the file size of halving the colour depth. [4]

Bits $= 1024 \times 768 \times 24 = 18\,874\,368$. Bytes $= 18\,874\,368 \div 8 = 2\,359\,296$. MiB $= 2\,359\,296 \div 1024 \div 1024 =$ 2.25 MiB. Halving the colour depth to 12 bits halves the file size, to 1.125 MiB, because file size is directly proportional to colour depth.

The last mark needs the REASON, not just the halved number: file size is proportional to colour depth.

Colour depth of n bits gives 2^n colours – a different question, often asked in the same part. Do not confuse the two.

Transmission and error checking

A byte of seven data bits, '1011010', is to be sent with even parity. Give the byte that is transmitted, and explain why the receiver would reject '10110101'. [3]

The data bits contain four 1s, which is already an even number, so the parity bit is 0 and the transmitted byte is '10110100'. If '10110101' arrived, the receiver would count five 1s. Five is odd, and even parity was agreed, so the byte fails the check and is rejected as corrupted.

State the parity scheme you are using in words. The mark is for the count and the conclusion, not for the bit alone.

A parity check detects an error; it does NOT say which bit is wrong and cannot correct it. Questions asking for a limitation want exactly that, plus the fact that two errors can cancel out.

Explain why data sent over a long distance uses serial rather than parallel transmission. [3]

Serial sends one bit at a time down a single wire. Over a long distance parallel wires suffer skew, so bits sent together arrive at different times and the data is reassembled wrongly. Parallel wires also interfere with each other over distance (crosstalk), and a serial link needs far less cable, so it is cheaper and more reliable.

Naming skew is worth a mark on its own. 'It is more reliable' with no reason is not.

Parallel is not always worse – it is faster over SHORT distances, which is why it is used inside a computer. Saying so shows you understand the trade-off rather than a rule.

Logic expressions, truth tables and circuits

A system sounds an alarm when the door is open AND the system is armed, OR when the panic button is pressed. Using D, A and P, write the logic expression and complete a truth table for it. [6]

The expression is $X = (D \cdot A) + P$. The truth table needs $2^3 = 8$ rows, with a working column for $D \cdot A$: for $DAP =$ 000, 001, 010, 011, 100, 101, 110, 111 the column $D \cdot A$ is 0, 0, 0, 0, 0, 0, 1, 1, and the output X is 0, 1, 0, 1, 0, 1, 1, 1.

Write the working column. It earns marks by itself, and it is how the examiner sees a correct method behind a slipped final column.

'AND' in the sentence becomes the dot, 'OR' becomes the plus. Translate the words one at a time before drawing anything.

Draw the logic circuit for $X = \overline{A \cdot B} + C$ and identify the single gate that produces $\overline{A \cdot B}$. [4]

A and B feed a NAND gate – that single gate is $\overline{A \cdot B}$, since NAND is AND followed by NOT. Its output and C then feed an OR gate, whose output is X.

Work outwards from the innermost bracket: build $A \cdot B$ first, then its inversion, then the OR.

Using an AND followed by a NOT is also correct and earns full marks, but the question asks which SINGLE gate does it, so NAND must be named.

Algorithms, pseudocode and trace tables

Write pseudocode that inputs temperatures until the rogue value -99 is entered, then outputs the total, the average and the highest value. [**7**]

```
Total ← 0
Count ← 0
Highest ← -100
INPUT Temp
WHILE Temp <> -99 DO
    Total ← Total + Temp
    Count ← Count + 1
    IF Temp > Highest THEN
        Highest ← Temp
    ENDIF
    INPUT Temp
ENDWHILE
IF Count > 0 THEN
    OUTPUT "Total ", Total
    OUTPUT "Average ", Total / Count
    OUTPUT "Highest ", Highest
ELSE
    OUTPUT "No data entered"
ENDIF
```

The first INPUT must come BEFORE the loop, or the condition has nothing to test. This is the single commonest structural error in the question.

The second INPUT must be the LAST statement inside the loop. Put it first and the rogue value is added to the total.

Guarding the division against a zero count is a mark, and it is the one candidates skip.

Complete a trace table for the algorithm above with the input data 18, 21, 15, -99. [**4**]

Start: Total 0, Count 0, Highest -100. After 18: Total 18, Count 1, Highest 18. After 21: Total 39, Count 2, Highest 21. After 15: Total 54, Count 3, Highest 21. Temp then becomes -99, the loop ends, and the outputs are 54, 18 and 21.

Write every new value on a NEW row and never rub one out – the examiner is marking the sequence of values, not only the last one.

Highest stays at 21 for the third reading because 15 is not greater than 21. Showing the unchanged value is what proves you tested the condition.

State one validation check and one verification check for a password field, and explain the purpose of each. [4]

Validation: a length check that the password is at least eight characters, because a short password is easily guessed and the computer can test this automatically as the data is entered. Verification: double entry, where the user types the password twice and the two are compared, because a typing mistake would otherwise lock the user out of their own account.

Validation asks 'is this data sensible?'; verification asks 'was it copied correctly?'. Naming the wrong one costs both marks.

A presence check is validation, not verification, however carefully it is described.

Hardware, the CPU and security

Describe the fetch stage of the fetch–execute cycle, naming the registers used. [4]

The address held in the program counter (PC) is copied to the memory address register (MAR). The PC is then incremented so it points at the next instruction. The instruction stored at that address is fetched along the data bus into the memory data register (MDR). It is then copied from the MDR into the current instruction register (CIR), ready to be decoded.

The order is the answer. Incrementing the PC after the copy to the MAR, not before, is a mark in itself.

Name the registers by their abbreviations AND in full at least once. Questions frequently ask for the full name.

A company is worried about phishing and pharming. Explain the difference, and give one prevention for each. [5]

Phishing sends the user a message – usually an email – that pretends to come from a trusted organisation, and the attack only works if the user clicks the link or replies with their details. Pharming needs no message: malicious code on the user's computer, or on a DNS server, redirects them to a fake website even when they type the correct address. Phishing is prevented by not clicking links in unexpected email and checking the sender's address, or by an anti-spam filter. Pharming is prevented by running anti-malware to remove the redirecting code, and by checking the site's security certificate before entering details.

The distinguishing mark is that pharming requires no action by the user. Say that explicitly – it is the discriminator the mark scheme rewards.

Preventions must match the threat. 'Use a strong password' prevents neither, and scores nothing on this question.