A-LEVEL
COMPUTER SCIENCE

7517/2 Paper 2
Report on the Examination

7517/2
June 2023

Version: 1.0

Question 1

Question 1 was about the representation of images and sound.

For question part 1.1 students had to calculate the size of a sound file in mebibytes. It was more common for a student to achieve one mark than two, as whilst most students knew the correct basic calculation to perform, they often made mistakes when applying it. The most common mistakes were in the conversion of the answer to mebibytes, and included dividing by 1000 instead of 1024 and forgetting that it was necessary to divide by 8 to convert bits to bytes.

The vast majority of students correctly recognised that Nyquist's theorem was relevant to question part 1.2 and responded that a minimum sample rate of 30 000Hz would be required when sampling. Question part 1.3 was less well tackled, with a similar number of students responding that the component that carried out the process of converting binary values into a waveform was an analogue to digital converter as giving the correct response of digital to analogue converter.

Question 2

Question 2 was about packet switching.

Question part 2.1 required students to identify two fields that would be found in a typical packet. The majority of students were able to name one field, but only about a quarter could correctly name two fields.

Question part 2.2 was also tackled well, although students sometimes failed to achieve marks by writing too vaguely. Good responses recognised that a checksum would be used to ensure that the received data had not been changed during transmission and that the checksum would be calculated by applying a mathematical operation to the payload of the packet. Weaker responses, which were not credited, made more general points such as that the checksum would identify if the received data was correct or that a checksum was made by applying an algorithm to data. A checksum could not be used to test if received data is correct as this would depend upon whether the transmitted data was correct or not. Some students gave very specific examples of how a checksum might be calculated. These were credited if they were feasible for a packet of data.

For question part 2.3, students were asked to describe the role of a router in packet switching. Many responses were quite superficial and stated little more than that a router would send a packet to its destination. Better responses recognised that routing is generally done on a hop-by-hop basis and that a router would take a decision about the path to send a packet on for the next hop.

Question 3

Question 3 was about encryption.

Question part 3.1 required students to encrypt a word using the Caesar cipher and a key of 4. This was very well tackled, with the vast majority of students achieving the mark. Most mistakes involved shifting a single letter incorrectly, most frequently the last letter which was the only one in the word that wrapped around to the start of the alphabet.

For question part 3.2 students had to identify a weakness that applied to both the Caesar cipher and a different substitution cipher described on the question paper. Good responses usually focused on either the fact that each letter was always translated to the same letter or that frequency analysis could be performed on ciphertext produced by either of the two ciphers. Common errors were to state that there were a small number of keys or that the ciphers were susceptible to brute force cracking. Whilst the Caesar cipher does have a small number of keys, this was not true for the substitution cipher described on the question paper.

Question part 3.3 asked students to identify a reason why the substitution cipher on the question paper was harder to crack than the Caesar cipher. This was well answered, with the most common markworthy responses relating to the increased number of keys or the fact that cracking how one letter had been encrypted would reveal little about how other letters had been encrypted. Students often used the word “random” in their responses, and, where possible, responses that used this word were credited, but students should be careful about using this word which has a specific meaning in computer science in response. Sometimes, such responses were too vague to be credited, such as that “the encryption method is more random”.

The conditions required for the Vernam cipher to be perfectly secure were well known for question part 3.4 with around half of the students achieving full marks. The most common mistake was to refer to “it” rather than “the key” in a response. For example, whilst “the key must only be used once” was markworthy, “it must only be used once” was not, as the question was about the cipher not the key.

Question 4

Question 4 was about hardware, including the fetch-execute cycle, computer performance and interrupts.

Question part 4.1 was a twelve-mark question about how the fetch-execute cycle is used and how the hardware of a computer could be improved to carry out the instructions faster. There were many good responses to this question. The most common issues in responses related to missing out parts of the fetch-execute cycle or focusing very narrowly on a small number of measures to improve performance. The fetch stage of the fetch-execute cycle was often described in more detail than the decode and execute stages, with the execute stage being the least well tackled. Many students chose to explain how the measures that they were suggesting would improve the execution speed of instructions, but in some cases these were too vague, for example stating that using a processor with more cores would result in more instructions being executed per second instead of stating that this would allow multiple instructions to be executed simultaneously.

Students made a better job of explaining the purpose of an interrupt in question part 4.2 than stating what an interrupt was. Good responses stated that an interrupt was a signal sent by a device or program to request the urgent attention of the processor. Common mistakes when explaining the purpose of an interrupt were to write that the interrupt stopped the fetch-execute cycle and to state that interrupts were used to ensure that higher priority instructions were execute before lower priority ones.

Question 5

Question 5 was about relational databases and SQL.

For question part 5.1 students were required to complete the definition of a relational database to store details about bookings for a cinema. Most students achieved some marks, but relatively few achieved all five marks. The most commonly awarded marks were for identifying the correct attributes and entity identifier for the Customer relation. Common mistakes were to include redundant attributes such as the ScreenNumber in the Booking relation, when this could be identified from the ShowingID, and to try to represent a booking using just one relation which bookings were made for multiple specific seats, so needed to use two relations to produce a normalised design for all the details about a booking.

Question part 5.2 asked students to identify errors in an SQL query to delete all film showings on a specific date. This was fairly well answered, with the most common correct responses identifying that the Film table should not have been included and that quotation marks were missing around the date. Some students stated that the two tables in the query needed to be joined, which is possible in some database implementations in a DELETE query, but this was not relevant in this instance as the Film table was not required at all.

Question part 5.3 dealt further with the topic of deleting data from the database and asked students what issues might arise if all the showings on a particular date were deleted. Good responses recognised that bookings might already exist for this date and that attempting to delete the showings without deleting the bookings might result in referential integrity issues or the violation of foreign key rules. Many responses related to this but did not use standard database terminology. These were credited where they mapped to the relevant concepts, but students should attempt to use correct terminology where possible.

Question parts 5.4 and 5.5 were about CRUD and REST, which are relatively small parts of the specification and were not well answered. Good responses to question part 5.4 recognised that the URL would cause a database query to be executed on the server to retrieve a particular dataset. For question part 5.5 the correct response was B (GET→SELECT, POST→INSERT, DELETE→DELETE, PUT→UPDATE). This was the most commonly selected response, followed by response D.

In contrast, the advantages of JSON over XML were well known, with most students achieving at least one mark for question part 5.6

Question 6

Question 6 was about floating point number representation.

Question parts 6.1 and 6.2 were conversion questions and both were well answered. In question part 6.1, the most common error was to move the binary point in the wrong direction, resulting in an answer of 6.5 instead of the correct answer of 13/128. For question part 6.2, the most common error was to represent -23.25 by writing the representation of -23 followed by the representation of +0.25, which actually gives the representation of -22.75 instead of -23.25. Students who first represented +23.25 then converted this to -23.25 avoided this mistake.

Most students were unable to name the types of error described in question part 6.3. A common mistake was to confuse overflow with stack overflow.

For question part 6.4, students were required to explain how the floating point representation could be modified to represent bits more precisely, without increasing the number of bits used. Good responses recognised that bits could be taken from the exponent and used in the mantissa or that the bit before or after the binary point could be implied from the bit on the other side of it, so did not need to be stored and that this could provide an extra bit to store one more digit of precision.

Question 7

Question 7 was about assembly language programming and masking.

Question parts 7.1 and 7.2 were both relatively straightforward masking questions, and as such were answered very well.

For question part 7.3 students had to write an assembly language program to convert an 8-bit binary number into the two ASCII codes of its hexadecimal representation. This was the most complex assembly language programming task that has been asked on this specification and it was pleasing to see many good responses. Whilst fully correct responses were rare, the majority of students correctly performed the masking and/or shifting required to isolate the binary patterns of the two hexadecimal digits and made some attempt to convert these to ASCII codes. The most common errors were:

- writing operands in binary
- missing out the # required to use immediate addressing mode
- ANDing with the value 240 but then not shifting the bits four places to the right
- forgetting that code that is the target of a branch will carry on executing lines of code after another label is defined and therefore might execute some additional code unintentionally. For example, the code below correctly adds 48 to the value of numeric digits to convert to ASCII, but incorrectly adds 103 to the value of alphabetic digits instead of 55:

```
 CMP R0, #10
 BLT isnumericdigit
 ADD R0, R0, #55
isnumericdigit:
 ADD R0, R0, #48
```

A small number of students appeared to have had no prior experience of using the AQA assembly language instruction set. It is important that students have used this prior to the day of the examination.

Question 8

Question 8 was about Big Data.

For question part 8.1 students were asked to describe the third characteristic of Big Data. Good responses described that this was that the data arrived and had to be processed very quickly. Weaker responses often failed to recognise the very high speed appropriately, referencing timeframes that could easily be dealt with by a more traditional relational database system.

Question part 8.2 required students to complete a graph schema for a supermarket. There were many good responses to this question part, with the most common errors being using the wrong shapes or types of line or linking the cost of £1.50 and number of 20 biscuits in a packet to chocolate biscuits instead of iced biscuits. The majority of students achieved two or three of the available three marks.

Features of functional programming languages that made them appropriate for distributing code to run across more than one server were not well known for question part 8.3. Responses were often very general or did not really relate to functional programming. Good responses recognised specific features relating to functional programs, such as data structures being immutable and functions having no side-effects. Some responses were single words such as “immutable” which were not clear enough to be awarded a mark.

Question 9

Question 9 was about logic circuits and Boolean algebra.

For question parts 9.1 and 9.2 students were asked to draw a logic circuit based on a truth table to implement a half-adder circuit and then to identify its purpose. Both of these question parts were answered very well. The most common error for question part 9.2 was to give a very literal description of how the outputs were arrived at rather than to analyse the overall purpose of the circuit.

For question part 9.3 students had to simplify a Boolean expression that required the application of De Morgan’s laws. There were many good responses. The most common mistakes that students made were similar to those seen in previous questions of this type, namely:

- Simplifying $\overline{\mathrm{B}} + \mathrm{B}$ to 1 near to the start of the simplification, which was not possible in the expression since AND has a higher precedence than OR.

- Cancelling NOTs that extended over different parts of the expression, simplifying $(\overline{\overline{\mathrm{A}} + (\overline{\mathrm{B}} \cdot \mathrm{C})})$ to $(\mathrm{A} + (\mathrm{B} \cdot \overline{\mathrm{C}}))$ instead of applying De Morgan’s laws.

Question 10

Question 10 was about data communications.

For question part 10.1 students were required to identify how many of the bit patterns could be valid representations of a single character sent in an asynchronous data transmission system using even parity. This was well tackled with most students correctly identifying that two patterns were valid.

For question part 10.2 students were asked to describe synchronous transmission. This was not tackled well. Good responses recognised that the clocks of the transmitter and receiver would be constantly synchronised, or that a clock signal would be embedded in the data being transmitted. Students who did not achieve marks usually wrote about clocks but were too imprecise in their responses.

The limitations of parity bits were well understood for question part 10.3, with around half of students achieving the mark, the most common responses being that errors that involved an even number of bits being flipped could not be detected or that errors could be detected but not corrected. It was not enough to state that multi-bit errors could not be detected, as some of these could be. Students should note that an even number of errors occurring is not the same as an even number of bits being changed. For example, two errors might be the data being transmitted at an incorrect clock speed and to the wrong destination.

The vast majority of students correctly identified the relationship between bandwidth and bit rate in question part 10.4

Question 11

Question 11 was about networking and Internet security.

Question part 11.1 asked students to describe the role of the transport layer. Most students achieved one mark, but few achieved all three. The most frequently awarded mark was for the layer splitting data into packets. The fact that “packet” is not the correct term at this layer (the correct term is “segments”) was not penalised when marks were awarded. Good responses often went on to cover the use of port numbers and the establishment of an end-to-end connection. The most common mistake was to describe the role of the network layer instead of the transport layer.

Question part 11.2 asked students to describe the additional function of a gateway beyond that of a router. Relatively few students correctly identified that a gateway could repackage data that was being transmitted between networks using different protocols.

Question part 11.3 was about application layer protocols used when emails are sent. This was well answered, with the most commonly referenced protocols being SMTP, IMAP and POP3. Errors were infrequent but included naming layers instead of protocols and naming transport layer protocols instead of application layer protocols.

For question part 11.4 students had to explain what a well-known port is and why email servers used them. Good responses explained that a well-known port was a port number that had a particular purpose and was assigned by IANA. Students achieved this mark more frequently than the second mark, which was for why email servers used them. To achieve this mark, students had to recognise that the sender initiated the communication and so the port number must be fixed.

Question part 11.5 was a six-mark question about asymmetric encryption and digital signatures. Most students had some understanding of the topics and were able to describe them at least partially. The most common mistakes were:

- Referring to public and private keys, but not recognising that both the sender and receiver had different public and private keys.
- Mixing up which key was used for each part of the process.
- Writing vaguely about how the message digest would be used after reception to confirm that the message was sent by A. For example, students often wrote that the “recomputed message digest would be compared to the message”.

Question 12

Question 12 was about functional programming.

For question part 12.1 students had to show the arguments and values returned when a recursive function was called. The most commonly achieved marks were for correctly completing the argument column and for identifying the argument and returned value for the base case in the recursion. Students frequently went wrong when working out how the values returned were combined to produce the final value returned. Most students remembered to include brackets around lists, which were frequently omitted in previous years.

Student needed to identify the co-domain of the function for question part 12.2. The majority of students selected the correct response, which was A (the set of integers).

Mark Ranges and Award of Grades

Grade boundaries and cumulative percentage grades are available on the Results Statistics page of the AQA Website.