COMPUTER SCIENCE

Paper 9618/11
Theory Fundamentals

Key messages

Accuracy in answers is important. It is essential that candidates understand the technical language associated with the subject and that these technical terms are used appropriately in responses. Candidates are expected to demonstrate more in-depth knowledge at this level of study.

When answering questions, it is vital that candidates relate their answers to the context given.

In answers involving binary numbers, some candidates overwrite the digits to correct a mistake. In these instances, it is often impossible to determine if a zero has been replaced with a one, or a one replaced with a zero.

When the answer to a question is the result of a calculation candidates should ensure that it is very clear to the examiner which is the value to be marked.

General comments

Questions may have quite specific instructions and it essential that candidates follow these instructions. Some candidates had put the incorrect number of ticks or lines for tick box and line drawing questions.

Questions that required calculations were usually completed successfully. 'Describe' and 'explain' questions appeared to be more challenging.

Comments on specific questions

These comments should be read in conjunction with the published mark scheme for this paper.

Question 1

(a) Some candidates were able to successfully connect each binary value on the left with the equivalent value on the right. Some candidates need to improve their understanding of the difference between the binary prefixes and the decimal prefixes.

(b) (i) This question was answered very well. Many candidates were able to correctly carry out the binary addition. Some candidates should understand that when asked to show working, the method used must be clearly seen.

(ii) Some candidates found this question challenging. Imprecise answers such as '*bits going off the end*' are not enough at this level. Reference to the number of bits or the maximum possible value would be expected.

(c) There were many correct answers to this question. A frequent incorrect answer was 150 where the denary values of the Hexadecimal digits had been appended, rather than using the hexadecimal place values.

Question 2

(a) There was some confusion between data security and data integrity, with candidates explaining the difference the wrong way round. Answers such as, *'data security keeps the data safe'* and *'data integrity protects the integrity of the data'* occurred frequently. These are not precise enough at this level.

(b) (i) Imprecise answers such as *'validation checks that the data is valid'* were common. There also seems to be a misconception that validation ensures that data is correct. Validation only checks that the data is reasonable or that it fits certain parameters. There were many appropriate correct examples of validation seen.

(ii) There was some confusion with validation in this question, with some candidates describing validation checks rather than verification. There was a misunderstanding that verification ensures the data is correct. Verification only ensures that the data is unchanged on input or after transmission. If the original is incorrect, the verified data will also be incorrect.

(c) Many candidates found this question challenging. Answers such as *'they are both malware'* were common. The question already stated this. Candidates were asked to give similarities (things that are the same) and differences. Answers frequently described what is meant by malware without reference to spyware and viruses. There is also a need to include statements that demonstrate some more technical knowledge. When writing about a difference between the two threats, an individual statement about each would be expected. For example, *'a virus damages data, spyware does not'* is not enough.

Question 3

(a) This question was answered very well. Many candidates were able to give a correct logic expression for the logic circuit.

(b) This question was also answered very well. Many candidates were able to correctly complete the truth table. This was an example of a situation where some candidates had overwritten the binary values to make corrections and examiners could not determine if some digits were meant to be a one or a zero.

(c) Many candidates were able to correctly identify a logic gate not used in the given logic circuit.

Question 4

(a) Two examples of Professional Ethical Bodies are given in section 7.1 of the syllabus. There was considerable confusion about what sort of organisation Francis was advised to join. Many answers assumed that he was going to work for this organisation, or that the organisation was part of the multinational company. Candidates need to improve their understanding of this topic.

(b) (i) There were some interesting and imaginative answers to this question. Many candidates correctly suggested that he should discuss the situation with his manager and take advantage of various opportunities for training.

(ii) Candidates were told in the rubric of the question that the IDE provided debugging tools and were asked to identify three **other** features. While there were some very good answers to this question, many responses included items that would be part of the debugging tools.

(c) Some candidates found this question challenging. The question asked why Francis had acted unethically, and so responses just saying what he should have done, for example, *'he should act in the best interest of the company'* do not answer the question. Answers need to include statements about why he hasn't acted in the company's best interest by doing what he has done. A more suitable answer is, *'he did not act in the best interest of the company because not correcting the error now could lead to more problems later'*.

(d) This question was answered very well. Many candidates were able to insert suitable words in at least some of the spaces.

Question 5

(a) This question was answered very well. Many candidates were able to correctly identify the primary and foreign Keys.

(b) This question asked specifically about access rights to the database given in the question, so answers in that context would be expected to demonstrate the application of knowledge. Responses frequently described the generic implementation of authorisation methods such as passwords to access any database, rather than the specific use of access rights to limit operations that users can perform on the given database once they have logged in.

(c) (i) Candidates should be aware that when a set of database tables is given in the stem of the question, any subsequent use of table or field names must reflect exactly what has been given. A common mistake was to label the table in the `From` clause as `CARS` rather than `CAR`.

(ii) This question was answered well. Many candidates were able to correctly complete the DML statement. Some candidates need to improve their understanding of the `INSERT INTO` command.

Question 6

(a) There were some very good answers to this question. Many candidates were able to identify the errors and write the correct statement. Some candidates found it more challenging to describe the error. Statements such as, *'the MDR should be in double brackets'* are not enough. A description of why the MDR needs to be in double brackets is required.

(b) This question was answered well. Many candidates were able to correctly complete the instructions.

Question 7

(a) (i) This question too needed to be read carefully. The question asked why the sound is closer to the original when a higher sample rate is used. Many responses simply stated what is meant by a higher sample rate with little or no reference to any improvement in the accuracy of the sound. Candidates would be expected to make some reference to reduced quantisation errors or a better digital approximation to the analogue wave form.

(ii) There was some confusion here with an increased sample resolution. Some candidates wrote about increasing the bit depth, rather than number of samples.

(b) (i) There were some good answers to this question. Many candidates stated that the file would take less time to upload or download, and that email providers frequently had limits on the size of files. *'Easier to transmit'* is not enough at this level, a statement about speed of transmission would be expected.

(ii) This was another question where candidates were being asked to apply their knowledge. It specifically asked how lossless compression could compress the sound file. Generic descriptions of lossless compression were seen frequently and there was seldom any reference to the sound file. Many answers included examples of compressing text or images.

Question 8

(a) Many candidates correctly identified the network as a LAN and were able to give one justification of their choice, usually to do with the size of the area involved. Finding a second justification proved more challenging.

(b) This question was answered well. Many candidates were able to correctly identify two devices that could be used to physically connect the computers to the rest of the network.

(c) This question was more challenging and candidates need to improve their understanding of the functions of a WNIC. There was a frequent misconception that the WNIC provided an IP address rather than a unique MAC address. Answers such as, *'a WNIC functions like a NIC, but wirelessly'* are not precise enough and generic descriptions of a NIC do not answer the question.

COMPUTER SCIENCE

Paper 9618/12
Theory Fundamentals

Key messages

Accuracy in answers is important. It is essential that candidates understand the technical language associated with the subject and that these technical terms are used appropriately in responses. Candidates are expected to demonstrate more in-depth knowledge at this level of study.

When answering questions, it is vital that candidates relate their answers to the context given.

In answers involving binary numbers, some candidates overwrite the digits to correct a mistake. In these instances, it is often impossible to determine if a zero has been replaced with a one, or a one replaced with a zero.

When the answer to a question is the result of a calculation candidates should ensure that it is very clear to the examiner which is the value to be marked.

General comments

Questions may have quite specific instructions and it essential that candidates follow these instructions. Some candidates had put the incorrect number of ticks or lines for tick box and line drawing questions.

Questions that required calculations were usually completed successfully. 'Describe' and 'explain' questions appeared to be more challenging.

Comments on specific questions

These comments should be read in conjunction with the published mark scheme for this paper.

Question 1

Many candidates were able to correctly answer this question. Some candidates did not follow the instructions in the question and drew one line from each of the boxes on the right, instead of one line from each of the measures.

Question 2

(a) Many candidates were able to correctly answer this question. There was some confusion between the NAND and NOR gates, with some candidates reversing the truth tables.

(b) There were also many correct answers to this question. Some candidates had realised that the expression given was equivalent to two NAND gates and correctly drew the appropriate logic circuit. Common incorrect logic circuits either had both NOT gates on the final output, or a NOT gate on each of the two inputs A and B.

Question 3

(a) Some candidates need to improve their understanding of the principal operation of a microphone. Many candidates understood the need for a diaphragm or similar. The purpose of the diaphragm and the way in which sound waves are converted to electrical signals was less well understood. Some incorrect answers described the operation of a speaker rather than a microphone; others described sampling of the sound wave.

(b) (i) Many candidates correctly identified the topology. The justification for the topology was more challenging. Answers such as, '*because all the devices connect to the router*', added nothing to the information given in the question. At this level candidates are expected to demonstrate more in-depth knowledge.

(ii) The question made it clear that the router in this case formed part of a home network, so answers would be expected in that context. Many candidates found this challenging and listed generic functions of a router rather than applying the answers to the home network. Imprecise statements such as, *'to allow the devices to connect to the Internet'* which simply repeated information given in the question are not enough. Some candidates need to read the question carefully. It asked for functions of the router not the WAP.

Question 4

(a) This question was answered very well. Many candidates were able to correctly convert the unsigned binary integer to denary.

(b) There were many correct answers to this part question. A common error was the omission of the minus sign.

(c) This question was answered very well. Many candidates were able to correctly convert the binary value to hexadecimal.

(d) Some candidates found this question challenging and need to improve their understanding of BCD as a four-bit value. Incorrect answers stated that the register held a value greater than 9, rather than each nibble of the register being greater than 9.

(e) (i) There were many correct answers to this question. Some candidates need to understand that for the working mark to be awarded, it must be clear what method has been used.

(ii) Candidates found the binary subtraction more challenging. The question asked for the contents of register 2 to be taken away from the contents of Register 1. Some candidates performed the subtraction the wrong way around.

Question 5

(a) Many candidates need to improve their understanding of the terms associated with vector graphics. There was considerable confusion between a drawing list and a drawing object. Some candidates attempted to describe a property without using a different word or without associating the property with a drawing object. A suitable answer would be, *'a property is an aspect of the appearance of a drawing object'*. Examples given were often generic rather than from the logo shown in the question.

(b) (i) The command word in this question is *'describe'*. This means that, at this level, it is not enough to write, for example, *'bitmap images pixelate when enlarged, vector graphic images do not'*. A description is expected to include the reasons why a bitmap image pixelates and why a vector graphic does not. The question also asks for two differences.

(ii) Questions need to be read carefully. This question asks for two lossy methods of compression that can be used on a photograph. Incorrect answers frequently described lossy compression and the effects of the compression on the image, not methods of lossy compression. Statements such as, *'use JPEG'* or *'remove colours the eye cannot see'* are not enough at this level when the question asks for a description. The answer needs to include how using a JPEG or removing the colours would compress the photograph.

Question 6

(a) (i) This question was answered well. Many candidates were able to correctly identify the appropriate stage for each task.

(ii) Many candidates found completing this diagram challenging. Common incorrect answers included a relationship between `PLANT` and `CUSTOMER` or included the crow's feet on the wrong end of the one-to-many relationships. Some candidates need to improve their understanding of a recognised notation for the relationships.

(b) Some candidates need to improve their understanding of the purpose and contents of a data dictionary. Incorrect answers frequently described the purpose and contents of the Database Management System (DBMS) rather than the data dictionary. Care is needed in the use of the terminology. There is some confusion between data and information.

(c) (i) Many candidates were able to correctly complete the `FROM` clause and the first part of the `WHERE` clause. A common incorrect answer for the second part of the `WHERE` clause was the omission of the quotation marks. Some candidates need to improve their understanding of the use of the `SUM` function in SQL.

(ii) Some candidates found writing SQL statements without any guidance more challenging and need to improve their understanding of the use of the `ALTER TABLE` statement in SQL to include an additional field.

Question 7

(a) This question was answered well. Many candidates were able to correctly identify the events that were example of hardware interrupts and those that were examples of software interrupts.

(b) Section 5.1 of the syllabus clearly lists memory management and file management as two different management tasks of an operating system. There was some confusion between the two tasks in the answers to this question, with some candidates incorrectly using the terms *'memory'* and *'storage'* inter-changeably. Some candidates found this question very challenging and gave vague answers about the functions of an operating system in general such as providing a user interface.

(c) This is an example of a question that needed to be read carefully. Identifying two utility programs seemed to be straightforward. Stating how the utility programs improved the performance of a computer proved to be much more challenging. Most of the answers described the operation of the named program, giving little indication of how using the software would improve performance.

Question 8

(a) (i) Many candidates found this question challenging. Candidates need to improve their understanding of registers and their role in the F-E cycle. A register is a storage location; its function is to hold or store a binary value. A register cannot transfer the value; the content of the register is transferred by other means.

(ii) This question was answered well. Many candidates were able to correctly identify one other special purpose register.

(b) (i) This question was answered well. Many candidates were able to correctly complete the table. Some candidates need to improve their understanding of the `LSL` and `LSR` instructions.

(ii) This is another example of a question that needed to be read carefully. The question instructs candidates to use the given instruction set, yet many answers included mnemonics not included in the table provided. Another common error was the omission of any operands in the last three instructions.

(iii) Many candidates were able to correctly identify an additional mode of addressing. Indirect and indexed addressing were popular choices. Some candidates found describing their chosen method more challenging. Indexed addressing was regularly incorrectly described as adding the contents

of the index register to the contents of the operand, rather than correctly described as adding the contents of the index register to the actual operand.

COMPUTER SCIENCE

Paper 9618/13
Theory Fundamentals

Key messages

Accuracy in answers is important. It is essential that candidates understand the technical language associated with the subject and that these technical terms are used appropriately in responses. Candidates are expected to demonstrate more in-depth knowledge at this level of study.

When answering questions, it is vital that candidates relate their answers to the context given.

In answers involving binary numbers, some candidates overwrite the digits to correct a mistake. In these instances, it is often impossible to determine if a zero has been replaced with a one, or a one replaced with a zero.

When the answer to a question is the result of a calculation candidates should ensure that it is very clear to the examiner which is the value to be marked.

General comments

Questions may have quite specific instructions and it essential that candidates follow these instructions. Some candidates had put the incorrect number of ticks or lines for tick box and line drawing questions.

Questions that required calculations were usually completed successfully. 'Describe' and 'explain' questions appeared to be more challenging.

Comments on specific questions

These comments should be read in conjunction with the published mark scheme for this paper.

Question 1

(a) Some candidates were able to successfully connect each binary value on the left with the equivalent value on the right. Some candidates need to improve their understanding of the difference between the binary prefixes and the decimal prefixes.

(b) (i) This question was answered very well. Many candidates were able to correctly carry out the binary addition. Some candidates should understand that when asked to show working, the method used must be clearly seen.

(ii) Some candidates found this question challenging. Imprecise answers such as '*bits going off the end*' are not enough at this level. Reference to the number of bits or the maximum possible value would be expected.

(c) There were many correct answers to this question. A frequent incorrect answer was 150 where the denary values of the Hexadecimal digits had been appended, rather than using the hexadecimal place values.

Question 2

(a) There was some confusion between data security and data integrity, with candidates explaining the difference the wrong way round. Answers such as, *'data security keeps the data safe'* and *'data integrity protects the integrity of the data'* occurred frequently. These are not precise enough at this level.

(b) (i) Imprecise answers such as *'validation checks that the data is valid'* were common. There also seems to be a misconception that validation ensures that data is correct. Validation only checks that the data is reasonable or that it fits certain parameters. There were many appropriate correct examples of validation seen.

(ii) There was some confusion with validation in this question, with some candidates describing validation checks rather than verification. There was a misunderstanding that verification ensures the data is correct. Verification only ensures that the data is unchanged on input or after transmission. If the original is incorrect, the verified data will also be incorrect.

(c) Many candidates found this question challenging. Answers such as *'they are both malware'* were common. The question already stated this. Candidates were asked to give similarities (things that are the same) and differences. Answers frequently described what is meant by malware without reference to spyware and viruses. There is also a need to include statements that demonstrate some more technical knowledge. When writing about a difference between the two threats, an individual statement about each would be expected. For example, *'a virus damages data, spyware does not'* is not enough.

Question 3

(a) This question was answered very well. Many candidates were able to give a correct logic expression for the logic circuit.

(b) This question was also answered very well. Many candidates were able to correctly complete the truth table. This was an example of a situation where some candidates had overwritten the binary values to make corrections and examiners could not determine if some digits were meant to be a one or a zero.

(c) Many candidates were able to correctly identify a logic gate not used in the given logic circuit.

Question 4

(a) Two examples of Professional Ethical Bodies are given in section 7.1 of the syllabus. There was considerable confusion about what sort of organisation Francis was advised to join. Many answers assumed that he was going to work for this organisation, or that the organisation was part of the multinational company. Candidates need to improve their understanding of this topic.

(b) (i) There were some interesting and imaginative answers to this question. Many candidates correctly suggested that he should discuss the situation with his manager and take advantage of various opportunities for training.

(ii) Candidates were told in the rubric of the question that the IDE provided debugging tools and were asked to identify three **other** features. While there were some very good answers to this question, many responses included items that would be part of the debugging tools.

(c) Some candidates found this question challenging. The question asked why Francis had acted unethically, and so responses just saying what he should have done, for example, *'he should act in the best interest of the company'* do not answer the question. Answers need to include statements about why he hasn't acted in the company's best interest by doing what he has done. A more suitable answer is, *'he did not act in the best interest of the company because not correcting the error now could lead to more problems later'*.

(d) This question was answered very well. Many candidates were able to insert suitable words in at least some of the spaces.

Question 5

(a) This question was answered very well. Many candidates were able to correctly identify the primary and foreign Keys.

(b) This question asked specifically about access rights to the database given in the question, so answers in that context would be expected to demonstrate the application of knowledge. Responses frequently described the generic implementation of authorisation methods such as passwords to access any database, rather than the specific use of access rights to limit operations that users can perform on the given database once they have logged in.

(c) (i) Candidates should be aware that when a set of database tables is given in the stem of the question, any subsequent use of table or field names must reflect exactly what has been given. A common mistake was to label the table in the `From` clause as `CARS` rather than `CAR`.

(ii) This question was answered well. Many candidates were able to correctly complete the DML statement. Some candidates need to improve their understanding of the `INSERT INTO` command.

Question 6

(a) There were some very good answers to this question. Many candidates were able to identify the errors and write the correct statement. Some candidates found it more challenging to describe the error. Statements such as, *'the MDR should be in double brackets'* are not enough. A description of why the MDR needs to be in double brackets is required.

(b) This question was answered well. Many candidates were able to correctly complete the instructions.

Question 7

(a) (i) This question too needed to be read carefully. The question asked why the sound is closer to the original when a higher sample rate is used. Many responses simply stated what is meant by a higher sample rate with little or no reference to any improvement in the accuracy of the sound. Candidates would be expected to make some reference to reduced quantisation errors or a better digital approximation to the analogue wave form.

(ii) There was some confusion here with an increased sample resolution. Some candidates wrote about increasing the bit depth, rather than number of samples.

(b) (i) There were some good answers to this question. Many candidates stated that the file would take less time to upload or download, and that email providers frequently had limits on the size of files. *'Easier to transmit'* is not enough at this level, a statement about speed of transmission would be expected.

(ii) This was another question where candidates were being asked to apply their knowledge. It specifically asked how lossless compression could compress the sound file. Generic descriptions of lossless compression were seen frequently and there was seldom any reference to the sound file. Many answers included examples of compressing text or images.

Question 8

(a) Many candidates correctly identified the network as a LAN and were able to give one justification of their choice, usually to do with the size of the area involved. Finding a second justification proved more challenging.

(b) This question was answered well. Many candidates were able to correctly identify two devices that could be used to physically connect the computers to the rest of the network.

(c) This question was more challenging and candidates need to improve their understanding of the functions of a WNIC. There was a frequent misconception that the WNIC provided an IP address rather than a unique MAC address. Answers such as, *'a WNIC functions like a NIC, but wirelessly'* are not precise enough and generic descriptions of a NIC do not answer the question.

COMPUTER SCIENCE

Paper 9618/21
Problem-Solving and Programming

Key messages

The practical skills on this paper involve analysing and understanding the requirement as well as designing and presenting a solution. A scenario is often used to present the requirements. Candidates need to be able to identify the key elements of each requirement, for example, the need for an iterative structure, when designing their solution. The development of these skills requires practice.

This subject makes use of many. The technical words and phrases used have specific, defined meanings and they need to be used correctly.

Answers should be as precise and specific as possible. Candidates should familiarise themselves with the meanings of the command words used in this paper and write down their answers accordingly. Candidates need to read each question carefully to make sure they understand what is being asked and should not assume that simply because a particular question shares some key phrases with those from a past paper that the required answer is the same. Candidates must not answer a question by simply repeating phrases from the question.

General comments

If answers are crossed out, the new answers must be written clearly so that the text may be read easily and the correct mark awarded.

Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful if this text is crossed out.

If typed answers are provided, it is very helpful if these are organised so that individual answers do not span page breaks. This is particularly important for pseudocode answers. If the question involves completing a table, they typed answers should clearly indicate any unfilled rows.

Familiarity with fundamental programming concepts is vital. Lack of understanding is often illustrated by the confusion between a literal value and an identifier, or the misuse of `OUTPUT` in place of `RETURN`. Many candidates lack knowledge of the use of parameters, often replacing parameters to a procedure or function with a series of prompts and inputs within the body of the subroutine.

The functions and operators that are available for use in pseudocode answers are described in the **insert** which accompanies the paper. Candidates must use these in their responses.

Comments on specific questions

It is recommended that the following specific comments be read in conjunction with the published mark scheme for this paper.

Question 1

(a) (i) Many candidates successfully answered this question, usually by stating that the rules or the grammar of the programming language had been followed or had not been broken. Other gave an example, but often the example was too vague, such as ‘contains no spelling mistakes’.

Some simply gave a textbook definition of a syntax error, without relating it to the given scenario.

Very few candidates addressed the fact that the program could run or be compiled/interpreted.

(ii) Many candidates fully answered this question, either my suggesting a logic error which 'causes the program to not produce the expected results' or a run-time which 'causes the program to crash'. Many incorrect answers were seen that bore little relation to the question. Many candidates did not attempt this question.

(b) This question was answered well by most candidates. Where a mistake was made, it was usually to suggest Integer for the first row.

(c) (i) The majority of candidates gained two or three marks. A common mistake was to suggest that the flight number was essential when making the booking.

(ii) Very few answers correctly described the benefit. Often the description added little to the wording from the stem and on many occasions the description related to the customer booking activity, such as suggesting that it would be 'easier to make a booking'. There are many incorrect techniques that bore little relation to the question. Many candidates did not attempt this question.

(iii) Most candidates were able to suggest two additional pieces of essential information. Departure date and arrival airport were the most common.

Question 2

(a) There was a good spread of marks across the range.
Many candidates recognised that this was a bubble sort. The use of two nested loops was referred to in many answers but often the detail was unclear. Strong answers described the decreasing number of comparisons required by the inner loop and the use of a 'no swaps' variable or similar to efficiently terminate the outer loop.

A detailed description of the swapping process was often seen.

Many solutions focused on finding the biggest number and storing in a different variable, rather than sorting the whole array.

(b) A wide range of answers was seen, with a significant number of perfect solutions.

Strong responses used a pre-condition loop (MP2) and usually gained the full six marks.
Candidates who opted for a post-condition loop were often unable to correctly alter the terminating condition shown in the flowchart (MP3).
Many weaker answers did not use a conditional loop but attempted to use an `IF ... ENDIF` structure.

MP4 was usually gained, provided the statements were inside some form of loop.

An invalid pseudocode statement (usually involving the keyword `CALL`) prevented many candidates from gaining MP5.

MP6 was straightforward and gained by many candidates. A small number decided to output the value 27, and others failed to gain this point as the call was placed in an incorrectly formed `IF ... ENDIF` clause.

Question 3

(a) (i) Many candidates referred to the increment of the End of Queue pointer and the positioning of the new data item at the corresponding position in the array.

Vague answers were common through imprecise language or poorly constructed answers such as stating that 'Octopus becomes the new End of Queue pointer'.

(ii) A small number of well-described answers was seen.

Answers which used the correct technical language such as 'The value Frog is assigned to the variable `AnimalName`' were rare.

A reference to the increment of the Front of Queue pointer to point to Cat was often missing.

Several answers confused the operation of queues with other ADTs.

Many candidates explained the FIFO mechanism without applying it to the given scenario.

(iii) The simple scenario describing the operation of the queue stated that the Front of Queue pointer points to the next item to be removed. This indicates that there will be always at least one item in the queue. (More complicated scenarios may be introduced in future questions, possibly including a count variable to indicate the number of items in the queue or a Boolean variable to indicate the queue is empty.)

As for previous parts, vague phrasing was commonplace, such as stating 'there is only one element in the queue'.

Many candidates did not attempt this question.

(b) (i) This question was generally well answered with many candidates scoring full marks. A small number of candidates made no attempt at this question.

(ii) A more complex 'trace' where the circular operation of the stack was involved. It was not uncommon for the queue pointers to be pointing at the correct values, but that these would be in the wrong array locations.

(c) MP2 was the mark most often gained for reference to the increment of the End of Queue pointer.

Very few answers addressed the other points, and little understanding of how a circular queue operates was evident.

A common mistake was to refer to 'searching for an empty location'.

Question 4

(a) Most candidates understood that it was boundary/extreme values and abnormal values (outside of the 0 to 40 range) that were required.

Candidates often repeated normal test data values.

Candidates need to carefully read the question. The rubric stated, 'The sensor produces ... an integer...'. Despite this, many candidates offered character data and real numbers as test data.

(b) Most candidates gained the marks for the 'Heaters on/Heaters off' actions by showing the inputs that would trigger the corresponding changes of state.

Many also gained the final mark for labelling the two 'no change of state' arrows with an input that would not result in a state change.

Question 5

(a) Generally, well answered with most candidates. An incorrect reason suggested by many was that the separator character was 'easy to see', rather than the fact that the character did not occur in the data

(b) Many answers correctly stated 'design'. Some answers hinted at the correct answer by suggesting terms such as 'planning' but the correct term as given in the syllabus.
Several answers simply repeated the word 'development' from the question, which asked for 'one stage from the program development life cycle'.

A significant number of candidates made no attempt at this question.

(c) Many candidates gained the mark (MP1) for the correct function heading. Several answers omitted the `ENDFUNCTION` statement which prevented this mark being given. The parameter and return type were usually given correctly.

Most candidates did not gain MP2. In many cases, the quotation marks around the literal value were omitted, so treating `LogFile` as an identifier rather than a literal string. Opening the file in `WRITE` mode rather than `APPEND` mode was common, as was omitting the `CLOSEFILE` statement, or not including the filename.

Most solutions contained a loop and this mark (MP4) was often given.
In some answers, the loop statement incorrectly omitted the assignment operator as illustrated:

```
FOR X 1 TO 2000
```

MP4 and MP5 presented the biggest challenge to most candidates, and only a small number of answers correctly addressing these points.

Many candidates gained MP6, often as a follow-through mark if either MP4 or MP5 were incorrect, and the error had been repeated. An occasional mistake seen was to write only the first six characters to the file, rather than the whole line.

Most solutions included an attempt at MP7 and in many cases the mark was given. It is encouraging to see the number of solutions that included the declaration and initialisation of the variable used to store the count value. A small number of solutions chose to use `OUTPUT` rather than `RETURN`.

A significant number of candidates made little or no attempt at a solution.

Question 6

(a) The solution required a simple algorithm with a loop which included a 2D array assignment. Of the five marks available, three were available for simple and straightforward elements of the solution.

A full range of marks was given. Many answers suggested that the problem had not been understood.

Most meaningful answers contained a count-controlled loop. A count range from 1 to `SetNum` was probably the most common, although others were seen. The marks for the loop were split between MP3 and MP4.

MP5 was the most challenging mark, as not only did the 2D array syntax have to be broadly correct, but also the value used for the column index had to work with the count range used. This mark was given only in a small number of responses.

Many candidates made little or no attempt at a solution.

(b) The full range of marks were given for answers to this question.

Most answers did not address the requirement to search either up or down as specified by the start column value, and so did not gain MP1 and MP2. The small number of solutions that did include this feature usually tested the start column using an `IF .. THEN ... ELSE ... ENDIF` clause which then selected one of two separate linear searches. More compact solutions contained a single search, where the start, end and step vales were defined with reference to the given start column.

MP3 and MP4 were commonly given.

Referencing an individual array element and testing its value (MP5) were included in many solutions.

For MP6, many solutions continued to search after an element with value 1 had been found. A secondary error was that often the saved column value would be overwritten during subsequent iterations.

A common approach to MP7 was to initialise the return value to –1 before the search. This was usually successful, except in the case described above when the correct column value was overwritten.

Few solutions gained MP8. A curious construct seen on several occasions involved placing an additional conditional loop around the count-controlled loop, to provide a mechanism to stop the iteration when the condition is satisfied. This would not be successful as the inner count-controlled loop would always have run to completion.

Many candidates made little or no attempt at a solution.

(c) Several candidates made a very good attempt at this final question.

Most realistic attempts gained the first mark (MP1).

The second mark (MP2) was not given due to treating the module `SearchInRow()` as a procedure rather than a function, for example:

`CALL SearchInRow(ThisRow, 1)` rather than `StartCol ← SearchInRow(ThisRow, 1)`

This mistake also meant that no return value was available.

Many solutions did not gain MP3 because the parameters to `SearchInRow()` were either missing or incorrect.

The correct calculation of the centre column was seen in many solutions (MP4), usually by adding the start and end values and dividing by 2. In many cases, this mark was given as a follow-through from errors relating to MP3. Some solutions calculated the difference between the start and end values but often these failed to then add the calculated value to the start value.

Most solutions did not convert the result of the calculations to an integer value for MP5.

As for MP1, most realistic attempts gained MP6. A small number of solutions chose to use `OUTPUT` rather than `RETURN`.

A significant number of candidates made little or no attempt at this question.

COMPUTER SCIENCE

Paper 9618/22
Problem-Solving and Programming

Key messages

The practical skills on this paper involve analysing and understanding the requirement as well as designing and presenting a solution. A scenario is often used to present the requirements. Candidates need to be able to identify the key elements of each requirement, for example, the need for an iterative structure, when designing their solution. The development of these skills requires practice.

This subject makes use of many. The technical words and phrases used have specific, defined meanings and they need to be used correctly.

Answers should be as precise and specific as possible. Candidates should familiarise themselves with the meanings of the command words used in this paper and write down their answers accordingly. Candidates need to read each question carefully to make sure they understand what is being asked and should not assume that simply because a particular question shares some key phrases with those from a past paper that the required answer is the same. Candidates must not answer a question by simply repeating phrases from the question.

General comments

If answers are crossed out, the new answers must be written clearly so that the text may be read easily and the correct mark awarded.

Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful if this text is crossed out.

If typed answers are provided, it is very helpful if these are organised so that individual answers do not span page breaks. This is particularly important for pseudocode answers. If the question involves completing a table, they typed answers should clearly indicate any unfilled rows.

Familiarity with fundamental programming concepts is vital. Lack of understanding is often illustrated by the confusion between a literal value and an identifier, or the misuse of `OUTPUT` in place of `RETURN`. Many candidates lack knowledge of the use of parameters, often replacing parameters to a procedure or function with a series of prompts and inputs within the body of the subroutine.

The functions and operators that are available for use in pseudocode answers are described in the **insert** which accompanies the paper. Candidates must use these in their responses.

Comments on specific questions

It is recommended that the following specific comments be read in conjunction with the published mark scheme for this paper.

Question 1

(a) Most candidates were able to describe decomposition as the process of breaking down a problem into smaller parts. Many gained the second mark by stating that decomposition would make a problem easier to solve.

A common mistake was to refer to decomposition being used to break down an existing program, rather than a problem. Relative terms are important and the absolute phrase 'easy to solve' was not acceptable. Decomposition can lead to an easier solution, but not necessarily an easy one.

(b) A wide range of answers was given. Common errors included suggesting 42 for the number of dimensions and offering the term 'element' instead of 'index/subscript' for the variable `n`. Another common mistake was giving the maximum and minimum values for the index, rather than the term itself.

(c) This question was generally answered well.

Common mistakes included suggesting `DIV` instead of `INT` and providing incorrect parameter values to the `MID()` function.

(d) Most candidates gained at least one mark, with many providing a completely correct answer.

The terms `TRUE` and `FALSE` were given in the question and candidates should be directed to use these terms rather than alternatives such as 0 and 1, which are ambiguous.

Question 2

(a) The initialisation was often omitted and some solutions placed the corresponding statements inside the loop.

A few solutions used only one count value and incremented the count by 1 each time regardless of whether the number was negative or positive. Some candidates seemed to have misunderstood the problem and used `MOD` to check if the number input was odd or even.

The use of incorrect flow lines was common. Either there would be no output line from a symbol, or several lines would be joined, combining inputs and outputs to give a meaningless 'flow'.

(b) (i) This was a challenging question for candidates**.** Many of those candidates who gave a correct name for the method then simply repeated a phrase from the question to provide the 'reason'. Few candidates mentioned a prototype or working model which are key phrases.

Many answers given were unrelated to the question.

(ii) Only a very small number of candidates gained full marks, for suggesting activities mapping generally to those given on the mark scheme.

Where a single mark was given, this was usually for any decisions relating to the website appearance and content.

Many answers were unrelated to the question. Common mistakes included offering three computing terms (for example: Input, Process and Output) or suggesting three separate stages of the program development cycle, despite the question focussing only on the design stage.

Question 3

(a) (i) Many candidates successfully declared the record structure in pseudocode. Other candidates often gained marks for correct field declarations.

A common problem was to omit the `ENDTYPE` keyword which is needed to mark the end of the declaration.

A small number of answers consisted of unrelated pseudocode statements, often appearing to be an attempt at initialising an array.

(ii) Many candidates provided correct answers with most containing the correct array name and bounds.

A common mistake was to give the array type as string or integer, rather than `Student`.

(iii) There was lack of precision in a large number of responses. Only a small number of candidates specified that an individual field should be assigned a particular value. Where this was done, it was most often to assign an empty string to `StudentID`.

The most common mistake was to ignore the fact that this was an array of records with fields made up of different data types, and to attempt to initialise a complete array element to a value such as an empty string.

A significant number of candidates did not attempt this question.

(iv) More correct answers were seen that for the previous part, with most suggesting 0 or –1 as the value for the unused club field.

A number of answers concentrated on how an unused field should be detected (e.g. 'by using an `IF` statement...'), rather than how an unused field should be indicated.

Many candidates suggested the assignment of a non-integer value to a club field.

A significant number of candidates did not attempt this question.

(b) Mark points MP1, MP2 and MP3 were the most accessible and were given in many cases. Some candidates did not gain MP2 because the input of the club number was inside the loop or was without a required prompt.

Few answers contained the correct syntax for extracting a field from the array of records (MP4 and MP6).

Many candidates recognised the need to check all three club fields and MP5 was sometimes given as a follow-through mark from MP4 if the same error was repeated.

The correct comparison of all club fields was usually implemented either with a logical comparison of each field or by using an appropriate `IF ... ENDIF` clause.

MP7 was attempted by the majority. A common reason for not gaining this mark was the use of the concatenation operator '`&`' on an integer value. Other reasons included placing the `OUTPUT` statement inside the loop, our outputting a variable that had not been incremented (MP6)

A significant number made little or no attempt at this question.

Question 4

(a) Many candidates provided fully correct answers. The first answer was the one provided by most candidates, while the third and fourth were most often incorrect.

Candidates should use the loop construct terminology used in the syllabus, so 'pre-condition' was expected in this case.

(b) Few candidates provided fully correct answers.

Candidates mostly referred to layout features. Answers that mentioned capitalisation were often vague about what had been capitalised. Similarly, the phrase 'meaningful variable names' was often only hinted at with answers such as 'good variables' which was not enough.

Many answers simply suggested pseudocode features such as 'declaring variables' of 'functions'

A significant number of candidates seemed to have not read the question effectively and suggested non-existent features such as 'comments' and 'use of colour'.

(c) (i) **O**ften, candidates did not use the term 'count-controlled'.

Justifications were frequently imprecise. Describing the number of iterations as 'fixed' rather than 'known' was quite common. References to 'the number of times the program will run' were often given, as was the justification 'to save space'.

(ii) A minority of candidates provided fully correct answers.

The most common incorrect answer was to suggest lines 17 and 24. These are the lines which would be *changed* (`WHILE ... ENDWHILE` becoming `FOR ... NEXT`) rather the lines which would be *removed*.

Question 5

(a) Many candidates gained the mark for the correct procedure heading (MP1). One mistake seen occasionally was to declare the search status parameter as type Boolean.

Most candidates did not gain MP2. The concept of passing a filename as a string parameter and then using this in file operations was correct in only a small number of solutions. The difference between an identifier name and a string value is not widely understood. The parameter name was often enclosed in quotation marks when used and in many cases, the parameter names had the suffix '`.TXT`' added. In some solutions, the parameter values were ignored and completely different filenames were used.

Opening the files in the wrong modes was relatively common, and many candidates omitted to close the files before the end of the procedure.

Many recognised that a loop until the end of file was needed (MP3). A common mistake was to omit the filename parameter from the `EOF()` function.

The main problem most encountered was reading 3 lines from the file, comparing the 3 lines with the status parameter and then writing the 3 lines to the new file. Only a few solutions were seen where 3 lines were read and stored separately. Some of these used a `FOR ... NEXT` loop to read the 3 lines but these often assigned each line to the same variable and so the 3 lines could not then be written to the new file following the comparison.

MP6 was commonly given, and it is encouraging to see the number of answers that included the declaration and initialisation of the two count values. Some candidates made this mark unnecessarily complex by counting each line read (and written) and then dividing these figures by 3.

MP7 was a straightforward mark which was missed by many solutions where the concatenation operator ('`&`') was applied to an integer variable. Candidates need to use the `NUM_TO_STRING()` function in this case, or more simply, to use `OUTPUT` with a comma-separated list.

(b) (i) A minority of candidates provided fully correct answers.

A large percentage of candidates seemed to have missed the directive in the question that 'A text file will still be used' and gave responses relating to databases, stacks and, very commonly, arrays.

(ii) As most candidates gave an incorrect response for the previous part, they could not give a correct response here. Many candidates gained a mark here for stating a disadvantage. It was rare to see a correct suggestion for the advantage.

Question 6

(a) This question had a mixed response.

The first mark point (MP1) was very straightforward, but a large number of solutions lacked any form of function heading or ending. Several solutions declared the module as a Procedure.

Weaker solutions often treated the `Screen` array as having only one dimension and, in these cases, a single loop would occasionally gain MP2.

Better solutions acknowledged that `Screen` was a 2D array and included some sort of nested loop. Many of these used nested `FOR` loops which gained the loop marks (MP2 and MP3) and the incrementing index mark (MP6). Solutions based on conditional loops were less common.

A curious construct seen on several occasions involved placing an additional conditional loop around one (or two) count-controlled loops. It is assumed that this was an attempt to provide a mechanism to stop the iteration when the condition is satisfied, but it would not have been successful.

Many candidates could not write a correct statement to access a single element from the array (MP4)

Many solutions were inefficient and would continue to search through the array after an element with the value 1 had been found. In many cases these would overwrite the saved value from the first row (MP5)

An attempt at MP7 was present in most answers. Common mistakes included returning a string value rather than an integer and using `OUTPUT` rather than `RETURN`.

Many candidates made little or no attempt at a solution.

(b) Many candidates recognised that the loop should terminate when an element with the value 1 is found, although in some cases, this had not been a feature of the solution presented in **part 6(a)**.

There were some vague descriptions, such as ‘when the screen is touched’.

Several answers simply referred to basic features of the pseudocode such as ‘a loop is used’ or ‘variables are initialised’.

(c) (i) Few candidates were successful with this question. If one mark was awarded it was usually for describing copying all the code from the other three modules into the new module.

Few answers described the use of parameters to specify which of the individual searches was required or, in the case where a candidate has assumed that the new module would perform all the searches, how the four values could be returned. Many answered stated that the return value(s) should be output rather than returned.

(ii) Answers to this part suggested that candidates had understood the scenario given in **part (c)** better than their answers to **part 6 (c)(i)** had indicated.

Few candidates gained the first mark for stating that there would be fewer lines of code, and several more gained the second mark for stating that the single module would be more complex/difficult to understand.

There were many imprecise answers and a significant number of candidates made no attempt at this part.

(d) A significant number of candidates made a very good attempt at this final question.

Some candidates often treating the four defined modules as procedures rather than functions. For example:

`CALL FirstRowSet()` rather than `StartRow ← FirstRowSet()`

Many candidates gained the mark for testing whether a function returned –1 and if so assigning –1 to the global `CentreRow` (MP2)

Many candidates made use of all four functions and MP3 was often awarded, in many cases, as a follow-through mark from MP1 if the same error had been repeated.

The correct calculation of the centre row and column was seen in many solutions (MP4), usually by adding the start and end values and dividing by 2. Some solutions calculated the difference between the start and end values but often these did not add the calculated value to the start value.

Most solutions did not convert the result of the calculations to an integer value for MP5.

Some candidates ignored the four defined modules and attempted to code the complete solution.

A significant number of candidates made little or no attempt at this question.

COMPUTER SCIENCE

Paper 9618/23
Problem-Solving and Programming

Key messages

The practical skills on this paper involve analysing and understanding the requirement as well as designing and presenting a solution. A scenario is often used to present the requirements. Candidates need to be able to identify the key elements of each requirement, for example, the need for an iterative structure, when designing their solution. The development of these skills requires practice.

This subject makes use of many. The technical words and phrases used have specific, defined meanings and they need to be used correctly.

Answers should be as precise and specific as possible. Candidates should familiarise themselves with the meanings of the command words used in this paper and write down their answers accordingly. Candidates need to read each question carefully to make sure they understand what is being asked and should not assume that simply because a particular question shares some key phrases with those from a past paper that the required answer is the same. Candidates must not answer a question by simply repeating phrases from the question.

General comments

If answers are crossed out, the new answers must be written clearly so that the text may be read easily and the correct mark awarded.

Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely helpful if this text is crossed out.

If typed answers are provided, it is very helpful if these are organised so that individual answers do not span page breaks. This is particularly important for pseudocode answers. If the question involves completing a table, they typed answers should clearly indicate any unfilled rows.

Familiarity with fundamental programming concepts is vital. Lack of understanding is often illustrated by the confusion between a literal value and an identifier, or the misuse of `OUTPUT` in place of `RETURN`. Many candidates lack knowledge of the use of parameters, often replacing parameters to a procedure or function with a series of prompts and inputs within the body of the subroutine.

The functions and operators that are available for use in pseudocode answers are described in the **insert** which accompanies the paper. Candidates must use these in their responses.

Comments on specific questions

It is recommended that the following specific comments be read in conjunction with the published mark scheme for this paper.

Question 1

(a) (i) Many candidates successfully answered this question, usually by stating that the rules or the grammar of the programming language had been followed or had not been broken. Other gave an example, but often the example was too vague, such as ‘contains no spelling mistakes’.

Some simply gave a textbook definition of a syntax error, without relating it to the given scenario.

Very few candidates addressed the fact that the program could run or be compiled/interpreted.

(ii) Many candidates fully answered this question, either my suggesting a logic error which 'causes the program to not produce the expected results' or a run-time which 'causes the program to crash'. Many incorrect answers were seen that bore little relation to the question. Many candidates did not attempt this question.

(b) This question was answered well by most candidates. Where a mistake was made, it was usually to suggest Integer for the first row.

(c) (i) The majority of candidates gained two or three marks. A common mistake was to suggest that the flight number was essential when making the booking.

(ii) Very few answers correctly described the benefit. Often the description added little to the wording from the stem and on many occasions the description related to the customer booking activity, such as suggesting that it would be 'easier to make a booking'. There are many incorrect techniques that bore little relation to the question. Many candidates did not attempt this question.

(iii) Most candidates were able to suggest two additional pieces of essential information. Departure date and arrival airport were the most common.

Question 2

(a) There was a good spread of marks across the range.
Many candidates recognised that this was a bubble sort. The use of two nested loops was referred to in many answers but often the detail was unclear. Strong answers described the decreasing number of comparisons required by the inner loop and the use of a 'no swaps' variable or similar to efficiently terminate the outer loop.

A detailed description of the swapping process was often seen.

Many solutions focused on finding the biggest number and storing in a different variable, rather than sorting the whole array.

(b) A wide range of answers was seen, with a significant number of perfect solutions.

Strong responses used a pre-condition loop (MP2) and usually gained the full six marks.
Candidates who opted for a post-condition loop were often unable to correctly alter the terminating condition shown in the flowchart (MP3).
Many weaker answers did not use a conditional loop but attempted to use an `IF ... ENDIF` structure.

MP4 was usually gained, provided the statements were inside some form of loop.

An invalid pseudocode statement (usually involving the keyword `CALL`) prevented many candidates from gaining MP5.

MP6 was straightforward and gained by many candidates. A small number decided to output the value 27, and others failed to gain this point as the call was placed in an incorrectly formed `IF ... ENDIF` clause.

Question 3

(a) (i) Many candidates referred to the increment of the End of Queue pointer and the positioning of the new data item at the corresponding position in the array.

Vague answers were common through imprecise language or poorly constructed answers such as stating that 'Octopus becomes the new End of Queue pointer'.

(ii) A small number of well-described answers was seen.

Answers which used the correct technical language such as 'The value Frog is assigned to the variable `AnimalName`' were rare.

A reference to the increment of the Front of Queue pointer to point to Cat was often missing.

Several answers confused the operation of queues with other ADTs.

Many candidates explained the FIFO mechanism without applying it to the given scenario.

(iii) The simple scenario describing the operation of the queue stated that the Front of Queue pointer points to the next item to be removed. This indicates that there will be always at least one item in the queue. (More complicated scenarios may be introduced in future questions, possibly including a count variable to indicate the number of items in the queue or a Boolean variable to indicate the queue is empty.)

As for previous parts, vague phrasing was commonplace, such as stating 'there is only one element in the queue'.

Many candidates did not attempt this question.

(b) (i) This question was generally well answered with many candidates scoring full marks. A small number of candidates made no attempt at this question.

(ii) A more complex 'trace' where the circular operation of the stack was involved. It was not uncommon for the queue pointers to be pointing at the correct values, but that these would be in the wrong array locations.

(c) MP2 was the mark most often gained for reference to the increment of the End of Queue pointer.

Very few answers addressed the other points, and little understanding of how a circular queue operates was evident.

A common mistake was to refer to 'searching for an empty location'.

Question 4

(a) Most candidates understood that it was boundary/extreme values and abnormal values (outside of the 0 to 40 range) that were required.

Candidates often repeated normal test data values.

Candidates need to carefully read the question. The rubric stated, 'The sensor produces ... an integer...'. Despite this, many candidates offered character data and real numbers as test data.

(b) Most candidates gained the marks for the 'Heaters on/Heaters off' actions by showing the inputs that would trigger the corresponding changes of state.

Many also gained the final mark for labelling the two 'no change of state' arrows with an input that would not result in a state change.

Question 5

(a) Generally, well answered with most candidates. An incorrect reason suggested by many was that the separator character was 'easy to see', rather than the fact that the character did not occur in the data

(b) Many answers correctly stated 'design'. Some answers hinted at the correct answer by suggesting terms such as 'planning' but the correct term as given in the syllabus.
Several answers simply repeated the word 'development' from the question, which asked for 'one stage from the program development life cycle'.

A significant number of candidates made no attempt at this question.

(c) Many candidates gained the mark (MP1) for the correct function heading. Several answers omitted the `ENDFUNCTION` statement which prevented this mark being given. The parameter and return type were usually given correctly.

Most candidates did not gain MP2. In many cases, the quotation marks around the literal value were omitted, so treating `LogFile` as an identifier rather than a literal string. Opening the file in `WRITE` mode rather than `APPEND` mode was common, as was omitting the `CLOSEFILE` statement, or not including the filename.

Most solutions contained a loop and this mark (MP4) was often given.
In some answers, the loop statement incorrectly omitted the assignment operator as illustrated:

```
FOR X 1 TO 2000
```

MP4 and MP5 presented the biggest challenge to most candidates, and only a small number of answers correctly addressing these points.

Many candidates gained MP6, often as a follow-through mark if either MP4 or MP5 were incorrect, and the error had been repeated. An occasional mistake seen was to write only the first six characters to the file, rather than the whole line.

Most solutions included an attempt at MP7 and in many cases the mark was given. It is encouraging to see the number of solutions that included the declaration and initialisation of the variable used to store the count value. A small number of solutions chose to use `OUTPUT` rather than `RETURN`.

A significant number of candidates made little or no attempt at a solution.

Question 6

(a) The solution required a simple algorithm with a loop which included a 2D array assignment. Of the five marks available, three were available for simple and straightforward elements of the solution.

A full range of marks was given. Many answers suggested that the problem had not been understood.

Most meaningful answers contained a count-controlled loop. A count range from 1 to `SetNum` was probably the most common, although others were seen. The marks for the loop were split between MP3 and MP4.

MP5 was the most challenging mark, as not only did the 2D array syntax have to be broadly correct, but also the value used for the column index had to work with the count range used. This mark was given only in a small number of responses.

Many candidates made little or no attempt at a solution.

(b) The full range of marks were given for answers to this question.

Most answers did not address the requirement to search either up or down as specified by the start column value, and so did not gain MP1 and MP2. The small number of solutions that did include this feature usually tested the start column using an `IF .. THEN ... ELSE ... ENDIF` clause which then selected one of two separate linear searches. More compact solutions contained a single search, where the start, end and step vales were defined with reference to the given start column.

MP3 and MP4 were commonly given.

Referencing an individual array element and testing its value (MP5) were included in many solutions.

For MP6, many solutions continued to search after an element with value 1 had been found. A secondary error was that often the saved column value would be overwritten during subsequent iterations.

A common approach to MP7 was to initialise the return value to –1 before the search. This was usually successful, except in the case described above when the correct column value was overwritten.

Few solutions gained MP8. A curious construct seen on several occasions involved placing an additional conditional loop around the count-controlled loop, to provide a mechanism to stop the iteration when the condition is satisfied. This would not be successful as the inner count-controlled loop would always have run to completion.

Many candidates made little or no attempt at a solution.

(c) Several candidates made a very good attempt at this final question.

Most realistic attempts gained the first mark (MP1).

The second mark (MP2) was not given due to treating the module `SearchInRow()` as a procedure rather than a function, for example:

`CALL SearchInRow(ThisRow, 1)` rather than `StartCol ← SearchInRow(ThisRow, 1)`

This mistake also meant that no return value was available.

Many solutions did not gain MP3 because the parameters to `SearchInRow()` were either missing or incorrect.

The correct calculation of the centre column was seen in many solutions (MP4), usually by adding the start and end values and dividing by 2. In many cases, this mark was given as a follow-through from errors relating to MP3. Some solutions calculated the difference between the start and end values but often these failed to then add the calculated value to the start value.

Most solutions did not convert the result of the calculations to an integer value for MP5.

As for MP1, most realistic attempts gained MP6. A small number of solutions chose to use `OUTPUT` rather than `RETURN`.

A significant number of candidates made little or no attempt at this question.

COMPUTER SCIENCE

Paper 9618/31
Advanced Theory

Key messages

Candidates are required to demonstrate a detailed study of the topics covered by the syllabus, ensuring they make good use of technical terminology as appropriate for this advanced theory paper. Candidates who have studied the relevant theory, practised, and made use of the relevant tools and techniques, are more likely to be able to solve the problems set on the examination paper.

Candidates must respond to each question after carefully reading it to make sure they understand what is required.

Candidates must answer each question in an appropriate manner for the command word of the question; for example, a question beginning with 'explain' requires more detail than a question beginning with 'identify'. If a question asks for working to be shown, candidates must ensure that they do this, to gain maximum marks.

Candidates are always advised to answer questions in the context of any scenario described in the question, rather than in generic terms.

Candidates are also advised to use the published pseudocode guide when preparing for this examination, for example, in the areas of user-defined data types or algorithm construction, and answer questions requiring pseudocode answers using this preferred syntax.

General comments

Many candidates demonstrated good knowledge of the process of packet switching and the use of the router within this process.

Candidates demonstrated the ability to represent a sum of products taken from a truth table in the form of a Karnaugh map and then went on to write a simplified sum-of-products.

Most candidates demonstrated knowledge of the application of stacks in relation to recursion.

Comments on specific questions

Question 1

(a) (i) Most candidates recognised that the binary point needed to be moved from the location given in the binary number in the question, to give a mantissa, and that the exponent would show how many places the binary point would need to be moved to recreate the original number. Candidates who recognised that the given binary number would be represented as a positive number generally achieved both marks.

(ii) Candidates who recognised that the given binary number had too many bits for it to be stored in the stated computer system, therefore resulting in loss of precision, achieved marks.

(b) A wide range of answers was seen for this question with candidates recognising various reasons for storing numbers in normalised form; including, to maximise the range of numbers for the number of bits available, to minimise the number of redundant leading zeros, or to avoid the possibility of numbers being stored using multiple representations.

Question 2

Candidates were generally able to match most of the stated programming paradigms with their appropriate descriptions.

Question 3

(a) Candidates were mostly able to gain some credit for writing the correct content for a given enumerated data type, however, some responses incorrectly used the statement `DECLARE` for the definition of the data type.

(b) Some candidates correctly applied the pseudocode syntax of a pointer non-composite data type. However, for some responses, the data type referenced was not `SelectParts` as given in the question.

Question 4

(a) Candidates were generally able to partially demonstrate how to represent a syntax diagram using Backus-Naur Form (BNF) by showing the content of a particular syntax diagram. However, errors were seen with the application of fully correct BNF syntax.

(b) (i) The vast majority of candidates demonstrated their ability to supply an example of a valid password within the constraints of the given syntax diagrams.

(ii) Candidates who recognised that two separate BNF expressions were required to solve this problem, so that the second expression was called by the first, scored high marks. A small number of candidates were able to achieve this.

Question 5

(a) Candidates were generally able to differentiate between the sequential and serial methods of file access with the full range of possible marks seen for this question. Candidates must be careful, when answering this type of question, to state that it is the records that are stored one after the other, rather than the files that are stored one after the other.

(b) Most candidates recognised that the application described in the question was most suited to the direct method of file access.

(c) Most candidates recognised that the application described in the question was most suited to the sequential method of file access.

Question 6

(a) The vast majority of candidates gained some marks for explaining how packet switching is used to transfer messages across the internet, with many of these candidates achieving high marks.

(b) Most candidates also gained marks for outlining the function of the router in packet switching, however, generic points, such as the router connects to the internet, or it helps packets find its destination, were not specific enough.

Question 7

(a) The vast majority of candidates were able to write a Boolean expression corresponding to the given truth table as a sum-of-products. Some candidates had simplified their answers when the sum-of-products was required.

(b) (i) Candidates were generally able to populate the given Karnaugh map (K-map) diagram to correctly represent the truth table in **part (a)**. Some candidates did not fully complete the K-map with zeros and ones or used some other notation. Marks were only awarded where the correct complete notation of zeros and ones was given.

(ii) The vast majority of candidates were able to demonstrate appropriate loops around groups of ones in their K-maps, so that an optimal sum-of-products could be produced.

(iii) Most candidates provided an appropriate simplified sum-of-products corresponding to their K-map. Some candidates provided a simplification to the Boolean expression rather than the sum-of-products.

(iv) Candidates who had provided a correct answer to **part (b)(iii)** were generally able to go an extra step and provide the simplified Boolean expression for their answer.

Question 8

(a) Candidates who were aware that the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols provide communications security over the internet and encryption gained credit for this question. Most candidates gave one of these facts, with some giving both.

(b) Most candidates were able to at least partially explain how the SSL/TLS protocols are used when client-server communication is initiated and some comprehensive answers were seen. Some candidates provided responses where the various stages of the process were in the wrong order.

Question 9

(a) (i) Many candidates were aware that having multiple hidden layers in an artificial neural network enables deep learning to take place.

(ii) The vast majority of candidates were able to provide at last a partial explanation of how artificial neural networks enable machine learning. Answers that simply stated or described supervised and unsupervised machine learning were not sufficient to answer this question.

(b) Most candidates were able to demonstrate the method by which the shortest path between two points may be determined using the A* algorithm. A significant number of candidates gave fully correct answers.

Question 10

(a) Candidates were generally able to state at least one essential feature of recursion, with many candidates going on to state two or three.

(b) The vast majority of candidates recognised that as the stack is a Last In First Out (LIFO) Abstract Data Type (ADT), it's method of operation closely matches the process of recursion. Many answers dwelt on this aspect, rather than going deeper into the explanation to talk about recursive calls being pushed onto the stack and then being popped off the stack, as the function ends, thereby limiting their marks.

(c) Candidates demonstrated a good knowledge of types of ADT, with the vast majority naming the queue and linked list. A few candidates alternatively named the binary tree as an example of an ADT.

(d) Many candidates successfully completed the algorithm for the given function. Candidates who made use of the information given, such as variable names and the function `StackFull()` achieved the highest marks. Candidates whose answers attempted to re-write code that was already given, or who outputted an error message, rather than returning it, tended to score fewer marks.

COMPUTER SCIENCE

Paper 9618/32
Advanced Theory

Key messages

Candidates are required to demonstrate a detailed study of the topics covered by the syllabus, ensuring they make good use of technical terminology as appropriate for this advanced theory paper. Candidates who have studied the relevant theory, practised, and made use of the relevant tools and techniques, are more likely to be able to solve the problems set on the examination paper.

Candidates must respond to each question after carefully reading it to make sure they understand what is required.

Candidates must answer each question in an appropriate manner for the command word of the question; for example, a question beginning with 'explain' requires more detail than a question beginning with 'identify'. If a question asks for working to be shown, candidates must ensure that they do this, to gain maximum marks.

Candidates are always advised to answer questions in the context of any scenario described in the question, rather than in generic terms.

Candidates are also advised to use the published pseudocode guide when preparing for this examination, for example, in the areas of user-defined data types or algorithm construction, and answer questions requiring pseudocode answers using this preferred syntax.

General comments

Many candidates demonstrated good knowledge of the process of packet switching and the use of the router within this process.

Candidates demonstrated the ability to represent a sum of products taken from a truth table in the form of a Karnaugh map and then went on to write a simplified sum-of-products.

Most candidates demonstrated knowledge of the application of stacks in relation to recursion.

Comments on specific questions

Question 1

(a) (i) Most candidates recognised that the binary point needed to be moved from the location given in the binary number in the question, to give a mantissa, and that the exponent would show how many places the binary point would need to be moved to recreate the original number. Candidates who recognised that the given binary number would be represented as a positive number generally achieved both marks.

(ii) Candidates who recognised that the given binary number had too many bits for it to be stored in the stated computer system, therefore resulting in loss of precision, achieved marks.

(b) A wide range of answers was seen for this question with candidates recognising various reasons for storing numbers in normalised form; including, to maximise the range of numbers for the number of bits available, to minimise the number of redundant leading zeros, or to avoid the possibility of numbers being stored using multiple representations.

Question 2

Candidates were generally able to match most of the stated programming paradigms with their appropriate descriptions.

Question 3

(a) Candidates were mostly able to gain some credit for writing the correct content for a given enumerated data type, however, some responses incorrectly used the statement `DECLARE` for the definition of the data type.

(b) Some candidates correctly applied the pseudocode syntax of a pointer non-composite data type. However, for some responses, the data type referenced was not `SelectParts` as given in the question.

Question 4

(a) Candidates were generally able to partially demonstrate how to represent a syntax diagram using Backus-Naur Form (BNF) by showing the content of a particular syntax diagram. However, errors were seen with the application of fully correct BNF syntax.

(b) (i) The vast majority of candidates demonstrated their ability to supply an example of a valid password within the constraints of the given syntax diagrams.

(ii) Candidates who recognised that two separate BNF expressions were required to solve this problem, so that the second expression was called by the first, scored high marks. A small number of candidates were able to achieve this.

Question 5

(a) Candidates were generally able to differentiate between the sequential and serial methods of file access with the full range of possible marks seen for this question. Candidates must be careful, when answering this type of question, to state that it is the records that are stored one after the other, rather than the files that are stored one after the other.

(b) Most candidates recognised that the application described in the question was most suited to the direct method of file access.

(c) Most candidates recognised that the application described in the question was most suited to the sequential method of file access.

Question 6

(a) The vast majority of candidates gained some marks for explaining how packet switching is used to transfer messages across the internet, with many of these candidates achieving high marks.

(b) Most candidates also gained marks for outlining the function of the router in packet switching, however, generic points, such as the router connects to the internet, or it helps packets find its destination, were not specific enough.

Question 7

(a) The vast majority of candidates were able to write a Boolean expression corresponding to the given truth table as a sum-of-products. Some candidates had simplified their answers when the sum-of-products was required.

(b) (i) Candidates were generally able to populate the given Karnaugh map (K-map) diagram to correctly represent the truth table in **part (a)**. Some candidates did not fully complete the K-map with zeros and ones or used some other notation. Marks were only awarded where the correct complete notation of zeros and ones was given.

(ii) The vast majority of candidates were able to demonstrate appropriate loops around groups of ones in their K-maps, so that an optimal sum-of-products could be produced.

(iii) Most candidates provided an appropriate simplified sum-of-products corresponding to their K-map. Some candidates provided a simplification to the Boolean expression rather than the sum-of-products.

(iv) Candidates who had provided a correct answer to **part (b)(iii)** were generally able to go an extra step and provide the simplified Boolean expression for their answer.

Question 8

(a) Candidates who were aware that the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols provide communications security over the internet and encryption gained credit for this question. Most candidates gave one of these facts, with some giving both.

(b) Most candidates were able to at least partially explain how the SSL/TLS protocols are used when client-server communication is initiated and some comprehensive answers were seen. Some candidates provided responses where the various stages of the process were in the wrong order.

Question 9

(a) (i) Many candidates were aware that having multiple hidden layers in an artificial neural network enables deep learning to take place.

(ii) The vast majority of candidates were able to provide at last a partial explanation of how artificial neural networks enable machine learning. Answers that simply stated or described supervised and unsupervised machine learning were not sufficient to answer this question.

(b) Most candidates were able to demonstrate the method by which the shortest path between two points may be determined using the A* algorithm. A significant number of candidates gave fully correct answers.

Question 10

(a) Candidates were generally able to state at least one essential feature of recursion, with many candidates going on to state two or three.

(b) The vast majority of candidates recognised that as the stack is a Last In First Out (LIFO) Abstract Data Type (ADT), it's method of operation closely matches the process of recursion. Many answers dwelt on this aspect, rather than going deeper into the explanation to talk about recursive calls being pushed onto the stack and then being popped off the stack, as the function ends, thereby limiting their marks.

(c) Candidates demonstrated a good knowledge of types of ADT, with the vast majority naming the queue and linked list. A few candidates alternatively named the binary tree as an example of an ADT.

(d) Many candidates successfully completed the algorithm for the given function. Candidates who made use of the information given, such as variable names and the function `StackFull()` achieved the highest marks. Candidates whose answers attempted to re-write code that was already given, or who outputted an error message, rather than returning it, tended to score fewer marks.

COMPUTER SCIENCE

Paper 9618/41
Practical

Key messages

This practical exam required candidates to write solutions in only VB.NET, Python or Java. Candidates need to be using suitable software to write their programs that allows them to test their solutions.

When completing the evidence documents, candidates need to copy the code into the document, they should not be taking a screenshot of this code. The program code needs to clearly visible. Many responses had coloured code on a black background with a poor resolution, which meant that some statements could not be read. When taking screenshots of the results, candidates need to make sure that the images are large enough, and have sufficient resolution, to be easily read. Where possible, these should be either black text on white, or white text on black. Coloured outputs should not be produced because these can be more difficult to read. Where the results of a test required a large amount of content, multiple smaller screenshots are often more appropriate because they can be enlarged and viewed easier.

Candidates should read the whole question, including all parts, before starting to produce their response. This will allow candidates to determine the full requirements, and they may make some different decisions that mean they do not need to amend code in a later part. If a candidate produces a solution to one part, then changes this code later, they can amend their code in the evidence document.

Candidates should be testing their solutions throughout the development of each question part, even where evidence of testing is not required. Testing evidence is only required at specific points, but it is expected that candidates will be testing with a range of data at each stage.

When candidates are given an algorithm in pseudocode, they need to consider how it can be written in their own language without changing the logic of the algorithm. The algorithms are generic and written without a specific language as its focus, for example, the passing of parameters by reference is a specification requirement but is only directly possible in VB.NET. Therefore, candidates need will need to amend the generic algorithms to work in their language. These changes should not be fundamental to the algorithm, for example, if a bubble sort is given, then this bubble sort algorithm should be retained and not rewritten to a different sorting algorithm, or a different way of doing the same sort.

General comments

There were a range of approaches to the questions in the paper.

Candidates found the conversion of a recursive algorithm to an iterative one challenging, with many retaining the recursive call in their solution.

Candidates need to make sure they are reading the requirements of each algorithm carefully and should check their solution meets each of these on completion. On some questions, candidates were required to include specific outputs that were missed in the solutions.

There were some strong examples of object-oriented programming with many candidates being able to declare classes and constructors. Fewer were able to then make use of these classes, specifically the get and set method(s) declared.

There were few attempts at exception handling when reading from the text file. Candidates must be able to detect and catch exceptions in their chosen language, giving an appropriate output (or equivalent) when caught.

Comments on specific questions

Question 1

(a) This question required candidates to convert the given algorithm into their chosen programming language. Many candidates were able to convert the structure of the code appropriately. The weaker responses did not calculate the integer division, instead using standard division. Some candidates used standard division but then rounded down or converted to an integer which produced the same outcome. When candidates are asked to convert pseudocode to their language, they should follow the structure of the code. Changes can be made to tailor it to their own programming language requirements, but the fundamental principles and operations of the algorithm need to be retained without additional features, for example, some candidates implemented additional output statements in their solution, and some candidates returned values at different positions within the algorithm.

(b) (i) Many candidates successfully called their function with the appropriate parameters. Fewer candidates met the requirements to output the value of the two parameters and to then output the return value of each.

(ii) The stronger responses had outputs for each set of parameters, the correct outputs from the function and the final return value. Some responses, where they did not output the returned value in **part (b)(i),** were missing the final value from each function call.

(c) Candidates found this question challenging, with many rewriting the same algorithm but changing the identifier, hence retaining the recursive calls. The stronger responses were able to identify the need for a loop and use this appropriately. Some candidates used a single loop, whilst others used additional loops within each selection section to gain the same functionality.

(d) (i) The stronger responses met all requirements by calling the new function, along with the required outputs. Some candidates did not call their new function from **part (c)**, instead calling the original recursive function.

(ii) This question asked for the output of both functions for each set of parameters. This required the outputs for the recursive and iterative functions. Few candidates met this requirement, with many only providing the solution for one function.

Question 2

(a) This question required candidates to declare a class and constructor in their chosen language. The question required candidates to include attribute declarations using comments if they used Python, because these are not required to function but are required in this answer for candidates to demonstrate their understanding of declarations and data types. In this case, these declarations could be separate to the initialisations, or comments included with each initialisation to identify the data type, and in some cases, to indicate that they are intended to be private.

Many candidates were able to accurately declare a class, but fewer could use their language specific constructor, with some candidates creating methods with the identifier constructor instead. The question specified that attributes were required to be private, and many candidates were able to do this successfully.

When declaring a constructor, candidates must make sure they are meeting the requirements for the parameters and initialisation, in this case, using the four parameters. Some candidates initialised the attributes to numbers, or null values and others attempted to read in data from the user.

(b) Many candidates were able to demonstrate a good understanding of the purpose and function of get methods. They were able to declare the methods appropriately without parameters and returned the correct attribute. Some weaker responses sent parameters to the function, and some returned this parameter or read in an input and returned this value. Candidates using Python need to make sure they are including ‘self’ as a parameter when writing methods within a class.

(c) There were a mix of responses to this question. The stronger responses were able to write a set method that took a parameter and correctly assigned this to the methods. Some of the weaker responses had the assignment statement the wrong way around, assigning the attribute to the parameter instead of the parameter to the attributes. Some candidates attempted to read in a value from the user and assign this instead of using a parameter.

(d) This question required candidates to declare an array of a set number of elements of type `Picture` class. The stronger responses declared arrays, or lists in Python, that had this number of elements. In some languages, they were declared with this number of values. In Python, candidates often looped 100 times to append that quantity of elements to a list.

The weaker candidates did not identify that the elements would be of type `Picture` class. In Python, stronger candidates either declared 100 empty objects and appended these to the array or used a comment to indicate the data type intended.

A common error was declaring the array with the identifier Picture, instead of the array being of type `Picture`.

(e) There were a mix of responses to this question. The question required candidates to make use of the class and methods they had declared to read the data from the text file, declare `Picture` objects and assign them to the array they declared in **part (d)**. Some candidates wrote a method for the class instead of a function within the program.

Many candidates were able to open the text file and read in some of the data. Fewer candidates were able to read in each set of 4 lines and use these to create an instance of an object of type `Picture`. Some responses read each line one at a time and until the end of the file was reached, and then attempted to create an object.

There were a range of approached to reading in the data, some candidates used a loop that read in 4 lines repeatedly, whilst others read in all of the data into a structure such as an array, and then looped through this structure to declared the objects.

Few candidates closed the text file in an appropriate place, i.e. once all the data had been read.

The question asked candidates to raise an exception if the file could not be found. Few candidates attempted to write an exception, those that did often did this accurately, giving a suitable output if the file could not be found.

(f) Many candidates were able to call the appropriate function. Fewer candidates were able to identify the need to store or use the return value from the function call.

(g) This question required an algorithm to search the objects declared in **part (e)** for a picture that met the colour and did not exceed the maximum width and height all entered by the user.

Many candidates were able to take the values as input, but fewer converted the colour to allow any case to be entered.

The stronger responses looped through the array that was returned from their function call in **part (f)**, using the get methods they have declared previously. The weaker responses attempted to access the attributes directly or did not make use of the array to search through each element serially.

A common error was the comparison of the height and width, where candidates checked if the input was ‘<’ the width and height, instead of ‘<=’.

Some candidates output the data for all the pictures that met the requirements, whilst others only output one. Either of these were acceptable.

(h) This question required candidates to test their solutions with the given search criteria. This required the program to be run twice, and hence two screenshots produced. Few candidates had the correct output for both criteria, but more output that there were not results for silver. Some screenshots did

not show the data being input by the user, candidates must make sure that all inputs and outputs are shown clearly.

Question 3

(a) Many candidates were able to assign the identified values to the two pointers. Fewer candidates declared a 2-dimensional array with the appropriate elements in each dimension.

(b) This question required candidates to complete the algorithm and write it in their chosen programming language. This algorithm made use of passing parameters by reference. Candidates should understand how their chosen language can work in this scenario, and they may need to adapt the algorithm to make sure the appropriate values are amended in the program where it is called. This should not require significant changes to the structure and sequence within the algorithm. In this example, there were a variety of solutions that were implemented:

- candidates who wrote in VB.NET could pass the parameters by reference
- some responses declared the array and pointers as having global scope
- some responses in Python returned the array and both pointers and then overwrote these when called
- some responses in Java declared a class and an instance for these parameters which was then returned.

This list is not exhaustive but demonstrates some of the ways candidates adapted the algorithm for their language without changing its function.

Candidates should compare their final solution with the given algorithm to make sure that they have correctly used each statement. Some responses did not include some parts of the inner selection statements.

(c) Candidates found this question challenging, with many responses not using the 2 dimensions accurately. The stronger responses were able to isolate each element and output them on the same line, but fewer included the required spaces between values.

(d) (i) Many candidates were able to write an appropriate loop that repeated 10 times and some candidates called the function defined in **part (b)**. Fewer candidates were able to pass the parameters that followed on from their previous solution to create a working solution, for example, values were returned but not stored when returned, or parameters were given in their response to **part (b)** and not included in the calls. The stronger responses had consistent solutions that worked with their previous answer.

(ii) Many responses had not accurately added the data to the array, with pointers and the array not being updated, and therefore the values were not stored accurately in tree.

(e) (i) The stronger responses were able to write a traversal that attempted to access the data in the correct order. Some of these responses also checked whether each node was empty being (or after) calling depending on the design of their algorithm, which meant these solutions would run indefinitely.

(ii) Many candidates did not give a response to this question. Some candidates were able to accurately output the values from the tree in the correct order.

COMPUTER SCIENCE

Paper 9618/42
Practical

Key messages

This practical exam required candidates to write solutions in only VB.NET, Python or Java. Candidates need to be using suitable software to write their programs that allows them to test their solutions.

When completing the evidence documents, candidates need to copy the code into the document, they should not be taking a screenshot of this code. The program code needs to clearly visible. Many responses had coloured code on a black background with a poor resolution, which meant that some statements could not be read. When taking screenshots of the results, candidates need to make sure that the images are large enough, and have sufficient resolution, to be easily read. Where possible, these should be either black text on white, or white text on black. Coloured outputs should not be produced because these can be more difficult to read. Where the results of a test required a large amount of content, multiple smaller screenshots are often more appropriate because they can be enlarged and viewed easier.

Candidates should read the whole question, including all parts, before starting to produce their response. This will allow candidates to determine the full requirements, and they may make some different decisions that mean they do not need to amend code in a later part. If a candidate produces a solution to one part, then changes this code later, they can amend their code in the evidence document.

Candidates should be testing their solutions throughout the development of each question part, even where evidence of testing is not required. Testing evidence is only required at specific points, but it is expected that candidates will be testing with a range of data at each stage.

When candidates are given an algorithm in pseudocode, they need to consider how it can be written in their own language without changing the logic of the algorithm. The algorithms are generic and written without a specific language as its focus, for example, the passing of parameters by reference is a specification requirement but is only directly possible in VB.NET. Therefore, candidates need will need to amend the generic algorithms to work in their language. These changes should not be fundamental to the algorithm, for example, if a bubble sort is given, then this bubble sort algorithm should be retained and not rewritten to a different sorting algorithm, or a different way of doing the same sort.

General comments

There were a range of approaches to the questions in the paper.

Candidates found the conversion of a recursive algorithm to an iterative one challenging, with many retaining the recursive call in their solution.

Candidates need to make sure they are reading the requirements of each algorithm carefully and should check their solution meets each of these on completion. On some questions, candidates were required to include specific outputs that were missed in the solutions.

There were some strong examples of object-oriented programming with many candidates being able to declare classes and constructors. Fewer were able to then make use of these classes, specifically the get and set method(s) declared.

There were few attempts at exception handling when reading from the text file. Candidates must be able to detect and catch exceptions in their chosen language, giving an appropriate output (or equivalent) when caught.

Comments on specific questions

Question 1

(a) This question required candidates to convert the given algorithm into their chosen programming language. Many candidates were able to convert the structure of the code appropriately. The weaker responses did not calculate the integer division, instead using standard division. Some candidates used standard division but then rounded down or converted to an integer which produced the same outcome. When candidates are asked to convert pseudocode to their language, they should follow the structure of the code. Changes can be made to tailor it to their own programming language requirements, but the fundamental principles and operations of the algorithm need to be retained without additional features, for example, some candidates implemented additional output statements in their solution, and some candidates returned values at different positions within the algorithm.

(b) (i) Many candidates successfully called their function with the appropriate parameters. Fewer candidates met the requirements to output the value of the two parameters and to then output the return value of each.

(ii) The stronger responses had outputs for each set of parameters, the correct outputs from the function and the final return value. Some responses, where they did not output the returned value in **part (b)(i),** were missing the final value from each function call.

(c) Candidates found this question challenging, with many rewriting the same algorithm but changing the identifier, hence retaining the recursive calls. The stronger responses were able to identify the need for a loop and use this appropriately. Some candidates used a single loop, whilst others used additional loops within each selection section to gain the same functionality.

(d) (i) The stronger responses met all requirements by calling the new function, along with the required outputs. Some candidates did not call their new function from **part (c)**, instead calling the original recursive function.

(ii) This question asked for the output of both functions for each set of parameters. This required the outputs for the recursive and iterative functions. Few candidates met this requirement, with many only providing the solution for one function.

Question 2

(a) This question required candidates to declare a class and constructor in their chosen language. The question required candidates to include attribute declarations using comments if they used Python, because these are not required to function but are required in this answer for candidates to demonstrate their understanding of declarations and data types. In this case, these declarations could be separate to the initialisations, or comments included with each initialisation to identify the data type, and in some cases, to indicate that they are intended to be private.

Many candidates were able to accurately declare a class, but fewer could use their language specific constructor, with some candidates creating methods with the identifier constructor instead. The question specified that attributes were required to be private, and many candidates were able to do this successfully.

When declaring a constructor, candidates must make sure they are meeting the requirements for the parameters and initialisation, in this case, using the four parameters. Some candidates initialised the attributes to numbers, or null values and others attempted to read in data from the user.

(b) Many candidates were able to demonstrate a good understanding of the purpose and function of get methods. They were able to declare the methods appropriately without parameters and returned the correct attribute. Some weaker responses sent parameters to the function, and some returned this parameter or read in an input and returned this value. Candidates using Python need to make sure they are including ‘self’ as a parameter when writing methods within a class.

(c) There were a mix of responses to this question. The stronger responses were able to write a set method that took a parameter and correctly assigned this to the methods. Some of the weaker responses had the assignment statement the wrong way around, assigning the attribute to the parameter instead of the parameter to the attributes. Some candidates attempted to read in a value from the user and assign this instead of using a parameter.

(d) This question required candidates to declare an array of a set number of elements of type `Picture` class. The stronger responses declared arrays, or lists in Python, that had this number of elements. In some languages, they were declared with this number of values. In Python, candidates often looped 100 times to append that quantity of elements to a list.

The weaker candidates did not identify that the elements would be of type `Picture` class. In Python, stronger candidates either declared 100 empty objects and appended these to the array or used a comment to indicate the data type intended.

A common error was declaring the array with the identifier Picture, instead of the array being of type `Picture`.

(e) There were a mix of responses to this question. The question required candidates to make use of the class and methods they had declared to read the data from the text file, declare `Picture` objects and assign them to the array they declared in **part (d)**. Some candidates wrote a method for the class instead of a function within the program.

Many candidates were able to open the text file and read in some of the data. Fewer candidates were able to read in each set of 4 lines and use these to create an instance of an object of type `Picture`. Some responses read each line one at a time and until the end of the file was reached, and then attempted to create an object.

There were a range of approached to reading in the data, some candidates used a loop that read in 4 lines repeatedly, whilst others read in all of the data into a structure such as an array, and then looped through this structure to declared the objects.

Few candidates closed the text file in an appropriate place, i.e. once all the data had been read.

The question asked candidates to raise an exception if the file could not be found. Few candidates attempted to write an exception, those that did often did this accurately, giving a suitable output if the file could not be found.

(f) Many candidates were able to call the appropriate function. Fewer candidates were able to identify the need to store or use the return value from the function call.

(g) This question required an algorithm to search the objects declared in **part (e)** for a picture that met the colour and did not exceed the maximum width and height all entered by the user.

Many candidates were able to take the values as input, but fewer converted the colour to allow any case to be entered.

The stronger responses looped through the array that was returned from their function call in **part (f)**, using the get methods they have declared previously. The weaker responses attempted to access the attributes directly or did not make use of the array to search through each element serially.

A common error was the comparison of the height and width, where candidates checked if the input was ‘<’ the width and height, instead of ‘<=’.

Some candidates output the data for all the pictures that met the requirements, whilst others only output one. Either of these were acceptable.

(h) This question required candidates to test their solutions with the given search criteria. This required the program to be run twice, and hence two screenshots produced. Few candidates had the correct output for both criteria, but more output that there were not results for silver. Some screenshots did

not show the data being input by the user, candidates must make sure that all inputs and outputs are shown clearly.

Question 3

(a) Many candidates were able to assign the identified values to the two pointers. Fewer candidates declared a 2-dimensional array with the appropriate elements in each dimension.

(b) This question required candidates to complete the algorithm and write it in their chosen programming language. This algorithm made use of passing parameters by reference. Candidates should understand how their chosen language can work in this scenario, and they may need to adapt the algorithm to make sure the appropriate values are amended in the program where it is called. This should not require significant changes to the structure and sequence within the algorithm. In this example, there were a variety of solutions that were implemented:

- candidates who wrote in VB.NET could pass the parameters by reference
- some responses declared the array and pointers as having global scope
- some responses in Python returned the array and both pointers and then overwrote these when called
- some responses in Java declared a class and an instance for these parameters which was then returned.

This list is not exhaustive but demonstrates some of the ways candidates adapted the algorithm for their language without changing its function.

Candidates should compare their final solution with the given algorithm to make sure that they have correctly used each statement. Some responses did not include some parts of the inner selection statements.

(c) Candidates found this question challenging, with many responses not using the 2 dimensions accurately. The stronger responses were able to isolate each element and output them on the same line, but fewer included the required spaces between values.

(d) (i) Many candidates were able to write an appropriate loop that repeated 10 times and some candidates called the function defined in **part (b)**. Fewer candidates were able to pass the parameters that followed on from their previous solution to create a working solution, for example, values were returned but not stored when returned, or parameters were given in their response to **part (b)** and not included in the calls. The stronger responses had consistent solutions that worked with their previous answer.

(ii) Many responses had not accurately added the data to the array, with pointers and the array not being updated, and therefore the values were not stored accurately in tree.

(e) (i) The stronger responses were able to write a traversal that attempted to access the data in the correct order. Some of these responses also checked whether each node was empty being (or after) calling depending on the design of their algorithm, which meant these solutions would run indefinitely.

(ii) Many candidates did not give a response to this question. Some candidates were able to accurately output the values from the tree in the correct order.