AQA
Questions matter

A-level

Computer Science

7517/2 Paper 2

Report on the Examination

7517
June 2024

Version: 1.0

Question 1

This question was about measures that could be taken to reduce the threat posed by viruses. Most responses focussed on monitoring for and protection against viruses rather than how the quality of code might reduce the threat. Common responses included the use of firewalls, encryption, access rights, digital certificates and proxy servers. Students who did not achieve marks often failed to do so because whilst they may have named an appropriate measure, they did not go on to describe it in sufficient detail, which was what the question required. Some students responded with measures already given to them in the question, which were not markworthy or made multiple points about firewalls, which together could only be worth one mark.

Question 2

This question was about images, analogue and digital data and run-length encoding.

Question part 2.1 asked students to calculate the size of an image in megabytes. The most common mistake was to work out the colour depth incorrectly, for example by obtaining the square root of the number of colours in the image. It is important that students show their working clearly to ensure that they achieve working marks if their final response is incorrect. Some students did not show the simpler working steps, such as dividing by 1000.

For question part 2.2 students had to work out how many of the images from question part 2.1 could fit on a 256-gigabyte memory card. This part was well tackled, with many students who answered question part 2.1 incorrectly still achieving the mark as they correctly worked out how many images of their calculated size could fit on the card.

Question part 2.3 was about analogue and digital data. The simplest way to achieve a mark for this question part was to state that the photosensor voltages were continuous and the pixel data was discrete. It was equally acceptable to explain these concepts, for example by stating that the pixel data would be coded into a fixed number of bits, instead of stating that it was discrete. It was not enough to state that the pixel data was binary. A common incorrect response was that the pixel data was digital because it could be understood by a computer.

Question parts 2.4 and 2.5 were about run-length encoding (RLE). Most students correctly performed the calculation in part 2.4 and identified that compression had not been successful because the number and length of the runs in the data meant that more memory was used after RLE had been carried out.

Question 3

This question was about computer hardware.

Students were required to name the labelled bus on a diagram for question part 3.1 and just over half of students correctly identified the address bus.

Question part 3.2 was about synchronous transmission. Most students recognised that this related to the clock used, but some confused it with either parallel or bi-directional transmission. Good responses recognised that the sending and receiving components would be constantly synchronised by a common clock.

Question part 3.3 was about the suitability of parallel and serial transmission for two different scenarios. This was particularly well answered, with a third of students achieving full marks. The most common reason for not achieving a mark was because a student did not explain why parallel transmission was suitable for use by the data bus.

For question part 3.4 students had to describe the role of an I/O controller. Students found this the most challenging part of question 3. They often failed to achieve marks because what they wrote was not enough rather than because it was wrong. For example, writing about the controller acting as an interface or being for communication without explaining that it would translate signals sent between the CPU and peripherals.

Question part 3.5 asked students to calculate the maximum amount of memory that a computer with 36 wires in the address bus and 16-bit memory locations could access, in gibibytes. Less than a quarter of students were able to do this correctly, with the most common mistakes being using 2^{16} in the calculation instead of 2^{36} and dividing by 1000 instead of 1024.

For question part 3.6 about a third of students were able to give an example of how the control bus is used when storing data. The most common response was that it would send a memory write signal. A small number of students confused the control bus with the control unit.

Question 4

Question 4 was about floating-point numbers, and all parts of the question were tackled well.

Question part 4.1 consisted of five statements about fixed and floating point representations and students were asked to identify all of the true statements. Most students correctly recognised that calculations could be carried out more quickly on fixed point numbers and that in a given number of bits a floating point system could represent a bigger range of values. Only a minority of students also recognised that in a given number of bits a fixed-point system could represent some numbers more precisely than a floating-point system. This is because in fixed point no bits are required to store a mantissa so all of the bits can be used to represent the significant digits in a number.

For question part 4.2 students had to convert a floating-point number to decimal. Most students achieved at least one mark for this and nearly half achieved both marks. The most common errors were shifting the binary point the right distance but in the wrong direction and failing to deal correctly with the fact that the value was negative.

Question part 4.3 was a conversion the other way, from decimal to floating point. This was tackled slightly better than part 4.2 despite the fact that the conversion was complicated as the number could not be represented exactly. The most common mistakes were to not move the binary point the correct number of places when normalising and to represent the exponent incorrectly. The majority of students recognised that three extra bits would be required to represent the number precisely for question part 4.4, which was pleasing as this type of question has not been asked previously.

For question part 4.5 students had to calculate the most negative number that could be represented in a different floating-point representation. Just under half of students were able to do this correctly.

Question 5

Question part 5.1 was an extended response question about how data would be represented using a magnetic hard disk drive and how it would be transmitted using the TCP/IP stack. Half of students achieved at least four marks for this question but fewer than ten percent achieved a mark of ten or above. Common mistakes were confusing magnetic disks and optical disks and writing about the TCP/IP stack without linking tasks to the individual layers, which the question had asked students to do. Most students failed to achieve marks for simply not knowing enough content or writing too vaguely rather than because what they wrote was incorrect. This was surprising as both of these topics are covered at GCSE as well as at A-level.

For question part 5.2 students had to state an advantage and a disadvantage of using SSDs in a file server instead of magnetic hard disk drives. This part was very well answered with most students achieving both marks. If a question has a context, then students should consider whether or not the advantage/disadvantage that they give is relevant in the context. For example, it was not considered markworthy that SSDs are quieter than hard disks in the context of a file server, but this point might have been markworthy in a different context.

Question 6

Question 6 was about logic circuits, flip-flops and Boolean algebra.

For question part 6.1 students had to draw a logic circuit for a given Boolean expression. This was very well tackled with the majority of students achieving all four marks. The most common mistakes related to misunderstanding the precedence of operators and so performing the operation OR between B and C and then ANDing the result with D, instead of ANDing the values of C and D before ORing the result with B.

Question part 6.2 asked students to explain how the output of an edge-triggered D-type flip-flop would be affected when a pulse was received on the clock input. Good responses recognised that the flip-flop would act as a memory unit and the output would be changed to reflect the current value of the data input at the time when the clock pulse occurred. Just over a quarter of students correctly recognised this, with the most common incorrect response being that the state of the output would change from 0 to 1 or 1 to 0, presumably based on the name flip-flop.

Question part 6.3 was a Boolean simplification and it was well tackled, with the majority of students achieving at least two of the four available marks. Approximately a third of students achieved all four marks by recognising that the expression could be simplified to $\mathrm{A} \oplus \mathrm{B}$ or $\overline{\mathrm{A}} \cdot \mathrm{B} + \mathrm{A} \cdot \overline{\mathrm{B}}$. Some students correctly got to the expression $\overline{\mathrm{A}} \cdot \mathrm{B} + \mathrm{A} \cdot \overline{\mathrm{B}}$ but then went on to further simplify this incorrectly.

Question 7

Question 7 was about the Internet. None of the question parts were tackled well and many students' knowledge of technical terminology was limited.

For question part 7.1 students had to identify the protocol and domain name in a URL. The most common errors were to give `www.loveapug.org.uk` or just `loveapug` as the domain name.

Question part 7.2 asked how domain names are organised. Good responses recognised that they are organised hierarchically and identified some of the parts such as top-level and second-level domains. Some students described how the Domain Name System is organised instead of how domain names are organised.

Question part 7.3 was about the role of Internet registries. Good responses recognised that an Internet registry would register domain names and their owners, to ensure that domain names were unique. As with part 7.3, some students wrote about the Domain Name System instead of the correct topic.

For question part 7.4 students had to explain how a computer could access a web server inside a LAN, using port forwarding. Whilst most students achieved at least one mark, fewer than ten percent achieved all three marks. Very few responses included the term port forwarding, and students often missed out on marks by not referring to how the IP addresses would be changed by the router.

Question part 7.5 was about why a web server might not be configured by DHCP. Good responses recognised that for port forwarding to work the web server would need to have a static IP address that would not change.

For question part 7.6 students had to describe how a router with Network Address Translation (NAT) would be used to allow a computer inside a LAN to access a server outside a LAN. The majority of students achieved at least one mark but very few achieved all four marks. The most commonly awarded marks related to how the router would change the IP addresses when packets of data were passed out of and back into the LAN. Far fewer students were awarded marks for successfully describing how port numbers and a NAT translation table would be used in the process.

Most students recognised that the IPv6 system would allow more unique IP addresses for question part 7.7 but some failed to achieve the mark because they did not go on to explain that there would be enough routable IP addresses for every device to be allocated a unique one.

Question part 7.8 was about the Websocket protocol. Just under half of students correctly identified that the protocol established a full-duplex communication channel. Students should ensure that they read each question carefully. Despite the question stating that only one option should be selected approximately fifty students selected two or more options.

Question 8

Question 8 was about databases and SQL.

For question part 8.1 students had to identify an assumption that had been made in the design of a database. The majority of students correctly identified that it had been assumed that each product was only supplied by one supplier.

Question parts 8.2 and 8.3 asked students to write an INSERT INTO and an UPDATE query. These types of query are asked about less frequently than SELECT queries and this was perhaps one of the reasons why the question parts were not tackled well. Many students did not use valid SQL commands. Those who did use valid commands often missed marks by making mistakes such as omitting delimiters around the date in question part 8.2 and putting the fieldname instead of the table name after the UPDATE command in question part 8.3.

Question part 8.4 was about timestamp ordering. The topic was not well understood and only a minority of students achieved any marks. The most commonly awarded mark was for identifying that each database transaction would have a timestamp recorded with it. Few students went on to explain how records in a table would have a read and write timestamp or how these would be used to prevent a loss of database integrity.

Question part 8.5 asked students to describe two problems that might occur if a database was not fully normalised. Just under half of students achieved some marks for this question. The most common reason for marks not being achieved was stating problems such as "data inconsistency" rather than describing them. Another common mistake was to state properties of normalised databases that would not be true of a database that was not normalised, such as not all attributes being dependent on the primary key, without explaining why this would be a problem.

Question 9

Question 9 was about the moral, ethical and legal issues that might arise as a result of using computer systems to assist with diagnosing medical conditions. The question was well answered with the majority of students achieving at least three of the six available marks.

The most common problem was focussing a response too narrowly on the issue of blame for a misdiagnosis, rather than considering a wider range of issues. Some students did not read the question carefully and discussed computer systems replacing humans instead of being used to assist with diagnosis, which was what was asked about.

Reponses that went beyond the issue of blame for misdiagnosis often covered topics such as training and bias, data security and consent and relevant legislation. The best responses covered complex issues such as whether or not an AI system should be used if it were less reliable than a human diagnosis but could be delivered at lower cost or to more people and what the long-term impact of the use of AI on the role of medical professionals could be.

Question 10

Question 10 was about assembly language.

For question part 10.1 students had to trace the execution of an assembly language program. This was very well tackled, and more than a quarter of students achieved full marks for it. The majority of these students were able to at least partially describe the purpose of the program for question part 10.2, although only about ten percent of students correctly identified that the program added an odd parity bit to a character code. The most common mistakes in the trace were shifting `R1` in the wrong direction, missing the value `4` in `R4` and not setting the parity bit, so not achieving the very last mark point. Some students who completed the trace successfully recognised that the number of 1s was being counted and so achieved one mark for question part 10.2 but did not interpret this in the context of character codes to deduce that a parity bit was being set.

For question part 10.3 students had to describe two advantages of writing programs in assembly language. Good responses referred to the fact that the resultant object code might execute quicker or use less memory than object code produced from a high-level language as a human might optimise the code better than a compiler, and that there might be better access to hardware such as registers or memory.

For question part 10.4 students had to explain what “imperative” meant in the context of a high-level language. Approximately a quarter of students made a valid point, most commonly that the program would describe how to perform a task by specifying the steps necessary to carry it out. The most common mistake was to state the opposite - that an imperative program described what a problem was rather than how to complete a task.

Question 11

Question 11 was about functional programming.

For question part 11.1 students had to state the co-domain of a function. Approximately half of students provided a suitable description of the set of real numbers. A common mistake was to describe the function type, ie to write that the function mapped from natural numbers to real numbers, without recognising what the co-domain was.

Question part 11.2 asked students to describe two features of functional programming languages that made it easier to write code that could be distributed to run across multiple servers. Students often failed to achieve marks as they simply stated features such as "data structures are immutable" instead of describing that this meant that their values could not be changed once they were assigned. Answers that achieved both marks most commonly did this by describing what statelessness and immutability are.

Mark Ranges and Award of Grades

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