

Principal examiner reports for teachers

Principal examiner reports for teachers summarise what candidates did well and what candidates needed to improve for this examination series only.

This report is to be used in conjunction with the question paper and mark scheme.

For guidance on teaching and learning we provide resources for many of our syllabuses.

Resource	Where to find it
Grade descriptions – Grade descriptions describe the level of performance typically demonstrated by candidates achieving the different grades awarded for a qualification.	School support hub Select: Subject > Resource finder > Syllabus and specimen materials
Schemes of work – Schemes of work are designed to support you in your teaching and lesson planning.	School support hub Select: Subject > Resource finder > Teaching and learning
Lesson planning – Guidance on how to plan a lesson.	Lesson planning
Resource plus – There may be additional resources such as teaching packs containing lesson plans, resources and worksheets and videos for aspects of your syllabus.	School support hub Select: Subject > Resource finder > Teaching and learning
Teaching tools	Teaching Tools
Exemplar candidate responses	School support hub Select: Subject > Resource finder > Teaching and learning
Learner guide – A guide for students to help them understand what the course involves.	School support hub Select: Subject > Resource finder > Teaching and learning
Community forums – Do you have a question or something to share? Our online forums are great way to keep up to date with your subject and the global Cambridge community.	Community forum on the school support hub
Access to scripts for this series.	Access to scripts service
Results analysis is available for some syllabuses.	Results analysis
Test maker is available for some subjects.	Test maker

Other services from Cambridge International Education

- [Digital mocks service](#)
- [Insight assessments](#)
- [Professional development](#)
 - [Focus on assessment](#)
 - [Online marking workshops](#)

COMPUTER SCIENCE

Paper 9618/11 Theory Fundamentals
--

Key messages

What candidates did well

- Many candidates demonstrated a good understanding of different number systems and were able to convert from one number base to another.
- The identification of the logic expression and the construction of the truth table were completed well.

What candidates needed to improve

- When specific information is given in a question, such as, for example, the school logo in **Question 5(c)**, candidates should ensure that their answers refer to that information.
- Candidates should take care not to repeat answers given in the question, for example, three modes of addressing are given in **Question 3(c)**.
- When a question asks for differences between two entities, such as in **Question 1(a)(ii)** and **Question 1(d)**, candidates should ensure that both entities are referred to for each difference.
- Candidates should ensure that they use correct identifiers. For example, in **Question 2(d)**, the names of the licences need to be exact.
- Some candidates need to read the questions carefully and ensure answers correspond to the question requirements.

Comments on specific questions

Question 1

(a) (i) Comprehensive responses

- Many candidates correctly calculated most of the missing values.

Limited responses

- Some candidates correctly calculated the missing denary and binary values.
- Some candidates need to improve their understanding of how to convert values into hexadecimal.

(ii) Comprehensive responses

- Comprehensive responses correctly identified an alternative character set, usually ASCII and clearly explained at least two differences with UNICODE.

Limited responses

- Some candidates correctly identified ASCII as an alternative character set.
- Some explanations of the differences with UNICODE needed greater detail.

(b) Comprehensive responses

- Some candidates correctly completed the conversion.

Limited responses

- Some candidates understood the words ‘two’s complement’ to mean that the value was automatically negative even though the most significant digit was zero.

- (c) **Comprehensive responses**
- Comprehensive responses correctly converted the denary number 30 to binary, then converted that binary value to its negative equivalent by flipping the bits and adding 1, before adding this calculated value to the given binary number using binary arithmetic.

Limited responses

- Limited responses often correctly converted the denary number 30 to binary, then omitted to add 1 when converting to the negative equivalent.

- (d) **Comprehensive responses**
- A few candidates correctly identified two differences between binary prefixes and denary prefixes.

Limited responses

- Many candidates overlooked the word 'prefix' in the question and wrote about differences between the binary number system and the denary number system.

Question 2

- (a) **Comprehensive responses**
- Some candidates gave complete answers describing the need to convert program code written in a language that is easier for human programmers to understand into a form that the computer processor can execute.

Limited responses

- Some candidates misread the question and described the need for computerised translation programs which convert between spoken languages such as English and French.

- (b) **Comprehensive responses**
- Many candidates were able to correctly identify two benefits of using an interpreter during the development phase of a computer program.
 - Comprehensive responses also included more detail about each of these benefits.

Limited responses

- Some candidates correctly identified two benefits.
- Some candidates need to include greater detail in their answers.

- (c) **Comprehensive responses**
- This part of the question was answered well by candidates who gave the correct name of the translator.

Limited responses

- Some candidates gave brand names of spoken language translation software.

- (d) **Comprehensive responses**
- Some candidates correctly identified the three types of software licence.

Limited responses

- Some candidates gave inappropriate names for the licences, such as, 'trial licence' for 'shareware'.
- There was considerable confusion between Free Software and Freeware.

Question 3

- (a) **Comprehensive responses**
- Some candidates successfully completed the three sets of instructions.

Limited responses

- Some candidates correctly completed the first ADD instruction without then increasing the contents of the Accumulator (ACC).
- Some candidates overlooked the # symbol in the LDM instruction and loaded the contents of location 200 to the ACC instead of the denary value 200.

(b)(i) Comprehensive responses

- Some candidates correctly used the AND instruction with an operand of zero.

Limited responses

- Some candidates misread the question and gave a data movement instruction instead of a bit manipulation instruction.

(ii) Comprehensive responses

- Some candidates correctly carried out the given shift instruction.

Limited responses

- Some candidates performed a cyclic shift instead of a logical shift.
- Some candidates shifted left instead of right.

(c) Comprehensive responses

- Comprehensive responses correctly identified an additional mode of addressing with an appropriate description and example.

Limited responses

- Some candidates need to include correct technical terms, such as 'operand', in their descriptions of the modes of addressing.
- Some candidates repeated modes of addressing given in the question.

Question 4

(a) Comprehensive responses

- Some candidates correctly labelled the order of the two relationships using a recognised convention and described each relationship in terms of the primary and foreign keys in each table.

Limited responses

- Some candidates attempted to define the relationships without using a recognised convention.
- Some candidates need to refer to the primary and foreign keys when describing a relationship.

(b) Comprehensive responses

- Comprehensive responses provided an SQL script which included statements to return all the elements requested.

Limited responses

- Limited responses provided an SQL script that satisfied some of the requirements.
- Greater understanding of the GROUP BY clause is needed.

(c)(i) Comprehensive responses

- Comprehensive responses provided an SQL script which included statements to add a field to the required table with an appropriate name and data type.

Limited responses

- Many candidates need to understand how the ALTER TABLE statement is used in SQL.

(ii) Comprehensive responses

- Comprehensive responses provided an SQL script which included statements to update the field added in **Question 4 (c)(i)** using a data type matching their previous answer.

Limited responses

- Some candidates provided an appropriate SQL condition statement.
- Many candidates need to understand how the UPDATE statement is used in SQL.

(d)(i) Comprehensive responses

- A few candidates gave complete, well-structured explanations of the ways in which validation and verification are used in ensuring the integrity of a database.

Limited responses

- Some candidates were able to give examples of either validation checks or verification checks.
- Some candidates wrote that validation ensures the data is valid, which is not enough and that verification ensures the data is correct, which is not always true.

(ii) Comprehensive responses

- A few candidates understood that the original data could have been wrong.

Limited responses

- Many candidates gave vague answers such as 'user error'.
- Some candidates need to understand that validation and verification checks do not necessarily ensure that the data is correct.

Question 5

(a) Comprehensive responses

- Some candidates gave good, complete answers describing how the colour of each individual pixel in the image is translated into a sequence of binary digits.

Limited responses

- Some candidates described compression methods instead of how the bitmap is encoded.
- Some candidates gave vague responses about the colours of the pixels and associated binary codes, without emphasising the unique correspondence.

(b) Comprehensive responses

- Few candidates were able to correctly identify the three vector graphic terms.
- Some candidates were able to correctly identify one or two of the terms.

Limited responses

- Some candidates inserted terms for bitmap graphics such as 'pixel'.
- Some candidates need to improve their understanding of the terminology associated with vector graphics.

(c) Comprehensive responses

- Comprehensive responses for both options were well-reasoned and appropriate. The design of the logo was considered also the different situations where the logo would be used and the various requirements of each.

Limited responses

- Some candidates justified their choice in terms of scalability of the logo or file size with little additional information.
- Some candidates gave generic answers about bitmap images or vector graphics.

Question 6

(a) Comprehensive responses

- Many candidates correctly calculated both of the outputs in the given logic circuit.

Limited responses

- Some candidates confused the XOR and NOR gates.

(b) Comprehensive responses

- Many candidates were able to correctly complete the truth table.

Limited responses

- Some candidates included three inputs in their table.

Question 7

(a) Comprehensive responses

- Some candidates gave two appropriate characteristics of a thin-client system.
- The idea that the client does minimal processing and that it is very heavily reliant on the server was often included in candidate responses.

Limited responses

- Limited responses referred to a client-server system without making any distinction between a thin-client and a thick-client.

(b)(i) Comprehensive responses

- A few candidates wrote about data being broadcast along a backbone and the use of terminators.

Limited responses

- Many candidates misread this question.
- Few candidates described data transmission in a bus network.
- Some candidates gave a full description of collision management in an Ethernet network.

(ii) Comprehensive responses

- Some candidates gave two appropriate drawbacks of a bus network. The most popular correct answers were problems with the main cable and privacy concerns.

Limited responses

- Some candidates were able to state that the complete shutdown caused by a break in the main cable was a drawback of this type of network.

(c)(i) Comprehensive responses

- A few candidates gave complete answers to this question stating what was meant by each of the two elements of the given term.

Limited responses

- Some candidates simply stated what was meant by a static IP address.
- Some candidates simply stated what was meant by a private IP address.

(ii) Comprehensive responses

- Comprehensive responses included three clear benefits of subnetting. Reduced congestion and scalability were popular correct choices.

Limited responses

- Some candidates gave vague answers about the network being faster.
- Some candidates need to understand that words like 'faster' are not enough without further expansion.

COMPUTER SCIENCE

Paper 9618/12 Theory Fundamentals
--

Key messages

What candidates did well

- Many candidates demonstrated a good understanding of different number systems and methods of conversion between them.
- The construction of a truth table from a given circuit with more than one output was completed successfully by many candidates.
- Discussions of the benefits and drawbacks of cloud storage were usually well written and comprehensive.

What candidates needed to improve

- When specific information is given in a question, for example, the image in **Question 1(d)** or the date of birth in **Question 2(b)**, candidates should ensure that their answers relate to that information.
- When a question asks for differences between two entities, such as in **Question 6(c)**, candidates should ensure that both entities are referred to for each difference.
- Candidates should ensure that they use the terminology correctly. For example, several parts of **Question 7** required a distinction between memory and file storage and **Question 7(a)(iii)** needed a distinction between data transfer speed and data access times.

Comments on specific questions

Question 1

(a) Comprehensive responses

- Many candidates correctly identified the description of a pixel.

Limited responses

- Some candidates chose one of the other options.

(b) Comprehensive responses

- Some candidates were able to clearly define what was meant by both image resolution and screen resolution.
- Comprehensive responses also included a statement about the difference between the two.

Limited responses

- Some candidates confused image resolution with sample resolution.

(c) Comprehensive responses

- Some candidates correctly identified another item included in a bitmap header. The most popular correct choice was the file size.

Limited responses

- Some candidates gave resolution or colour depth as their answer. These are both given in the question.

(d)(i) Comprehensive responses

- Some candidates correctly calculated the number of bits needed.

Limited responses

- Some candidates need to understand that if there are three different colours in an image, such as the one shown, then three different binary values are needed to uniquely identify each individual colour (00, 01, 10). Three values need a minimum of two bits.

(ii) Comprehensive responses

- Comprehensive responses identified a suitable method of lossless compression and described their chosen method in detail using examples taken from the image provided to illustrate the process.

Limited responses

- Less detailed candidate responses identified the method and gave partial explanations often referring to repeated characters instead of pixels.

Question 2

(a) Comprehensive responses

- Some candidates identified and fully described two appropriate methods of data verification. Double entry and a visual check were the most frequent correct choices.

Limited responses

- Although some candidates identified two appropriate methods of data verification, descriptions needed greater detail, especially about the need for a comparison with original data.

(b) Comprehensive responses

- Some candidates identified and described two appropriate methods of data validation. Format check, type check and length check were the most frequent correct choices.

Limited responses

- Although some candidates identified two appropriate methods of data validation, descriptions needed to be applied to the example date of birth given in the question.

(c) (i) Comprehensive responses

- Many candidates correctly described a virus and identified and described an appropriate method of restricting the risks to data caused by viruses. The most popular correct choice for restricting the risk was anti-virus software.

Limited responses

- Some candidates named a type of virus, rather than describing the generic term.
- The method of restricting the risk was often appropriate, descriptions needed greater detail.

(ii) Comprehensive responses

- Some candidates correctly described a hacker and identified and described an appropriate method of restricting the risks to data caused by hackers. The most popular correct choice for restricting the risk was a firewall.

Limited responses

- Some descriptions of a hacker needed a clear statement about unauthorised access.
- Methods to reduce the risk posed by hackers were often vague and needed greater detail.

Question 3

(a) Comprehensive responses

- Some candidates correctly described the order of two relationships between the three entities using a recognised convention.

Limited responses

- Some candidates attempted to define the relationships without using a recognised convention with the result that the descriptions needed greater detail.

(b)(i) Comprehensive responses

- Comprehensive responses identified the line numbers of the lines containing errors and gave appropriate corrections.

Limited responses

- Limited responses identified some of the line numbers of the lines containing errors.
- Some candidates identified lines that were correct as errors.

(ii) Comprehensive responses

- Comprehensive responses gave an SQL script which included statements to return all the elements requested.

Limited responses

- Limited responses gave an SQL script that satisfied some of the requirements. Greater understanding of the GROUP BY and ORDER BY clauses is needed.

(c) Comprehensive responses

- Some candidates were able to complete all the sentences appropriately.

Limited responses

- Some candidates completed some of the sentences appropriately.
- Some candidates repeated examples of data given in the question in their answers, for example, field names and relationships.

Question 4

(a) Comprehensive responses

- Some candidates were able to correctly describe what was meant by the given sampling rate.

Limited responses

- Some candidates left the kHz units in their answer.

(b) All responses

- Due to an issue with **Question 4 (b)**, full marks have been awarded to all candidates for this question to make sure that no candidates were disadvantaged.
- The issue has been corrected in the published version of the question paper.

Question 5

(a) Comprehensive responses

- Many candidates correctly calculated most of the missing values.

Limited responses

- Some candidates correctly calculated the missing denary and unsigned binary values.
- Some candidates need to improve their understanding of how to convert values into hexadecimal and Binary Coded Decimal (BCD).

(b) Comprehensive responses

- Some candidates correctly completed most of the missing values.

Limited responses

- Some candidates gave answers that were in the number systems different from those described in each of the statements.

Question 6

(a) Comprehensive responses

- Comprehensive responses contained well-structured answers that expanded on the characteristics of any generic embedded system by suitably applying them to the health monitor.

Limited responses

- Some candidates listed generic descriptions of any embedded system without showing any application to the health monitor described in the question.
- Some candidates repeated the statements from the question, such as 'it is an embedded system because it monitors a person's heart rate'.

(b) Comprehensive responses

- Some candidate answers successfully explained how the generic characteristics of any monitoring systems could be applied to the health monitor.
- Some candidates gave valid reasons why the health monitor could not be considered to be a control system.

Limited responses

- Some candidates listed generic descriptions of any monitoring system without showing any application to the health monitor described in the question.
- Some candidates gave vague, imprecise answers such as, 'it is a monitoring system because it does not control anything'.

(c) Comprehensive responses

- Many candidates were able to describe two differences between RAM and ROM, correctly referring to both RAM and ROM in each of their answers.

Limited responses

- Limited responses often only referred to one or other of RAM and ROM in each answer.

(d) Comprehensive responses

- There were many well-structured and informative discussions about the benefits and drawbacks of storing the data from the health monitor in the cloud.
- Both benefits and drawbacks were often discussed, with considerable emphasis on the positive aspect of ease of access and the negative aspect of possible privacy issues.

Limited responses

- Some candidates only discussed either the benefits or the drawbacks of storing data in the cloud.
- Some candidates needed to refer to the health data in their answers.

Question 7

(a) (i) Comprehensive responses

- A few candidates correctly identified two tasks performed by the Operating System as part of process management.

Limited responses

- Many responses included items that are performed by other management tasks, for example, allocation and management of memory.

(ii) Comprehensive responses

- A few candidates correctly identified two tasks performed by the Operating System as part of memory management.

Limited responses

- Many responses included tasks performed by File Management rather than Memory Management.
- There is a need to differentiate between memory and file storage.

(iii) Comprehensive responses

- Some candidates explained how defragmentation software reduced the amount of disk head movement, thus reducing access times and improving computer performance.

Limited responses

- Many of the descriptions of defragmentation software were vague.
- Some answers described increasing the transfer speed (bits per second), rather than decreasing access times.

(b)(i) Comprehensive responses

- Comprehensive answers named at least two of the tools available in an IDE for debugging and explained how they would be used to check the values of variables, or the logic of the code.

Limited responses

- Some candidates described the different tools instead of answering the question 'how they would be used'.
- Some candidates stated only that the debugging tools would be used to check for errors. This repeats what is meant by the term 'de-bugging'.

(ii) Comprehensive responses

- A few candidates explained why some high-level languages are partially compiled and partially interpreted.

Limited responses

- Many candidates tried to explain the process by incorrectly using an interpreter first to remove errors and then a compiler to produce executable code.
- There is a need for greater understanding of the process of partial compilation followed by partial interpretation.

Question 8

(a) Comprehensive responses

- Many candidates correctly calculated both of the required outputs in the truth table.

Limited responses

- Some candidates confused the XOR and NOR gates.

(b) Comprehensive responses

- Some candidates gave the correct four term answer.
- A few candidates recognised the negation of the XOR function.

Limited responses

- Some candidates gave expressions that were true for the first two rows only.

COMPUTER SCIENCE

Paper 9618/13 Theory Fundamentals
--

Key messages

What candidates did well

- The construction of truth tables from a given circuit was completed well and candidates were able to name missing gates.
- Many candidates were able to identify tasks carried out by the operating system.
- Many candidates demonstrated a good understanding of different number systems and methods of conversion between them.

What candidates needed to improve

- Candidates should ensure that when two examples are required, they should not answer with the same method for both, for example **Question 3(c)(i)**.
- Candidates should ensure that they use correct technical terminology whenever possible.
- When a specific context is given in the stem of a question, for example in Question 3, candidates should ensure that their answers are also in the given context.
- Candidates need to ensure that they understand the requirements of different command words specifically the difference between describe and state.

Comments on specific questions

Question 1

Comprehensive responses

- Some candidates were able to complete all parts of the sentences appropriately.

Limited responses

- Many candidates incorrectly identified checksum as the missing method of verification and length check as the second method of validation.

Question 2

(a) Comprehensive responses

- Many candidates correctly calculated both of the outputs in the given truth table.

Limited responses

- Some candidates incorrectly evaluated individual gates.

(b) Comprehensive responses

- Many candidates correctly identified the missing logic gates.

Limited responses

- Some candidates only identified one missing gate.

Question 3

(a) Comprehensive responses

- Many candidates correctly identified the meaning of sampling resolution.

Limited responses

- Some descriptions did not mention the fact that sampling resolution is the number of bits for each individual sample.



(b)(i) Comprehensive responses

- Candidates gave clear description of bitmaps being made up of pixels, each of which is assigned a code to represent its colour and stored in sequences. They also identified that each colour has a unique code.

Limited responses

- Some candidates incorrectly assumed that encoding referred to Run Length Encoding and incorrectly gave detailed descriptions of this.

(ii) Comprehensive responses

- Some candidates identified and performed the correct calculations.

Limited responses

- Some candidates omitted turning bytes into bits so were out by a factor of 8.

(c)(i) Comprehensive responses

- Comprehensive responses were able to identify two different methods of lossless compression.

Limited responses

- Some candidates only identified RLE as a method of lossless compression and applied this to both the image and the sound files rather than identifying a different method. Others incorrectly identified lossy methods rather than lossless.

(ii) Comprehensive responses

- Few candidates included both a description of public and cloud in their answers.

Limited responses

- Some candidates only included description of just one of public or cloud in their answer.

(iii) Comprehensive responses

- Some candidates identified and described, in context, two appropriate benefits for using the public cloud.

Limited responses

- Some candidates focused on cost without supporting justification. Others were under the misconception that public cloud equated to social media and that it would lead to sharing and a wider audience.

Question 4

(a) Comprehensive responses

- Many candidates correctly identified the three tables and their correct fields.

Limited responses

- Some candidates chose unsuitable primary keys for example name. Others overlooked identifying any foreign keys.

(b) Comprehensive responses

- Some candidates correctly identified the benefits of enforcing referential integrity.

Limited responses

- Some candidates just described the purpose of primary and foreign keys while others discussed the meaning of the word integrity or the use of validation checks.

(c)(i) Comprehensive responses

- Comprehensive responses identified the line numbers of the lines containing errors and gave appropriate corrections.

Limited responses

- Limited responses identified some of the line numbers of the lines containing errors.

(ii) Comprehensive responses

- Comprehensive responses included an SQL script which included statements to return all the elements requested.

Limited responses

- Limited responses comprised an SQL script that satisfied some of the requirements. Greater understanding of the `GROUP BY` and `ORDER BY` clauses is needed.

Question 5

(a) (i) Comprehensive responses

- Some candidates correctly identified two tasks performed by the Operating System as part of hardware management.

Limited responses

- Many responses included tasks performed by other subsystems such as file-management.

(ii) Comprehensive responses

- Many candidates were able to identify two tasks performed by the OS. Memory, file and process were the three most frequently seen.

Limited responses

- Some responses only identified one task.

(b) (i) Comprehensive responses

- Some candidates were able to identify the purpose of utility software.

Limited responses

- Some candidates gave examples of specific utility software rather than giving an overview.

(ii) Comprehensive responses

- Some candidates were able to identify the separate tasks carried out by a disk formatter.

Limited responses

- Some candidates confused disk formatter with file management or disk recovery software.

(c) (i) Comprehensive responses

- Some candidates were able to correctly define the term program library.

Limited responses

- Some candidates referred to the library routines as a single routine rather than as a collection of many related routines.

(ii) Comprehensive responses

- Some candidates understood the concept of DLL vs static library and were able to provide a comparison.

Limited responses

- Many candidates incorrectly believed that DLLs updated themselves and updated automatically. Some candidates described libraries in general rather than a comparison of dynamic vs static.

Question 6

(a) Comprehensive responses

- Many candidates correctly matched the denary numbers.

Limited responses

- Some candidates only drew one line from the denary number when the question highlights that there could be more than one equivalent for each.

(b) Comprehensive responses

- Many candidates correctly added together the binary numbers, clearly showed the workings and highlighted the overflow bit.

Limited responses

- Some candidates did not show their workings and others did not identify the overflow bit.

(c) Comprehensive responses

- Some candidates correctly converted the number.

Limited responses

- Some candidates incorrectly assumed that because it was a two's complement binary number, the denary number was negative.

(d) Comprehensive responses

- Many candidates answered this correctly.

Limited responses

- Some candidates incorrectly selected the first option.

Question 7

(a) Comprehensive responses

- Many candidates correctly identified all four modes of addressing in the question.

Limited responses

- Some candidates gave mnemonics such as LDD etc rather than the naming the address mode while others gave general computing terms such as ACC and ALU etc.

(b) Comprehensive responses

- A few candidates related bit-manipulation to control/monitoring. Few candidates understood isolating specific bits allowed control/monitoring.

Limited responses

- A few candidates just listed logical operations without relating them to monitoring/control.

(c) Comprehensive responses

- Many candidates only stated that 'bits are shifted right' for the similarity.

Limited responses

- Some candidates just assumed an arithmetic right shift places a 1 in the MSB.

Question 8

(a) Comprehensive responses

- Most candidates were able to name two alternative types of ROM and many were able to give a good description.

Limited responses

- Some candidates only gave a very brief description for their chosen type of ROM.

(b) Comprehensive responses

- Some candidates were able to give a detailed description of how a microphone works.

Limited responses

- Some candidates only referred to analogue to digital convertors turning analogue signal into digital.

(c)

Comprehensive responses

- Some candidates were able to describe the purpose of a buffer in relation to speakers.

Limited responses

- Many of the descriptions were vague and some candidates only described buffering out of context.

COMPUTER SCIENCE

Paper 9618/21 Problem-solving and Programming

Key messages

What candidates did well

- A very limited number of candidates were able to use the correct technical language when the question demanded it.
- A large minority of candidates wrote clear, well-structured program code which demonstrated both the production of an appropriate design for the given scenario and to follow this with accurate pseudocode using the correct structures.

What candidates needed to improve

- Candidates must prepare themselves with a more thorough understanding of the subject.
- Candidates would typically have sat this examination after attempting many of the past paper questions which are available. This includes questions requiring general subject knowledge and also questions which require the production of a problem design and its implementation with the pseudocode language.
- Candidates need to make the maximum use of the Insert provided with the list of functions and operators used within the pseudocode language and practice their use.
- Candidates should read any question carefully and appreciate that the key word in the question stem is usually a good indicator of the depth of answer which is required. 'State ...', 'Name ...' are indicators that a few words or a phrase will suffice. Key words such as 'Explain ...', 'Describe ...' require more detailed answers.
- The number of blank lines for the answer in the answer booklet also acts as an indicator.
- Often the candidate's pseudocode answer shows considerable amendments and has reached the stage where a complete rewrite is required to express a clear solution; typically on this paper for **Questions 8(a) and 8 (b)**. In this case the candidate should produce the final answer on additional paper sheets, with a short message in the appropriate place on the question paper that this has been done. The alternative is that the final answer is produced on one of the blank pages in the examination booklet. Any rough working should be clearly crossed out and marked as 'Rough Work'.

Comments on specific questions

Question 1

(a) Comprehensive responses

- A minority of candidates used the correct technical language; a count-controlled loop.

Limited responses

- Candidates need to be familiar with the three basic structures used to implement iteration in a problem design.
- Some candidates described the code as containing a `FOR` loop.

(b) Comprehensive responses

- Candidates successfully traced the pseudocode to conclude the algorithm would perform four iterations.

Limited responses

- A variety of incorrect numbers were suggested ranging from 1 to 9.
- Candidates need to understand the optional `STEP` clause used within a `FOR` statement.

Question 2

(a) (i) Comprehensive responses

- The majority of candidates correctly stated that the data type would be `BOOLEAN`.

Limited responses

- Some answers wrongly suggested `STRING` or `CHAR`.

(ii) Comprehensive responses

- Answers were seen both as a pseudocode statement or in prose.

Limited responses

- Candidates did not use the pseudocode syntax `AND`, instead used one of the programming language equivalents. Candidates should refer to the Insert provided with the paper which gives examples of the Boolean operators.

(iii) Comprehensive responses

- There were two alternative solutions possible; one using an `IF` structure or a single statement assigning a value to `WashStart`.

Limited responses

- Common errors seen were:
 - The use of the back-arrow character as the comparison operator.
 - The omission of the brackets around the `Tray1 OR Tray2` clause.

(b) (i) Comprehensive responses

- The correct answer of 'report window' (or an equivalent such as variable window or watch window) was rarely seen.

Limited responses

- Candidates should be familiar with the features and their technical terms used with an IDE; vague general answers such as 'output display' or 'console output' were insufficient.

(ii) Comprehensive responses

- Candidates successfully traced the given pseudocode and the use of the `INT` function produced an integer answer of 103.

Limited responses

- Candidates should be able to trace a sequence of given pseudocode statements. The paper's insert of functions and operators included the use of the `INT` function.

Question 3

(a) Comprehensive responses

- The stem of the question asked for 'an activity'; therefore, worthy answers included a key word such as discussion, investigation, observation, the production of something, or some indication that an activity was being carried out; for example *'Arrange and carry out a focus group of customers'*.

Limited responses

- Candidates needed to be clear after reading the question that this was referring to the analysis stage only. Often answers referred to other stages in the program development cycle such as testing or a feasibility study.

(b) Comprehensive responses

- The majority of candidates were familiar with some of the technical language used for methods of testing.

Limited responses

- This was another question that demanded the use of the correct technical terms.

Question 4

Comprehensive responses

- A minority of answers showed a solution with an attempt at distilling the problem into relevant steps.

Limited responses

- Candidates were unable to make the transition from the description of the scenario into a sequence of steps.
- Key errors and/or misunderstandings included:
 - Candidates often confused the requirement to calculate a total of the integers with a count of the number of input values that satisfied some condition.
 - The description required a loop. Candidates need to make clear where any loop begins and ends and its terminating condition.
 - Many answers merely repeated phrases from the rubric of the question; for example '*The user will input a sequence of integers*'.
 - The answer needed to make clear that the processing was to consider a single integer a number of times.

Question 5

(a) Comprehensive responses

- Few answers were seen which demonstrated a clear understanding of the problem.

Limited responses

- The solution required:
 - A function header which from the stem required a single integer value for the number of bulbs purchased.
 - The solution then required a calculation of the number of packs of 20, and any remainder.
 - The cost was calculated from the two totals and returned as a `REAL` value.
- Again, the candidate should have made use of the list of functions in the Insert to show the use of the `MOD` and `DIV` functions.
- Candidates should understand that an alternative to using these functions was to create a loop incrementing the number of batches on each iteration, decreasing the number of bulbs by 20 on each iteration and terminating the loop when the remaining bulbs was less than 20.

(b) Comprehensive responses

- A correct statement was in the minority.

Limited responses

- Candidates should appreciate that the `CALL` keyword is not used within this assignment statement, which called the `BulbCost` function.

Question 6

(a) Comprehensive responses

- Two correctly formed statements using the string handling functions were in the minority.

Limited responses

- Again, consulting with the Insert should have directed the candidate to the `MID` and `LEFT` functions to provide the answer.

(b) Comprehensive responses

- Solutions used the `MID` and `LENGTH` functions in combination, or `RIGHT` and `LENGTH` to compute the description string.

Limited responses

- A short question like this should be a vehicle to demonstrate understanding of the key features of a function.

- The solution required the stages:
 - Decide the parameter and its data type.
 - Decide the data type of the value to be returned.
 - Structure the statements within the correct framework and order:
 - Write the header, declare any local variables required, perform the calculation and finally return the value.

(c) Comprehensive responses

- The solution used the correct file handling functions, in the correct order and using the correct syntax.

Limited responses

- The solution did not attempt to isolate the data items for the length and the data, and this was not followed by a selection statement using these values.

Question 7

(a) (i) Comprehensive responses

- Candidates came up with an application which clearly would operate as a queue; popular examples included a shared printer queue on a network or a music playlist.

Limited responses

- Candidates should appreciate that there are practical examples of the use of the abstract data types in the syllabus.

(ii) Comprehensive responses

- Candidates followed through their answer for **Question 7(a)(i)** by stating that the first item to join the queue will be the first item to leave.

Limited responses

- Some candidates did choose an appropriate application and did not then follow this up by stating that it operates on the FIFO principle.

(b) (i) Comprehensive responses

- Where declaration statements were present they were usually correct and were seen in a minority of answers.

Limited responses

- The importance of ‘reading the question’ was an issue here as many answers were seen such as `EnQueue()` and `DeQueue()` modules which did not answer the stated question.
- Where a declaration was attempted it often had errors in the pseudocode syntax – omission of the `ARRAY` keyword, `MyQueue` and `ARRAY` reversed and incorrect brackets or range(s).

(ii) Comprehensive responses

- Correct answers were rarely seen. If the data values were in the correct locations and in the correct order, the pointer values were often omitted.

Limited responses

- Candidates need to understand how the contents of the queue are continually changing and how its current state is illustrated by the values of the `Front` and `Rear` pointers.

(iii) Comprehensive responses

- Very few answers illustrated that the concept of a circular queue was understood.

Limited responses

- Candidates need to understand how the circular behaviour of the queue maximises the storage space available.

(c) Comprehensive responses

- Completely correct solutions were rarely seen.

Limited responses

- Issues apparent in the pseudocode seen in the earlier pseudocode questions continued here.
- The solution required the basic design:
 - Setting up of a `FOR` loop.
 - Assignment to each array element of the 'empty string' value.
 - Assigning an initial value to both the `Front` and `Rear` pointers.

Question 8

(a) Comprehensive responses

- Many candidates successfully traced the algorithm.

Limited responses

- Although the binary search is an A2 level topic, the algorithm was given here and its trace was straightforward.

(b) Comprehensive responses

- The correct answer was seen from just under half of the candidates.

Limited responses

- Occasionally, the number was correct and the 'Found' text had been omitted.

(c) Comprehensive responses

- The correct answer was seen from just under half of the candidates.

Limited responses

- The incorrect answer 15 was frequently seen.

Question 9

(a) Comprehensive responses

- Some answers with completely correct syntax were seen, with these answers representing a minority of responses.

Limited responses

- Accuracy was required here, and errors included:
 - incorrect use of brackets.
 - declaration using a data type other than `CHAR`.
 - variations in the order of the syntax.

(b) Comprehensive responses

- Some good answers were seen and these were in the minority, with many candidates not meeting the requirements of the question.

Limited responses

- There were many candidates who made no attempt at a solution.
- Errors and omissions where an attempt had been made included:
 - no declaration of variables which were used.
 - no loop present.
 - no conditional test was present and if so, the delimiters had been omitted from the character `A`.

(c) (i) Comprehensive responses

- Relatively few candidates used appropriate field names.

Limited responses

- Common errors were the omission of the `DECLARE` and/or `ENDTYPE` keywords.

(ii) Comprehensive responses

- A similar level of response to that seen for the earlier answers to **Question 9 (b)**.
- This question had the additional complexity of the use of the record structure with the array elements.

Limited responses

- For candidates who make some attempt at a solution, the errors included:
 - The basic design was not two nested loops.
 - Candidates did not use the correct pseudocode for the array fieldnames. For example, `Version2[Floor, Room].View`
 - Errors with the syntax of an `OUTPUT` statement where the need was to concatenate an integer value within the string.

COMPUTER SCIENCE

Paper 9618/22 Problem-solving and Programming
--

Key messages

What candidates did well

- Many candidates showed good knowledge and understanding of the syllabus topics that were assessed.
- Candidates demonstrated a range of programming skills in the pseudocode-based questions.
- Candidates demonstrated a good understanding of how a stack operates.
- Candidates recognised when to use identifiers and meaningful variable names to provide correct answers.

What candidates needed to improve

- In explain-type questions, candidates need to use concise and clear explanations that make use of the correct technical language.
- This is a technical subject and questions may require specific answers using precise terms.
- The use of the provided insert to ensure the pseudocode functions are used correctly.
- When writing algorithms, candidates need to pay attention to key points of the question when writing pseudocode statements.

Comments on specific questions

Question 1

(a) (i) Comprehensive responses

- Most candidates completed the declaration statement correctly using the data type `STRING`.

Limited responses

- A few candidates incorrectly used `VARCHAR` as the data type. `VARCHAR` is not mentioned in **section 10.1** of the syllabus as an appropriate data type so should not be used in answers for Paper 2.

(ii) Comprehensive responses

- Most candidates completed the pseudocode statement correctly.

Limited responses

- Some candidates incorrectly used `STR_TO_NUM()` or `Characters()` instead of `LENGTH()`.

(iii) Comprehensive responses

- Many candidates completed all the pseudocode statements correctly.

Limited responses

- MP 3: A few candidates used a literal value rather than `LENGTH(Pass)`.
- MP 4: A few candidates extracted more than a single character or used a function other than `MID()`.
- MP 5: A few candidates incorrectly used a function other than `IS_NUM()`.
- MP 6: A few candidates set `LetterFlag` to `FALSE`.
- MP 7: A few candidates used `FALSE` (without equals) rather than `NOT`.

(b)(i) Comprehensive responses

- Some candidates correctly identified two other states often referring to activation or the alarm sounding.

Limited responses

- Some candidates did not understand the term state as used in state-transition diagrams.
- Many candidates incorrectly identified the 'off state' as one of the possible states.
- Many candidates gave answers not related to the scenario given.

(ii) Comprehensive responses

- Some candidates correctly identified two or three other events often referring to 'intruders entering a building' or 'triggering the alarm by opening a door' or 'entering a password when the alarm was ringing'.

Limited responses

- Some candidates did not understand the term event as used in state-transition diagrams and gave answers that were states.
- Some candidates incorrectly repeated the examples given in the question stem.

(c) Comprehensive responses

- Many candidates gave five correct examples of test data along with correct reasons for why test data was invalid.

Limited responses

- Some candidates did not follow the directive in the question to only break one of the three validation rules in each of the examples of test data used.
- Some candidates used lower case letters which instead of upper case letter
- Some candidates gave an incorrect reason for what was a valid example of test data for the rule being tested.
- Some candidates repeated the example reason given in the question.

(d) Comprehensive responses

- Many candidates gave three correct answers.
- The most common correct answers seen were:
 - Flowchart
 - Identifier Table
 - Choice of programming language.

Limited responses

- Some candidates gave one or two correct answers.
- Some candidates repeated examples given in the question.

(e) Comprehensive responses

- Most candidates gave two correct answers.

Limited responses

- A few candidates gave one correct answer.
- A very small number of candidates gave two incorrect answers.

(f) (i) Comprehensive responses

- Many candidates identified the correct term required.

Limited responses

- Many candidates gave incorrect terms, often alternative methods of testing.

(ii) Comprehensive responses

- A few candidates correctly identified the method of testing and gave a fully correct description.

Limited responses

- Many candidates identified the testing method, although the description given was incorrect or did not include sufficient detail.

Question 2

(a) Comprehensive responses

- Most candidates gave a correct explanation of the purpose of abstraction.

Limited responses

- The explanation given by a few candidates was too vague and demonstrated limited precision.
- Two typical incorrect answers were:
 - 'to simplify the program'
 - 'so that the data can be easily understood'.
- Other candidates gave answers that were not related to abstraction.

(b) Comprehensive responses

- Most candidates gave four correct data items.

Limited responses

- Some candidates gave two data items related to customers, such as customer name and customer address. Data items needed to be distinct.
- Some candidates gave answers outside the scenario such as 'number of items'.

(c) Comprehensive responses

- Many candidates gave two correct answers related to the scenario.

Limited responses

- Many candidates only gave one correct answer.
- Some candidates gave two answers that were too similar.

Question 3

Comprehensive responses

- Some candidates gave well-structured answers and presented a coherent algorithm using structured English that correctly answered the question.
- Many candidates' provided good answers that correctly addressed many of the required steps. In some cases, these could have been more logically structured thus, making the answer more coherent.

Limited responses

- The steps presented by some candidates used vague descriptions and could have been improved by more precision relating to the steps involved.
- Some candidates answers could have been improved by clarity related to the 'flow' of the algorithm.
- Most candidates' answers contained one or two correct steps.
- Some candidates repeated verbatim the phrases from the question.

Question 4

(a) Comprehensive responses

- Many candidates correctly answered this question by completing all six stack diagrams as required.
- Some candidates incorrectly completed the group two stack diagrams.

Limited responses

- Most candidates correctly completed some part of the diagrams with MP 2 and MP 6 being the most commonly given.

(b) Comprehensive responses

- Some candidates correctly identified an appropriate abstract data type (ADT) and described two distinct features of the ADT identified.
- Most candidates chose the queue as the ADT to describe.

Limited responses

- Many candidates correctly identified an appropriate ADT and gave incorrect or partial descriptions for its features.

- The description mark could have been more detailed.
- Some candidates gave multiple features that were too similar to each other.

Question 5

(a) **Comprehensive responses**

- Most candidates correctly described the change required.
- Candidates either gave a clear description of the change required or gave a correct example of how the specific pseudocode statement could be rewritten.

Limited responses

- A few candidates incorrectly stated that one of the array indices needed to be replaced by `Index - 1`.

(b)(i) **Comprehensive responses**

- A few candidates gave correct explanations of why algorithm B was more efficient than algorithm A that clearly referred to both the inner and outer loops.

Limited responses

- Many candidates' answers were vague and could have been improved by ensuring they identified which loop they were referring to and using technically correct language, for example using the term `sort` instead of `arrange` and `algorithm` instead of `program`.
- Many candidates correctly stated:
 - there were fewer iterations or comparisons
 - or
 - the algorithm ends once the data is sorted.

(ii) **Comprehensive responses**

- Many candidates correctly stated that the array is already sorted.

Limited responses

- Many candidates incorrectly paraphrased lines of code to answer the question such as 'the algorithm ends when no more swaps are made' these answers needed further details to make a connection with the original data in the array.

(c)(i) This was a demanding question requiring a precise explanation.

Comprehensive responses

- Few candidates correctly answered this question.
- Reference to `BYVAL` or `BYREF` was the most commonly seen correct answer.
- Fewer candidates referred to the array not changing.

Limited responses

- Most candidates' answers did not explain that the parameters were passed by value rather than by reference.

(ii) **Comprehensive responses**

- Few candidates correctly answered this question.
- Few candidates rewrote the pseudocode header using `BYREF`.

Limited responses

- Many candidates incorrectly used global variables when rewriting the pseudocode. The question clearly stated global variables should not be used.
- A few candidates incorrectly attempted to rewrite the procedure as a function rather than amending the procedure as directed to do in the question.

Question 6

(a) Comprehensive responses

- A wide range of viable solutions were seen.
- The vast majority of solutions were based around a two-loop design; the first loop to input the `Plaintext` and the second to pad it out as necessary with spaces to the required length.
- The vast majority of candidates recognised the need to use a loop to ensure the `Plaintext` inputted was between 1 and 25 characters in length.
- Many candidates correctly used a second loop to add the required number of spaces to the `Plaintext` inputted before returning the amended `Plaintext`.
- An occasional mistake was to calculate the length of the string *before* the second loop then never re-calculate it *inside* the loop (following the concatenation of a space character) so giving an infinite loop.

Limited responses

- A few candidates omitted loops altogether and attempted a solution based solely on conditional clauses. These solutions allowed for a single re-input of `Plaintext` (no MP 5 as the input was not within a loop).
- A few candidates incorrectly attempted to generate a padding string by multiplying the space character by an integer value.

(b)(i) Comprehensive responses

- Many candidates gave the correct syntax for declaring the upper and lower bounds for both of the array dimensions.
- Fewer candidates used the correct syntax for the rest of the declaration. In many cases, the incorrect data type of `STRING` was used instead of `CHAR`.

Limited responses

- Many candidates used incorrect syntax for the declaration which this question was testing.

(ii) Comprehensive responses

- A wide range of viable solutions were seen.
- Many candidates' solutions correctly generated the position value of the character to extract from the string along with the indices for the array where this character had to be stored. Correct solutions were most frequently based on initialisation of `Index` before the loops followed by incrementing within the inner loop.
- A few candidates used the same loop structure that populated the array to be generate and output each row of the array.
- Most candidates who successfully outputted each row of the array did so using a second loop structure.

Limited responses

- A few candidates made no attempt at producing a viable solution. This was often limited to just a procedure heading.
- Many candidates attempted a triple nested solution such as:

```
FOR Index ← 1 TO 25
  FOR Row ← 1 TO 5
    FOR Column ← 1 TO 5
      Grid[Row, Column] ← MID(Plaintext, Index, 1)
    NEXT Column
  NEXT Row
NEXT Index
```

Although the above is not fully correct, it partially met the requirements for populating the array.

- Some candidates used a nested loop when attempting to output each row although what they actually did was to output each character on a separate line.

COMPUTER SCIENCE

Paper 9618/23 Problem-solving and Programming

Key messages

What candidates did well

- Candidates were able to use the correct technical language when the question demanded it.
- Many candidates wrote clear, well-structured program code which demonstrated both:
 - A sound level of understanding the correct pseudocode.
 - A basic design using appropriate control structures for the given question scenario.
- Candidates appreciated that more than one or two-word answers were often required by the question and wrote clear prose to frame their answer.

What candidates needed to improve

- Candidates should read the question carefully and appreciate that the key word in the question stem is usually a good indicator of the depth of answer which is required. 'State ...', 'Name ...' are indicators that a few words or a phrase will suffice. Key words such as 'Explain ...', 'Describe ...' require more detailed answers.
- The number of blank lines for an answer in the answer booklet also acts as an indicator.
- Often the candidates' pseudocode answer shows considerable amendments and has reached the stage where a complete rewrite is required; typically on this paper for **Questions 8(a)(ii)** and **8 (b)(ii)**. In this case the candidate should produce the final answer on additional paper sheets, with a short message in the appropriate place on the question paper that this has been done. The alternative is that the final answer is produced on one of the blank pages in the examination booklet. Any rough working should be clearly crossed out and marked as 'Rough Work'.
- Candidates need to make the maximum use of the Insert provided with the list of functions and operators used within the pseudocode language.

Comments on specific questions

Question 1

(a) Comprehensive responses

- Candidates identified that this code used a post-condition loop structure.
- Variations of this were 'post-conditional iteration'.

Limited responses

- Many candidates omitted the key words indicating that the conditional test is done at the end of the block of code.

(b) Comprehensive responses

- Candidates traced the pseudocode correctly to state it would carry out five iterations.

Limited responses

- A common incorrect answer was six iterations.

Question 2

(a) Comprehensive responses

- The majority of candidates correctly stated that the three variables would have a data type `BOOLEAN`.

Limited responses

- No incorrect responses were seen.

(b) Comprehensive responses

- Candidates produced a correctly formed `IF` statement assigning the `DisplayOn` variable `TRUE` when `MainsOn` was `TRUE`.
- An alternative correct answer seen was a single assignment statement where `DisplayOn` is assigned the current value of `MainsOn`.

Limited responses

- Candidate answers contained errors in the pseudocode syntax, including:
 - The use of `WHEN` in place of `IF`:
 - `WHEN MainsOn = TRUE THEN DisplayOn ← TRUE`
 - Missing `ENDIF` statement.
 - Incorrect use of the assignment operator:
 - `IF MainsOn = TRUE THEN DisplayOn ← TRUE`

(c) Comprehensive responses

- Candidates described the operation of the robot such that it is ready to move when the mains power is on and the user has selected a non-zero value for the suction strength.

Limited responses

- Answers were not descriptive for the actual operation; they merely repeated the pseudocode.

(d)(i) Comprehensive responses

- Candidates gave the correct technical term for this IDE feature.

Limited responses

- Many candidates were not familiar with the technical term 'breakpoint'.

(ii) Comprehensive responses

- Candidates correctly substituted the value given to calculate the time remaining.

Limited responses

- Candidates did not use the `INT` function in the pseudocode and so stated an answer for the time remaining which was not an integer value.

Question 3

(a) Comprehensive responses

- Answers that correctly identified all four stages of the project were rarely seen. Most answers did appreciate this was all taking place at the analysis stage of the project.

Limited responses

- Candidates were often confusing the alpha and beta stages of testing. Few answers stated that after testing the software on the company's hardware 'acceptance' testing is completed. Other incorrect terms such as 'white box testing' and 'black box testing' were often seen.

(b) Comprehensive responses

- The majority of candidates were aware of the technical terms of Rapid Application Development and the Waterfall strategy for the program development life cycle.

Limited responses

- Some candidates were able to recall the two technical terms but were unable to follow this with a convincing feature of each approach. A feature of the waterfall method often hinted that something was happening 'step-by-step' with no detail. For RAD most correct answers referred to the involvement of the customer or the early production of a prototype.

(c) Comprehensive responses

- The majority of candidates were able to correctly state an activity which would be carried out; common correct answers included the selection of the programming language to be used or the production of various charts to illustrate some aspect of the design.

Limited responses

- The keyword in the question was to state an 'activity' which would be carried out, so answers such as 'flowchart' needed to be part of a clear activity; an answer such as '*The design of the user interface*' was required.
- Answers that referred to task allocation, the need to cost the project or consideration of the time scales were all considered to be inappropriate at this design stage.

Question 4

Comprehensive responses

- Few answers fully addressed the requirements of the question. A good answer would clearly state that it was a single input value which was to be considered and that it was a count of matching input values that was required (not the total of all the matching numbers).
- A good answer made clear where the loop started and ended and remembered to initialise the count before the start of the loop and output the calculated value after the loop had ended.

Limited responses

- Candidates often omitted one or more of the key stages of the algorithm. The most common errors were to confuse a total with a count and/or statements which did not make clear where a loop began and ended.
- Some answers did no more than repeat statements from the rubric of the question stem such as '*Input a sequence of integers*' (plural ...) without demonstrating an understanding that it was individual integer values which were being processed.
- Candidates were unable to make the transition from the general problem description in the stem of the question to an algorithm consisting of a sequence of steps.
- A common confusion was to attempt to combine two or more activities into a single step, so making the algorithm description unclear.

Question 5

(a) Comprehensive responses

- Some good answers were seen, which contained all the key aspects of the design – a correct function header followed by the combination of the `MID` and `LENGTH`, or `RIGHT` and `LENGTH` functions to calculate the string to be returned.
- Some candidates used a local variable for the return text, others simply calculated and returned the value in a single statement.

Limited responses

- Insufficient attention to detail; the most common error was a missing `ENDFUNCTION` statement.
- Many answers contained no more than the function header.
- The inappropriate use of `Length` as an identifier name.

(b) Comprehensive responses

- Generally well answered by the majority of candidates using the correct file handling pseudocode syntax.

Limited responses

- Attention to detail was important; a common error was to omit the speech mark delimiters from the file name.

Question 6

(a) Comprehensive responses

- A variety of responses. Almost all candidates stated the start pointer as 5; only some answers then completed the `Data` and `Pointer` table entries correctly.

Limited responses

- A common error was to populate the `Data` array in the same sequence as the nodes appeared in the list – ANT with index 1, BISON with index 2, etc.

(b) Comprehensive responses

- The majority of candidates stated that this indicated the end of the linked list.

Limited responses

- Some answers were insufficiently specific in relation to the linked list. For example *'There are no more values'* or *'There are no more nodes'*, which did not provide sufficient detail.

(c) (i) Comprehensive responses

- Well answered with a diagram showing that the BISON node would be bypassed.

Limited responses

- Some answers incorrectly suggested that the BISON pointer would remain unchanged at 2.
- A small number of answers did not complete all the flow lines and arrows on the revised diagram.

(ii) Comprehensive responses

- A common correct answer was that the node to be deleted is at the start of the list. The answer then followed this by explaining that this would require a change to the start pointer.

Limited responses

- A common incorrect answer was that the node to be deleted is at the end of the list (This does not constitute a 'special case').
- The two other correct answers which were often not stated **were** that the item is not in the list or is the only item in the list.

(d) Comprehensive responses

- Candidates found this question challenging. Answers were seen that stated that the new item would be stored in the `Data` array with index 6 (or some explanation that the value is to be placed at the next available 'free list' position). This was then followed by stating that the pointer for ANT would now connect to BAT and the pointer from BAT would link to BISON.

Limited responses

- The expectation of the question was that candidates would describe the sequence of steps which start with the node pointed to by the start pointer and then move through the linked list until the first node was found which was greater than the new value. Then the pointer changes can be made, and this requires the position of the previous node that was visited to be retained.

Question 7

(a) (i) Comprehensive responses

- Many complete and accurate traces were seen.

Limited responses

- An inaccurate trace table was the result of an incorrect number of iterations of the FOR loop. This would be then followed by the assignment of the data value to the incorrect element of the `Data` array.
- No general significant pattern of mistakes was apparent, and common errors were:
 - stopping the `Index` column short or extending it after the second 4 value.
 - Putting the final `Data` value 6 to `Data[5]` rather than `Data[4]`.

(ii) Comprehensive responses

- This answer required a clear explanation. Many candidates correctly used the term 'factors'. Other candidates correctly stated that these are the list of numbers which, when divided into 12, produce no remainder.

Limited responses

- Answers often fell short of a clear explanation.

(b)(i) Comprehensive responses

- Well answered by the majority of candidates showing the changing `Index` values of 4, 3 and 2 producing a `Match` value of `FALSE`, `FALSE` and `TRUE` respectively.

Limited responses

- Errors seen were rare and included extending the `Index` column below 2, and omitting the first `FALSE` value from the `Match` column.

(ii) Comprehensive responses

- As for the similar answer for **(a)(ii)**, this required a clear explanation that the value is the highest common factor of 8 and 20.

Limited responses

- The inclusion of the key word 'highest' was a requirement here to clarify the purpose of the algorithm.

Question 8

(a)(i) Comprehensive responses

- The majority of candidates used the correct pseudocode syntax for the declaration of the array.

Limited responses

- A variety of minor syntax errors were seen – incorrect order of the array name (`Bins`) and keyword `ARRAY`, incorrect brackets, incorrect array ranges, and incorrect `STRING` data type.

(ii) Comprehensive responses

- Many candidates successfully addressed all the requirements of the question. This required a correct basic design (a nested bins loop inside an outer levels loop). Other detail required was the use of a counter and code to produce the correct Level header.
- Some candidates did not use an outer loop and carried out the processing of the bins on each level with three separate blocks of code and three separate bin totals. This was considered of equal worth to the nested loops design.

Limited responses

- The solution contained minor pseudocode errors – often with the `OUTPUT` statement(s) – typically an attempt to concatenate an integer (the loop counter) with a string.

(iii) Comprehensive responses

- Well answered by almost all candidates, who realised that the five missing test data values could be determined from the sample output shown in the question rubric.

Limited responses

- No obvious errors were seen.

(b)(i) Comprehensive responses

- Well answered with candidates demonstrating meaningful variable names of the correct data type. It was expected that the record structure would contain a field for the active state of the bin.

Limited responses

- Errors included:

- omission of the keyword `DECLARE` before each variable.
- declaring three variables (`s`, `m` and `l`) in place of `Size`.
- declaring `Size` as data type `CHAR`.

(ii) Comprehensive responses

- Some quality solutions were seen demonstrating all aspects of understanding of the design and its pseudocode implementation.

Limited responses

- A correct design was often present in the solution. The syntax for the 2D array fields was incorrect and the correct formation of the string to be returned.
- Some answers did not appreciate that the design required both the return of the string and its output and if both were attempted, an error was to attempt to produce the output after the `RETURN` statement.

COMPUTER SCIENCE

Paper 9618/31 Advanced Theory
--

Key messages

What candidates did well

- Most candidates were able to convert a denary number to its binary floating-point equivalent and vice versa.
- Most candidates were able to list the benefits and limitations of using a virtual machine.
- Most candidates were able to complete a truth table from a given logic circuit diagram that included logic gates with up to three inputs.
- Most candidates were able to simplify a Boolean expression using a Karnaugh map (K-map).
- Many candidates were able to convert a Reverse Polish Notation (RPN) expression to infix form.
- Most candidates were able to demonstrate the use of Dijkstra's algorithm and find the shortest path from a home node to each of the nodes in a given diagram.
- Most candidates were able to write new facts for a given declarative language scenario.

What candidates needed to improve

- Some candidates were not able to complete the pseudocode to represent an insertion sort algorithm and instead gave a bubble sort algorithm.
- Some candidates did not provide sufficient detail, or use sufficient subject-appropriate language, in their answers for explanation or description questions to cover many of the requirements.
- Some candidates inappropriately used words from the question as a means to answer the question. For example, to describe the purpose of a pointer user-defined data type, they might say 'it points to a memory location'.

Comments on specific questions

Question 1

(a) Comprehensive responses

- Most candidates correctly named set, class/object or both as other types of composite user-defined data types.

Limited responses

- Some candidates incorrectly named enumerated as a composite user-defined data type, whereas that is a non-composite type.

(b)(i) Comprehensive responses

- Some candidates correctly stated that a pointer user-defined data type is used to reference a memory location.

Limited responses

- Some candidates inappropriately used the word 'point' to state the purpose of the pointer user-defined data type.
- Some candidates gave answers that were not specific enough, such as, 'identifies the location of something'.

(ii) **Comprehensive responses**

- Many candidates gave at least a partially correct declaration statement for the specified pointer data type.
- Some of these candidates gave a completely correct declaration statement.

Limited responses

- Some candidates missed out the second part of the statement: `=^STRING`.
- Some candidates used the wrong command word, `DECLARE` instead of `TYPE` in their pseudocode statement.

Question 2

(a) **Comprehensive responses**

- Some candidates stated appropriate uses of random file organisation including booking systems or encryption utilities.
- Some candidates gave examples based on database management systems, as this was stated in the question.

Limited responses

- Most candidates did not give enough detail in their examples of uses of random file organisation, stating what might be stored rather than the actual use.

(b) **Comprehensive responses**

- Many candidates were aware that records in random files are stored in no particular order.
- Many candidates were aware that a record key is used to determine where a record is to be stored, some of these candidates also related their answer to the use of a hash function.

Limited responses

- Some candidates gave answers that were too vague and did not use the correct terminology, such as 'key word' instead of 'record key'.
- Some candidates incorrectly used the word 'file' instead of 'record', for example 'files are stored in an empty location'.

(c) **Comprehensive responses**

- Many candidates gave clear, detailed descriptions of the process of direct file access.

Limited responses

- Some candidates incorrectly described processes such as a binary search as a process to find a record in a random file.
- Some candidates did not give enough detail in their answer to cover all of the requirements.

Question 3

(a) **Comprehensive responses**

- Most candidates gave the correct answer accompanied by sufficient working to show how they had converted the given denary number to binary floating-point representation.

Limited responses

- Some candidates achieved the correct answer without showing sufficient detail in their working as to how they arrived at their answer.
- Some candidates gave an incorrect final answer with some correct working shown.

(b) **Comprehensive responses**

- Most candidates gave the correct answer and working to show how they had converted the given binary floating-point number to denary.

Limited responses

- Some candidates achieved the correct answer without showing sufficient detail in their working.
- Some candidates missed out the fact that the number was negative and forgot to use a minus sign in their final answer.

Question 4

Comprehensive responses

- Most candidates were able to match some of the stated TCP/IP layers with the most appropriate description.
- Some of these candidates correctly matched all the given TCP/IP layers.

Limited responses

- Some candidates incorrectly matched the Internet layer with the description that begins with the word 'interconnects'.

Question 5

Comprehensive responses

- The majority of candidates gave some correct benefits and/or limitations of a virtual machine.
- Many of these candidates gave **two** benefits and **two** limitations as required by the question.

Limited responses

- Some candidates gave repeated answers, for example, giving two versions of the ability to run multiple operating systems or multiple applications on the same computer, with the result that it only counted as one answer.
- Some candidates gave answers that were not specific enough to match any of the required answers.

Question 6

(a) Comprehensive responses

- Most candidates correctly completed some of the truth table based on the given logic circuit.
- Many of these candidates completed the truth table completely correctly.

Limited responses

- Some candidates made errors in their working columns which fed through and gave incorrect final output results.

(b) Comprehensive responses

- Some candidates correctly stated that a K-map is an easier way to solve logic problems because you do not need to use Boolean algebra.

Limited responses

- Most candidates did not give enough detail in their description of a benefit of using a K-map to solve logic problems.

(c) (i) Comprehensive responses

- Many candidates drew the correct loops around appropriate groups of 1s in their K-map.

Limited responses

- Some candidates did not realise that the group of four 1s in the centre of the K-map, arranged as 2×2 , was also a group.
- Some candidates incorrectly divided the K-map into groups avoiding any overlapping groups.

(ii) Comprehensive responses

- Most candidates wrote a correct Boolean expression from their K-map as a sum-of-products.

Limited responses

- Some candidates wrote incorrect Boolean expressions that did not represent a sum-of-products taken from their K-map.

Question 7

(a) Comprehensive responses

- Many candidates correctly wrote the given infix expression in RPN.

Limited responses

- Some candidates incorrectly changed the order of the numbers given in the infix expression in their RPN expression.
- Some candidates incorrectly wrote too many numbers in their RPN expression before adding their first operator.

(b) Comprehensive responses

- Most candidates correctly wrote the given RPN expression in infix form.

Limited responses

- Some candidates did not include all the required brackets in their answer to give the correct final infix expression.

(c) Comprehensive responses

- Most candidates correctly completed the diagram to show the changing contents of the stack as the given RPN expression was being evaluated.

Limited responses

- Some candidates made errors in their calculations as they worked through the question, giving partially incorrect sections.
- Some candidates incorrectly included operators within the stack.

Question 8

(a) (i) Comprehensive responses

- The majority of candidates were aware that cipher text is encrypted text.

Limited responses

- Some candidates did not give a specific enough statement to be sure that they meant cipher text was encrypted text.

(ii) Comprehensive responses

- The majority of candidates were able to define a public key.

Limited responses

- Some candidates did not give enough detail in their answer, for example, if they said it was the key anyone can use to send a message they would need to at least also state that it would be used to encrypt a message before sending it.

(b) Comprehensive responses

- Most candidates gave some or all of the steps involved in sending a message to an organisation using asymmetric encryption.

Limited responses

- Some candidates thought that an individual would have access to an organisation's private key when sending a private message.
- Some candidates thought that the individual would use their public key to encrypt the message and the organisation would use the individual's private key to decrypt it.

Question 9

(a) Comprehensive responses

- Most candidates were aware that supervised learning uses labelled data when training and unsupervised learning uses unlabelled data.
- Some of these candidates were also able to expand one or both of these points.

Limited responses

- Some candidates gave answers that were not specific enough and did not use the appropriate technical language to meet the criteria, such as the system being given a large data set without saying whether it is labelled or unlabelled.
- Some candidates did not add any more depth to their answer than to say that, for example, supervised learning is trained using labelled data.

(b) Comprehensive responses

- Most candidates were able to find the shortest path to all the nodes on the diagram and demonstrate the use of Dijkstra's algorithm to do this.

Limited responses

- Some candidates did not find the shortest path to all of the nodes, only to some of them.
- Some candidates did not provide sufficient working to show that they had necessarily tried different paths to get to each node.

Question 10

(a) Comprehensive responses

- Many candidates wrote one or more correct lines of pseudocode for an insertion sort algorithm to match a given specification.
- Some of these candidates correctly completed the whole algorithm.

Limited responses

- Some candidates incorrectly wrote pseudocode for a bubble sort algorithm.

(b) Comprehensive responses

- The majority of candidates described appropriate factors that would change the time efficiency of an insertion sort algorithm.
- Many of these candidates identified the factors and went on to give a good expansion point.

Limited responses

- Some candidates only gave a limited description, such as stating a single factor that would affect the time efficiency of an insertion sort algorithm.

(c) Comprehensive responses

- Many candidates gave the correct Big O notation to represent the worst-case time efficiency of the insertion sort.

Limited responses

- Some candidates gave the Big O notation for other types of algorithms and not the insertion sort.

Question 11

(a) Comprehensive responses

- The majority of candidates were able to write some or all of the additional facts to represent the new information supplied for the given declarative language application.

Limited responses

- Some candidates incorrectly used uppercase text that should have been lowercase.
- Some candidates missed off the important full stop that should go at the end of each fact.

(b) Comprehensive responses

- The majority of candidates gave the correct result of the given declarative language goal.

Limited responses

- Some candidates missed out one or more of the names that would be listed for the given declarative language goal.

(c)

Comprehensive responses

- Most candidates wrote the correct declarative language rule for the given scenario either completely correctly or with only minor errors.

Limited responses

- Some candidates used rules that had not previously been defined in the question, for example 'wants' instead of 'owns'.
- Some candidates did not join the different parts of their rule together using appropriate logic conditions.

COMPUTER SCIENCE

Paper 9618/32 Advanced Theory
--

Key messages

What candidates did well

- Most candidates demonstrated good knowledge related to user-defined data types and were able to write pseudocode to define them.
- Most candidates were able to convert a binary floating-point number to its denary equivalent.
- Many candidates were able to convert Reverse Polish Notation (RPN) expressions to infix form and vice versa.
- Most candidates were able to complete a Karnaugh map (K-map) from a given truth table and use it to simplify a Boolean expression.
- Most candidates were able to demonstrate the use of Dijkstra's algorithm and find the shortest path from a home node to each of the nodes in a given diagram.

What candidates needed to improve

- Some candidates did not always answer the questions that were asked, such as instead of explaining the meaning of the term virtual machine, they gave benefits and drawbacks, or instead of outlining a process to send a verified message to a group of people using encryption keys, described the use of digital signatures.
- Some candidates did not apply pseudocode in the manner recommended in the pseudocode guide available to support this syllabus and instead used elements of program code.
- Some candidates were unable to define an Object-Oriented Programming (OOP) class and constructor as shown in the pseudocode guide available to support this syllabus.
- Some candidates inappropriately used words from the question as a means to answer the question. For example, to describe the meaning of inheritance in OOP, they might say 'it inherits methods and data from a parent class.'

Comments on specific questions

Question 1

(a) Comprehensive responses

- Most candidates understood that the correct pseudocode statement began with `TYPE` `Animals`.

Limited responses

- Some candidates used a colon instead of an equals sign to join the two parts of the statement together. Other candidates incorrectly enclosed the individual data items in quotation marks.

(b)(i) Comprehensive responses

- Many candidates recognised that they had created an enumerated data type in part 1 (a).

Limited responses

- Some candidates incorrectly thought the data type created in part 1 (a) was a record.

(ii) Comprehensive responses

- Many candidates thought the list used for the data type in part 1 (a) was ordinal, that the elements were unique, or that the list was complete.

Limited responses

- Some candidates inappropriately gave answers referring to data types, such as all the elements were of the same data type.

Question 2

Comprehensive responses

- Most candidates completed the paragraph with all, or nearly all, the correct missing words.

Limited responses

- Some candidates thought the final word should be controls rather than interconnects, to give 'this is the layer that controls nodes in the network'.

Question 3

(a) Comprehensive responses

- Most candidates gave the correct answer accompanied by sufficient working to show how they had solved the problem.

Limited responses

- Some candidates made errors in their calculations which led to an incorrect decimal value.
- Some candidates did not correctly apply the negative value of the most significant bit.

(b) Comprehensive responses

- Most candidates understood that an underflow meant that a value was too small to be represented and therefore becomes zero.

Limited responses

- Some candidates did not give a reason for the underflow to occur, such as a calculation being performed on a normalised floating-point number.

Question 4

(a) Comprehensive responses

- Many candidates gave good examples of where sequential file organisation could be used and specified how the data in the file would be ordered.

Limited responses

- Some candidates gave very general examples which were not specific enough to know that sequential file organisation was being used, or did not specify how the file was ordered, so they could have been writing about serial file organisation.

(b) Comprehensive responses

- Most candidates gave good answers to describe sequential file organisation including that 'records are stored one after the other in order of a key field'.

Limited responses

- Some candidates incorrectly used the word 'files' in place of 'records', for example, to say that 'files are stored one after the other, in order of their key field'.

(c) Comprehensive responses

- Most candidates gave fully correct or partially correct answers, including statements such as 'each record is searched linearly using a record key until the correct record is found'.

Limited responses

- Some candidates incorrectly used the word 'file' in place of 'record', as with part 4 (b) to give answers in which each file is searched one after the other.
- Some candidates gave descriptions of searching an index file as for index sequential files, which was not the focus of this question.

Question 5

Comprehensive responses

- Most candidates gave some good points to explain the meaning of a virtual machine.

Limited responses

- Some candidates gave answers that could be considered benefits and drawbacks of virtual machines, which was not the focus of this question.

Question 6

(a) Comprehensive responses

- Many candidates correctly wrote the given infix expression in RPN.

Limited responses

- Some candidates incorrectly began their expression with a number other than 16, which was the first number in the infix expression and should therefore have remained unchanged.
- Some candidates did not follow the principles of RPN by correctly writing the numbers followed by the operators, working through the expression from left to right.

(b) Comprehensive responses

- Most candidates wrote the numbers and operators for the given RPN expression in the correct order with many of these also completing the task with the correct brackets.

Limited responses

- Some candidates did not include sufficient brackets in their answer to give the correct infix expression, or placed their brackets in the wrong place.

(c) Comprehensive responses

- Most candidates correctly completed the diagram to show the changing contents of the stack as the given RPN expression was being evaluated.

Limited responses

- Some candidates made errors in their calculations as they worked through the question, giving partially incorrect sections.
- Some candidates incorrectly included operators within the stack.

Question 7

(a) (i) Comprehensive responses

- The majority of candidates correctly populated the K-map with 0s and 1s.

Limited responses

- Some candidates put one or more 1s or 0s in the wrong place.
- Some candidates only populated their K-map with 1s and left out the 0s.

(ii) Comprehensive responses

- Many candidates drew the correct loops around appropriate groups in their K-map.

Limited responses

- Some candidates did not realise that the central group was a single group of eight 1s and incorrectly divided it into two groups of four 1s.

(iii) Comprehensive responses

- Many candidates wrote a correct Boolean expression from their K-map as a sum-of-products.

Limited responses

- Some candidates wrote a partially correct sum-of-products to represent the loops on their K-map.

- Some candidates wrote a Boolean expression that did not match their K-map or was an attempted simplification of the expression representing the original truth table, and not a sum-of-products.

(b) Comprehensive responses

- Many candidates correctly demonstrated the use of De Morgan's laws and Boolean algebra to simplify a given Boolean expression.
- Some of these candidates also achieved the expected final answer.

Limited responses

- Some candidates demonstrated a limited understanding of how to use the various laws of Boolean algebra to simplify parts of the given Boolean expression.

Question 8

(a) Comprehensive responses

- Most candidates understood that asymmetric encryption uses two keys and symmetric encryption uses a single key.
- Many candidates were aware that asymmetric encryption uses one key to encrypt the data and a different key to decrypt the data, whereas symmetric encryption uses the same key for both.
- A number of candidates were aware that asymmetric encryption required longer keys than symmetric encryption to maintain the same level of security.

Limited responses

- Some candidates were aware of some of the facts noted above, and only gave one side of the argument. This question asked for the difference between symmetric and asymmetric encryption to be described, so the effect of a given feature had to be stated for both encryption types.

(b) Comprehensive responses

- Most candidates were aware that a decryption key would be made available to the group receiving the message, with the majority of these candidates realising that it was the business's public key.
- Many candidates were aware that the message would be encrypted by one key and then decrypted by its matching key, with the majority of these candidates realising that the business's private key was used to encrypt the message and the business's public key was used to decrypt it.
- Some candidates also realised that the sender of the message would be verified because the public key they received from them successfully decrypted the message that was encrypted with their private key.

Limited responses

- Some candidates confused this question with a question about digital signatures and incorrectly discussed the use of a message digest and a digital signature in their responses.

Question 9

(a) Comprehensive responses

- The majority of candidates correctly named Deep Learning as the subset of Artificial Intelligence (AI) that makes use of an artificial neural network.

Limited responses

- A significant number of candidates incorrectly named Machine Learning instead of Deep Learning.

(b)(i) Comprehensive responses

- The majority of candidates were aware that the A* algorithm made use of heuristics when finding the shortest path between two points.
- Some candidates were also aware that Dijkstra's algorithm finds the shortest path to all nodes.

Limited responses

- Many candidates incorrectly stated that Dijkstra's algorithm searches all possible routes or paths to find the shortest path, rather than recognising that Dijkstra's algorithm finds the shortest path to all nodes.

(ii) Comprehensive responses

- Most candidates were able to find the shortest path to all the nodes on the diagram and demonstrate the use of Dijkstra's algorithm to do this.

Limited responses

- Some candidates did not find the shortest path to all of the nodes, only to some of them.
- Some candidates did not provide sufficient working to show that they had necessarily tried different paths to get to each node.

Question 10

- (a)** Due to a series-specific issue with **Question 10(a)**, full marks were awarded to all candidates for this question to make sure that no candidates were disadvantaged.

(b) Comprehensive responses

- Most candidates wrote some correct pseudocode statements to fill in the missing parts of the given function.
- Some of these candidates filled in all of the missing lines or most of the missing lines correctly.

Limited responses

- Some candidates used incorrect logic when completing their function and so their pseudocode was not correct.

Question 11

(a) Comprehensive responses

- Some candidates wrote appropriate pseudocode statements to declare an OOP class based on the information given in the question.
- Some of these candidates completed the diagram correctly or nearly correctly.

Limited responses

- Some candidates didn't make correct use of all the information given in the question to complete their OOP class, such as using the attributes named in the question in their constructor rather than in the attribute declaration section before the constructor.
- Some candidates did not follow the correct order for the different sections, such as placing the constructor before the attribute declaration section.

(b) Comprehensive responses

- Most candidates were able to complete correctly some of the missing OOP terms or descriptions in the given table.
- Some of these candidates correctly completed all of the missing OOP terms and descriptions.

Limited responses

- Some candidates used words given in the question to describe their answer, such as using the word inherit to describe the meaning of inheritance in the context of an OOP term.

COMPUTER SCIENCE

Paper 9618/33 Advanced Theory
--

Key messages

What candidates did well

- Most candidates demonstrated good knowledge of user-defined data types and were able to write pseudocode to define a non-composite data type.
- Most candidates were able to convert a binary floating-point number to its denary equivalent.
- Many candidates were able to convert an infix expression to Reverse Polish Notation (RPN).
- Most candidates were able to simplify a Boolean expression using a Karnaugh map (K-map).
- Most candidates were able to demonstrate the use of the A* algorithm and find the best path between the W and E nodes on the given diagram.

What candidates needed to improve

- Some candidates did not always answer the questions that were asked. For example, instead of explaining regression methods in relation to machine learning, they explained another aspect of machine learning.
- Some candidates did not provide sufficient detail, or use sufficient subject-appropriate language, in their answers for explanation or description questions.
- Some candidates did not apply pseudocode in the manner recommended in the pseudocode guide available to support this syllabus and instead used aspects of program code.

Comments on specific questions

Question 1

(a) Comprehensive responses

- Many candidates were aware that user-defined data types were needed to meet the specific needs or requirements of the programmer.
- Many candidates understood that extending the data types already available could make the code easier to read.

Limited responses

- Some candidates did not offer sufficient discrete points in their response.

(b) Comprehensive responses

- Many candidates correctly declared a non-composite data type named `Fibonacci` with the numbers stated in the question.

Limited responses

- Some candidates did not separate the two parts of the pseudocode, the identifier and the list of elements with an equals sign.
- Some candidates used the wrong command word in their pseudocode, `DECLARE` instead of `TYPE`.

Question 2

Comprehensive responses

- The majority of candidates completed the paragraph with all the correct missing words.

Limited responses

- Some candidates made errors with one or two of the missing words.

Question 3

(a) Comprehensive responses

- Most candidates stated an appropriate use of serial file organisation, mostly involving an example where data is stored chronologically.

Limited responses

- Some candidates did not give enough detail in their response to demonstrate an understanding of which data was actually being stored.

(b) Comprehensive responses

- Most candidates gave good answers to describe the organisation of serial files so it was clear they understood that records are stored one after the other in chronological order.

Limited responses

- Some candidates gave answers that only stated one point, such as 'records are stored in chronological order', without any further expansion.
- Some candidates incorrectly used the word 'file' instead of 'record', for example 'files are stored in chronological order'.

(c) Comprehensive responses

- Many candidates gave clear responses outlining how a record is found in a serial file, for example, 'sequential access is used where each record is searched one after the other, until the required record is found'.

Limited responses

- Some candidates incorrectly used the word 'file' in place of 'record', as with part 3 (b) to give answers such as 'each file is searched one after the other'.
- Some candidates described stages relevant to other types of file access, for example, using record keys or indexes.

Question 4

(a) Comprehensive responses

- Many candidates were aware that massively parallel computers consist of many processors linked together to work simultaneously on the same problem.

Limited responses

- Some candidates did not provide sufficient distinct points in their description.
- Some candidates did not give enough detail in their description to convey the scale of massively parallel computers, such as saying 'they have several processors' whereas 'they have an extremely large number of processors' is more appropriate.

(b) Comprehensive responses

- Many candidates were aware that one difference between the SIMD and MISD computer architectures is that SIMD systems all work on the same instruction while MISD systems perform different instructions.
- Many candidates were aware that one similarity between SIMD and MISD architectures is that they are both parallel systems.

Limited responses

- Some candidates simply stated the meaning of the acronyms SIMD and MISD as a difference, which does not demonstrate an understanding of these computer architectures.
- Some candidates stated that both systems use multiple processors, as a similarity, which is not true.

Question 5

(a) Comprehensive responses

- Most candidates gave the correct answer accompanied by sufficient working to show how they had converted the given denary number to binary floating-point representation.

Limited responses

- Some candidates gave the correct answer without showing sufficient detail in their working as to how they arrived at their answer.
- Some candidates gave an incorrect mantissa with a correct exponent, or vice versa.

(b) Comprehensive responses

- Many candidates gave a good explanation of how the two numbers given in the question, when added together, would cause an overflow.

Limited responses

- Some candidates did not give sufficient distinct points in their answer.
- Some candidates recognised that overflow would occur without giving any detail as to how this would happen.

Question 6

(a) (i) Comprehensive responses

- Many candidates drew the correct loops around appropriate groups of 1s in their K-map.

Limited responses

- Some candidates did not realise that the top and bottom groups of two 1s actually wrapped around to form a group of four 1s.
- Some candidates divided the K-map into excessive groups, typically in groups of two 1s.

(ii) Comprehensive responses

- Most candidates wrote a correct Boolean expression from their K-map as a sum-of-products.

Limited responses

- Some candidates wrote a partially correct sum-of-products to represent the loops on their K-map.
- Some candidates wrote a Boolean expression that represented an unsimplified version of the Boolean expression that would have been used to form the K-map.

(iii) Comprehensive responses

- Many candidates wrote a correct simplified Boolean expression from their K-map sum-of-products.

Limited responses

- Some candidates wrote a simplified Boolean expression that was not in its simplest form.
- Some candidates wrote a simplified Boolean expression that did not match their sum-of-products from part 6 (a)(ii).

(b) Comprehensive responses

- Many candidates correctly demonstrated the use of De Morgan's laws and Boolean algebra to simplify a given Boolean expression.
- Many of these candidates also achieved the expected final answer.

Limited responses

- Some candidates demonstrated a limited understanding to use the various laws of Boolean algebra to simplify parts of the given Boolean expression.

Question 7

(a) Comprehensive responses

- Most candidates were able to outline the steps involved to evaluate an RPN expression when using a stack.

Limited responses

- Some candidates gave responses that were too vague and did not clearly identify the steps involved when an RPN expression is evaluated using a stack.

(b) Comprehensive responses

- Many candidates correctly wrote the given infix expression in RPN.

Limited responses

- Some candidates incorrectly placed the minus sign after the 3 instead of before it.

(c) Comprehensive responses

- Most candidates correctly completed the diagram to show the changing contents of the stack as the given RPN expression was being evaluated.

Limited responses

- Some candidates made errors in their calculations as they worked through the question, giving partially incorrect sections.
- Some candidates incorrectly included operators within the stack.

Question 8

(a) Comprehensive responses

- Most candidates understood that quantum cryptography provides a more secure method of encryption and gave answers that related to this.
- Some of these candidates also included the fact that quantum cryptography uses the laws of quantum physics rather than mathematical formulae to generate keys.

Limited responses

- Some candidates only gave one point in their answer and did not expand on it.

(b) Comprehensive responses

- Most candidates were able to give benefits and drawbacks of quantum cryptography with benefits often related to its improved security features and ability to withstand hacking attempts, and drawbacks related to cost and availability issues.

Limited responses

- Some candidates did not give enough detail in some or all of their benefits and/or drawbacks.
- Some candidates gave benefits and drawbacks that were too similar to each other to be considered distinct.

Question 9

(a) Comprehensive responses

- Many candidates were aware that regression methods when applied to machine learning are a supervised learning technique.
- Many candidates were aware that regression methods are used to find lines of best-fit when applied to machine learning.

Limited responses

- Some candidates described other aspects of machine learning rather than regression methods, such as back propagation of errors.

(b) Comprehensive responses

- Most candidates were able to find the best path between the W and E nodes on the diagram and demonstrate the use of the A* algorithm to do this.

Limited responses

- Some candidates did not complete the table correctly to show their working and how they achieved their final path.

Question 10

(a) (i) Comprehensive responses

- Many candidates correctly completed some of the table, naming the data types and describing the identifiers listed, in the context of the algorithm that would follow.

Limited responses

- Some candidates inappropriately used the words in the question to describe the purpose of the identifier, for example `Left` is the left boundary of the searching range.
- Some candidates did not account for the fact that the boundaries would change during the course of the binary search and related their answers to the overall array, for example, `Left` is the lowest boundary of the array.

(ii) Comprehensive responses

- Most candidates wrote some correct pseudocode statements to complete the function `PartSearch`.
- Some of these candidates completed the whole function correctly.

Limited responses

- Some candidates used incorrect logic when completing their function and so their pseudocode was not correct.

(b) Comprehensive responses

- Most candidates gave the correct Big O notation representing the time efficiency of a binary search.
- Many of these candidates were also able to describe how the time efficiency of a binary search changes as the number of items in the search list increases.

Limited responses

- Some candidates did not give enough detail in their description of how the time efficiency of a binary search will change as the number of items in the search list increases.

Question 11

(a) Comprehensive responses

- The majority of candidates wrote correct assembly language code to meet the requirements stated in the question.

Limited responses

- Some candidates added incorrect details to their assembly language code, such as for operands: quotation marks around identifiers or angle brackets around numbers. Also, labels were sometimes written in inappropriate places.

(b) Comprehensive responses

- Most candidates gave the correct final value in the values table.
- Many of these candidates also gave correct values for the two labelled memory locations.

Limited responses

- Some candidates had added or subtracted the wrong values together to give incorrect values in the labelled memory locations.

COMPUTER SCIENCE

Paper 9618/41 Practical
--

Key messages

What candidates did well

- Candidates produced clear evidence of their code for each question in the correct space in the evidence document.
- Many candidates had a good understanding of object-oriented programming and could create classes and constructors.

What candidates needed to improve

- Candidates often missed the need to store or make use of a return value from a function call.
- Some candidates made use of global variables throughout the questions instead of passing data and arrays as parameters.
- Candidates need to consider the purpose of the head and tail pointers in a queue for the given questions, for example, whether they point to the current item or the next space and how this affects their use and manipulation within the questions.

There was an error in the first bullet point on **Question 2(c)(i)** where the wording was considered to be unclear. This has been corrected in the published version of the paper.

Comments on specific questions

Question 1

(a) Comprehensive responses

- Created a 1D array and initialised 10 elements to -1 . They also initialised both pointers to -1 .

Limited responses

- Created an array without initialising its elements to -1 .

(b) Comprehensive responses

- Created a function that checked the queue was full using the tail pointer and returned false.
- Stored the parameter in the position of the tail pointer $+1$ and returned true in an appropriate place.
- Identified the need to deal with the queue being empty and changing the head pointer to be 0.

Limited responses

- Created a function header without a parameter.
- Stored the value in the tail pointer position instead of the tail pointer $+1$.
- Incremented the tail pointer after returning a value, meaning that code will not run.
- Did not check for the condition of the queue being empty and that the head pointer will still be at -1 .

(c) (i) Comprehensive responses

- Took 11 values as input within a loop and called `Enqueue` with each input.
- Stored and checked the value of the returned value and output a suitable message if this was false.
- Output the contents of the array.

Limited responses

- Took 10 (or fewer) items as input.
- Did not store the return value from `Enqueue()` or called `Enqueue()` before a selection statement and then again within a selection statement with the same data.

(ii) Comprehensive responses

- Produced a screenshot that showed all of the inputs of the correct values, a queue full message and the output of the array.

Limited responses

- Showed the outputs of “Queue full” and the array, they did not include the inputs of the numbers.

(d) Comprehensive responses

- Checked all conditions that could indicate that the queue is empty, including both no data ever being put in the queue, as well as data being enqueued and dequeued.
- Accessed and returned the data stored in the head pointer position.
- Included all of the functionality before a value was returned in the function.

Limited responses

- Checked if the queue was initially empty, they did not consider if data had been inserted into the queue and removed from the queue, where the pointers would not equal `-1`.
- Accessed the data at head pointer `+1` and returned this value instead of the data at the head pointer.
- Incremented the head pointer after already returning a value, therefore the code would not run.

(e) (i) Comprehensive responses

- Read 6 inputs from the user and called `Enqueue` with each value.
- Called `Dequeue` 4 times and checked the value returned each time to output is.

Limited responses

- Took 10 integers as input from the previous question part instead of amending to 6.
- Output the value of the queue using the array and pointer instead of the return value.

(ii) Comprehensive responses

- Produced an output showing the 6 numbers input and the correct output.

Limited responses

- Showed the output and did not include the input of the numbers.

Question 2

(a) Comprehensive responses

- Created a record structure, or a class, for the toy record.
- Included a constructor if they created a class to assign the four fields their initial values.

Limited responses

- Created an array or individual variables without making use of a record structure.

(b) (i) Comprehensive responses

- Opened and closed the appropriate file in a suitable place, for example, after all code was read and inside the exception's try clause.
- Included all file access within one (or multiple) try clauses, including a suitable output of an error.
- Read in each line from the file and used an appropriate method to split each line by the commas.
- Created a new record, or object, using the data that was read and split and stored each of these records in an array that was then returned by the function.

Limited responses

- Opened the file and they did not close the file.
- Did not include exception handling, or opened the file within the try clause, they then tried to close the file outside of the exception.
- Read in the incorrect number of lines, for example too few.
- Assigned the split data to variables instead of creating a record or object.

(ii) Comprehensive responses

- Created a procedure that iterated through each element in the array parameter and output the three values.

Limited responses

- Output the data for one element or did not use a parameter and tried to access a global variable.

(iii) Comprehensive responses

- Called `ReadData()` and made use of the returned array by passing it as an argument to `OutputToys`.

Limited responses

- Called the appropriate subroutines and did not make use of the return value from `ReadData()`.

(iv) Comprehensive responses

- Produced a screenshot showing all of the data read from the file.

Limited responses

- Produced a screenshot showing some of the data, often with the last element missing or some of the data values incomplete.

(c) (i) Comprehensive responses

- Created a function that returned the populated array.
- Prompted and read in the difficulty level until it was one of the three options, some responses took the words 'easy', 'medium' and 'hard', whilst some responses asked the user to enter 1, 2 or 3 which made validation easier.
- Iterated through each element in the array parameter and compared the difficulty level, assigning the record (or object) to the new array if it was a match.

Limited responses

- Took the difficulty as input without validation, or inaccurate validation such as using the wrong `Boolean` comparison in the condition.
- Used a global array instead of the parameter array.
- Assigned elements to the new array, they did not return this array.

Due to an issue with this question, its treatment in marking was considered carefully to make sure that no candidates were disadvantaged.

(ii) Comprehensive responses

- Opened the file for writing and closed the file in an appropriate place, making use of exception handling for file access.
- Formatted the data accurately for each record (or object) in the array parameter and wrote each line to the file.

Limited responses

- Opened the file to write, they did not close the file.
- Formatted the data inaccurately, for example, including a space instead of a comma.

(iii) Comprehensive responses

- Called `SearchToys()` and stored or used the array that was returned by the function.
- Called `WriteToys()` with the returned array.

Limited responses

- Called both subroutines and did not store or use the returned array from `SearchToys()`.

(iv) Comprehensive responses

- Produced a screenshot showing the contents of the text file with the filename.

Limited responses

- Produced a screenshot of the data, but without showing that it was stored in a file and without the filename visible.

Question 3

(a) (i) Comprehensive responses

- Created a class containing the attribute declarations and the constructor.
- Assigned both parameters to the attributes within the constructor.

Limited responses

- Produced evidence of the constructor without evidence of the class creation.
- Assigned the data to the parameters and did not show the data types of the attributes.

(ii) Comprehensive responses

- Created a class and constructor within the class.
- Created an array attribute within the constructor and initialised all 100 elements to an empty object using the correct values of "" and -1.
- Created the get method, with no parameter, that returned the correct attribute, all within the class definition.

Limited responses

- Created the array attribute without initialisation, or initialised it to the data "" or -1 without instantiating an object.
- Did not include the get method within the response.

(iii) Comprehensive responses

- Created a method that took the array parameter and checked the number of elements.
- Compared the length to the attribute `NumberSpaces` and returned `False` if greater; otherwise stored each value in the parameter in the array attribute.

Limited responses

- Created a function that was not a method, most notably in Python where `self` (or equivalent) was missing.
- Compared the length of the array and assigned one of the values to `BoardArray` instead of all of them.

(iv) Comprehensive responses

- Created a method header and iterated through each object within the array `BoardArray`.
- Created a string containing all objects that were not null in the correct format.

Limited responses

- Created a function that was not a method, for example in Python missing the `self` (or equivalent).
- Output each of the data from each object instead of returning a final string.
- Returned a string of the first object type and amount, without concatenating each object from the array.
- Used an inappropriate concatenation operator, for example a comma in Python, to create a tuple instead of a string.

(v) Comprehensive responses

- Created a get method that took the array parameter and returned the element at that index.

Limited responses

- Created a function without it being a method, for example missing `self` (or equivalent) in Python.

(b)(i) Comprehensive responses

- Created 5 objects with the appropriate values. Each object was stored in the correct array.
- Created a new instance of the class `Path` with the correct value of 5.
- Called `InitialiseData()` with the correct argument.
- Called `ReturnPath()` and output the return value.

Limited responses

- Stored the data in variables instead of as an object, or created a null object and then tried to assign the values.
- Created an instance of `Path` without the argument of 5.
- Called `ReturnPath()` and did not output the return value.

(ii) Comprehensive responses

- Produced a screenshot showing the data in the correct format.

Limited responses

- Did not include a screenshot.
- Produced a screenshot with the data in an incorrect format.

COMPUTER SCIENCE

Paper 9618/42 Practical
--

Key messages

What candidates did well

- Candidates produced clear evidence of their code for each question in the correct space in the evidence document.
- Many candidates had a good understanding of object-oriented programming and could create classes and constructors.

What candidates needed to improve

- Candidates often missed the need to store or make use of a return value from a function call.
- Some candidates made use of global variables throughout the questions instead of passing data and arrays as parameters.
- The creation, initialisation and manipulation of 2D arrays. Some candidates created tuples within a 1D array, whilst some candidates chose to create a class instead of using the required 2D arrays.

There was an error in **Question 2(a)(iii)** where there was a spelling error of the word `MeetCriteria`. This has been corrected in the published version of the paper.

Comments on specific questions

Question 1

(a) Comprehensive responses

- Created a 1D array and stored the correct string data in the order given within the array.

Limited responses

- Created an array and combined multiple words into one string that meant it had the incorrect number of elements.
- Stored the string data without the appropriate "" to define each as a string.
- Stored inaccurate elements in the array, for example, missing one word or with misspellings.
- Took the strings as input from the user instead of hard coding them into the array.

(b)(i) Comprehensive responses

- Looped through each element in the array and used a method to calculate the number of characters in that string.
- Stored the longest and shortest strings in a variable that was used in each comparison within the loop.
- Wrote an algorithm that would return the longest and shortest string no matter which position in the array these were located.
- Output a message containing the longest and shortest string with appropriate additional text.

Limited responses

- Output the shortest and longest string without additional text.
- Produced an algorithm that would not detect if the first element was the longest or shortest string.
- Compared each element with the next element, instead of with the current longest element.
- Identified and output the number of characters in the longest and shortest strings, not the strings themselves.

(ii) Comprehensive responses

- Called the function with the array as an argument.

Limited responses

- Called the function without the array as the parameter.

(iii) Comprehensive responses

- Produced a screenshot that showed the longest and shortest string.

Limited responses

- Produced a screenshot that showed an error in the program.
- Produced a screenshot that output the longest and shortest words without appropriate text.
- Produced a screenshot that output words other than the longest and shortest words.

(c) (i) Comprehensive responses

- Created an insertion sort algorithm that contained two loops and inserted each element into the correct position within the sorted list.
- Sorted the data in-place and returned the sorted array.
- Calculated the length of the array and used this as the stopping condition in the outer loop.

Limited responses

- Created a bubble sort algorithm.
- Sorted the data based on the length and not the content.
- Sorted the array and did not return the sorted array.
- Used a global array instead of taking one as a parameter.

(ii) Comprehensive responses

- Stored the return value from the call to `InsertionSort()` and used this returned array to output the sorted contents.
- Output the required messages before each array output.

Limited responses

- Output a global array and used this instead of passing one to `InsertionSort()`.

(iii) Comprehensive responses

- Produced an output showing the required messages and the data before and after sorting.

Limited responses

- Produced a screenshot showing only the sorted list.
- Produced a screenshot showing an incorrectly sorted list.

Question 2

(a) (i) Comprehensive responses

- Created a class containing a constructor.
- Declared the four attributes with appropriate data types.
- Assigned a parameter to each attribute.

Limited responses

- Created a constructor that was not within a class, for example in Python where the indents were incorrect.
- Did not identify the data types for each attribute.
- Assigned values to the attributes instead of the parameters.

(ii) Comprehensive responses

- Created two get methods without parameters, which returned the correct attributes without any additional changes to these values.

Limited responses

- Took a parameter and attempted to assign this to the attribute or return this value.

- Created a function instead of a method, for example in Python not including `self` (or equivalent).

(iii) Comprehensive responses

- Created a method that took two parameters and compared the parameters to the attributes, returning Boolean `TRUE` if both matched and `FALSE` otherwise.

Limited responses

- Compared one attribute, for example `Hair`, and not the second attribute.
- Took data as input and tried to compare the inputs with the attributes.
- Compared the data to variables and not attributes.
- Returned a string value instead of a Boolean.
- Created a selection statement that did not have a result for all conditions, therefore in some cases nothing would be returned by the function.

Due to an issue with this question, its treatment in marking was considered carefully to make sure that no candidates were disadvantaged.

(b)(i) Comprehensive responses

- Opened the file and closed the file in appropriate places.
- Used exception handling for all file access and included an appropriate output message in case of error.
- Read in each line from the file and split it by a space.
- Created a new instance of the class `Avatar` with the split data and stored these in the array.

Limited responses

- Opened the file without closing the file or closed the file in an inappropriate place, for example within the loop where each line was read.
- Stored the data read within an array named `Avatar` instead of instantiating an object.

(ii) Comprehensive responses

- Called `LoadAvatars()`.

(c)(i) Comprehensive responses

- Created a procedure that output a suitable message to prompt the user to input the two values, then read these as input.
- Continued to ask for the inputs until both were valid.
- Iterated through each object in the array and used the inputs as arguments in the call to `MeetCriteria()`.
- Used the return value from the method call and stored the object in a different area when `True`.

Limited responses

- Took the inputs without validation or had inaccurate validation that missed a criteria or made use of the wrong Boolean operators.
- Accessed only one object from the array instead of each object in turn.
- Compared the inputs directly with the attributes instead of using the method `MeetCriteria()`.

(ii) Comprehensive responses

- Called `EnterChoices()` and output the length of the array or made use of a counter variable from the previous question part.
- Iterated through each object in the array and output the name and age using the get methods along with appropriate text.

Limited responses

- Iterated through an incorrect number of array elements.
- Iterated through the original array and not those that match the criteria.
- Output the name and age without appropriate text and using the attributes directly instead of the get methods.

(d)(i) Comprehensive responses

- Produced a screenshot showing the two required inputs, the 4 matching and the data for each avatar.

Limited responses

- Showed the output without the inputs from the user.
- Had the input of the incorrect hair colour as a separate screenshot instead of appropriate validation.

(ii) Comprehensive responses

- Produced a screenshot showing the required inputs and the output of 0 avatars.

Limited responses

- Produced a screenshot of the output without the required inputs.

Question 3

(a) Comprehensive responses

- Created a 2D array and stored the correct data in each element.
- Initialised the two variables to the correct values.

Limited responses

- Created a 1D array, two 1D arrays, an array of tuples or created a class instead of using a 2D array.

(b)(i) Comprehensive responses

- Used the 2D array to access each data element and pointer.
- Started at the index stored in `StartLinkedList` and followed the pointer from the 2D array each time until this was `-1`.

Limited responses

- Iterated through each array element from index 0 without following the pointers.
- Output the data and the pointer instead of just the data.

(ii) Comprehensive responses

- Used the 2D array to access each pointer.
- Started at the index stored in `StartEmptyList` and followed the pointer from the 2D array each time until this was `-1`.

Limited responses

- Iterated through each array element from index 0 without following the pointers.
- Output the data in each element instead of the pointer.

(iii) Comprehensive responses

- Called each of the procedures and output the required messages before each call.

Limited responses

- Called the procedures without the required text.

(iv) Comprehensive responses

- Produced a screenshot showing each of the headings with the correct data and pointers beneath each heading.

Limited responses

- Produced a screenshot showing the incorrect data and pointers, for example output the data for each element in the array, or outputting the data in the empty list.

(c) (i) Comprehensive responses

- Used a 2D array correctly to access each data and pointer.
- Checked if the linked list was full to output the appropriate message.
- Checked if the linked list was empty and adjusted the appropriate pointer.
- Followed the pointers from the element at `StartLinkedList` until the end of the list was found.
- Stored the data in the array at the index `StartEmptyList` and initialised the pointer to `-1` to indicate the end of the list.
- Updated `StartEmptyList`.

Limited responses

- Used a 1D array or made use of a class and nodes to check and update data and pointers.
- Stored data in index `[StartEmptyList]` without indicating the second dimension, such as `[0]`, to store the data within.
- Stored the data in the index pointed to by `StartLinkedList` instead of `StartEmptyList`.
- Took a value as input instead of using a parameter.

(ii) Comprehensive responses

- Called `AddItem()` with the argument `8451`.
- Called both print procedures with suitable headings before each.

Limited responses

- Called `AddItem()` without an argument or with an incorrect value.
- Called the procedures without a heading.

(iii) Comprehensive responses

- Produced a screenshot showing the data and pointers with suitable headings after `8451` was inserted.

Limited responses

- Produced a screenshot showing insufficient, or additional, pointers and data.
- Produced a screenshot showing data for the empty list instead of pointers.
- Produced a screenshot without suitable headings before each output.

COMPUTER SCIENCE

Paper 9618/43 Practical
--

Key messages

What candidates did well

- Candidates produced clear evidence of their code for each question in the correct space in the evidence document.
- Many candidates had a good understanding of object-oriented programming and could create classes and constructors.

What candidates needed to improve

- Candidates often missed the need to store or make use of a return value from a function call.
- Candidates need to improve their understanding of the creation, initialisation and manipulation of 2D arrays.
- Some candidates created tuples within a 1D array, whilst others chose to create a class instead of using the required 2D arrays.

Comments on specific questions

Question 1

(a) Comprehensive responses

- Created a 1D array.

Limited responses

- Used a comment to state that a 1D array was intended without the code to create one.

(b) Comprehensive responses

- Created a procedure and took 15 inputs from the user.
- Used a function to convert each input into lowercase and stored each in the array.

Limited responses

- Took 9 inputs from the user.
- Stored the input without converting it to lowercase.

(c) Comprehensive responses

- Created a procedure and output all of the data from the global array in the correct format.

Limited responses

- Output most of the elements in the array, for example only iterating 14 times.
- Output each element without a space between them, or with each on a new line.

(d)(i) Comprehensive responses

- Called both procedures in the correct order.

(ii) Comprehensive responses

- Produced a screenshot showing all 15 inputs from the user with the correct case and the correct final output.

Limited responses

- Produced a screenshot that showed the result from the program without including the inputs from the user.
- Changed the case of the data on input.

(e) Comprehensive responses

- Took an input and converted it to lowercase.
- Iterated through each element in the array and compared it to the input, keeping a counter of how many matched.
- Output the final count.

Limited responses

- Took an input without converting to lowercase.
- Returned the final count instead of outputting it.

(f) (i) Comprehensive responses

- Created a procedure that performed a bubble sort on the array.
- Sorted the array into ascending order by the length of each string.
- Used a flag to identify when a pass had been completed without a swap and used this in the outer loop to stop when the array was sorted.

Limited responses

- Sorted the array by comparing the strings instead of the length.

(ii) Comprehensive responses

- Called all three procedures in the given order.

(iii) Comprehensive responses

- Produced a screenshot that showed all 15 inputs with the correct cases.
- Produced a screenshot that showed the output of 5 and the sorted data.

Limited responses

- Produced a screenshot showing the sorted data without evidence of the data that was input.
- Produced a screenshot without the count of 5.
- Produced a screenshot showing the data incorrectly sorted.

Question 2

(a) (i) Comprehensive responses

- Created a class containing a constructor.
- Declared the five attributes with appropriate data types.
- Assigned a parameter to each attribute within the constructor.

Limited responses

- Created a constructor that was not within a class, for example in Python where the indents were incorrect.
- Did not identify the data types for each attribute.
- Assigned values to the attributes instead of assigning the parameters.

(ii) Comprehensive responses

- Created five get methods without parameters that returned the correct attributes without any additional changes to these values.

Limited responses

- Took a parameter and attempted to assign this to the attribute or return this value.
- Created functions instead of procedures, for example missing the `self` (or equivalent) in Python.

(b) Comprehensive responses

- Opened the file and closed the file in appropriate places.
- Used exception handling for all file access and included an appropriate output message in case of error.

- Read in each line from the file and split it by a comma.
- Created a new instance of the class `Character` with the split data and stored these in the array.

Limited responses

- Opened the file without closing the file, or closed the file in an inappropriate place for example within the loop where each line was read.

(c) Comprehensive responses

- Created a procedure that took a parameter that output the data for each object in the array.
- Made appropriate use of get methods.
- Included the additional text to format the output appropriately.

Limited responses

- Output the data for one object, for example a parameter, instead of iterating through the array.
- Output the data by accessing the attributes directly instead of using the get methods.
- Output the data directly without formatting it as required.

(d)(i) Comprehensive responses

- Stored the return value from `ReadData()` and used this as the argument in the call to `OutputCharacters()`.

Limited responses

- Called both subroutines without making use of the return value from `ReadData()`.

(ii) Comprehensive responses

- Produced a screenshot showing all of the 15 characters output in the correct format.

Limited responses

- Produced a screenshot that showed one or more characters, without all of them included.

(e)(i) Comprehensive responses

- Output a suitable message asking the user to input two values, then took these as input.
- Designed a way of identifying which criteria the user was inputting as well as the data. For example, asking them to enter a number for the criteria, then enter the actual criteria. This was then used to identify the attribute to use in the comparison.
- Identified characters that matched both of the criteria and used a count variable to identify the quantity identified.
- Stored the matching characters in a new array.
- Called `OutputCharacters()` with the new, or updated, array.
- Made appropriate use of get methods to access the attributes.
- Took one criteria at a time, created a new array of the characters that matched the first criteria, before repeating this with the new array and the second criteria.
- Used short inputs from the user instead of trying to match words and phrases, for example entering 1 for colour, 2 for strength.

Limited responses

- Only asked the user to input the two criteria that were to be used to generate the outputs required in the question paper, instead of allowing any criteria to be entered.
- Took the data criteria as input, for example 'red' without identifying which criteria this is for.

(ii) Comprehensive responses

- Produced two screenshots that showed the given inputs and the matching outputs.

Limited responses

- Produced screenshots that showed the incorrect results for each set of input data.

Question 3

(a) Comprehensive responses

- Created a 2D array with 200 by 3 elements and initialised each element to -1.
- Initialised both pointers to 0.

Limited responses

- Created a 1D array, multiple 1D arrays, created a record or class instead of using a 2D array.

(b) Comprehensive responses

- Checked if the tree was empty by checking the value of `TreeRootPointer` or `FirstFreeNodePointer`.
- Checked if the tree was full (more than 99 nodes in the 2D array) and output a suitable message.
- Stored the parameter in the element in the 2D array pointed at by `FirstFreeNodePointer`.
- Compared the data in `TreeRootPointer` to identify if the parameter needed to be stored to the left or the right of the node, then followed the pointers repeatedly until a free element was found.
- Updated the appropriate parent pointer to `FirstFreeNodePointer`.
- Incremented `FirstFreeNodePointer`.

Limited responses

- Stored the data in a 1D array or as the attribute to class.
- Compared the first data element to the parameter, then stored the data immediately in the next node without repeatedly checking to find an empty node.

(c) Comprehensive responses

- Iterated through each array element and output both pointers and the data in the given format.

Limited responses

- Iterated through some array elements, or only one element.
- Output each index of the array without formatting the data.
- Accessed data from a 1D array or a record or class, instead of from the 2D array.

(d)(i) Comprehensive responses

- Took 10 inputs from the user and called `AddNode()` with each input.
- Called `OutputArray()` once.

Limited responses

- Took an incorrect number of inputs, for example iterating 9 times instead of 10.

(ii) Comprehensive responses

- Produced a screenshot showing the 10 given inputs followed by the tree data and pointers.

Limited responses

- Produced a screenshot showing the data in the tree without the inputs.
- Produced a screenshot showing part of the binary tree without all of the data and pointers.

(e)(i) Comprehensive responses

- Created a recursive procedure.
- Accessed the data and pointers using the 2D array.
- Checked the pointers and output the data in the order left call, output, right call.

Limited responses

- Created an iterative procedure.
- Attempted to access the data and pointers from a 1D array or as attributes in a class.

(ii) Comprehensive responses

- Called `InOrder()` and passed the root pointer as the argument.

Limited responses

- Called `InOrder()` without an argument.

(iii) Comprehensive responses

- Produced a screenshot showing the given data input and the output of the tree in the correct order.

Limited responses

- Produced a screenshot showing the result from `InOrder()` without showing the data input.