COMPUTER SCIENCE

Paper 9608/11
Written Paper

Key messages

Candidates need to use precise terms related to the subject at this level of study.

Each question needs to be read carefully. A question that begins with 'Describe' requires a different type of answer from one that begins with 'Explain'.

Candidates are required to respond appropriately in the given context of the question. Generic answers that apply to any situation will not gain full credit.

General comments

In general, candidates performed well on the low level language trace table, as well as on the databases question. Many candidates found the questions about video encoding and program libraries more challenging.

Some candidates write their answer first in pencil and then overwrote this in ink. Candidates must ensure that the pencil marks are completely removed.

Comments on specific questions

Question 1

(a) Many candidates correctly identified that the first address was invalid. Some candidates found it more challenging to give the reason for this. The most common answer was that the address was written in Hexadecimal, without any reference to H being an invalid Hexadecimal digit. Almost all candidates recognised that the second address was also invalid and correctly stated that the value 299 was out of the accepted range. Most candidates recognised that the third address was a valid IPv4 address; explanations of why this was valid were sometimes very vague. An explanation such as, 'because it fits the IPv4 format' are not enough for credit at this level of study.

(b) There were a small number of very good answers to this question. Many candidates need to improve their understanding of how a URL is converted into the corresponding IP address. There was a general misconception that the URL is sent to the Domain Name Service, rather than that the URL is first parsed to obtain the Domain Name.

(c) The majority of candidates answered this question correctly.

(d) (i) Almost all candidates appreciated that a recording of a concert would be a very large file.

(ii) The majority of candidates answered this question correctly.

(iii) There were a number of good answers to this question. Some candidates need to improve their understanding of lossy and lossless compression. The question begins with, 'Explain why lossy compression is more appropriate'. Many candidates need to understand that in order to answer this question each statement needs to refer to both types of compression. An example of a good answer is, 'The recording of the concert is a large file and so needs significant reduction in size, lossy compression creates a significantly smaller file than lossy compression'.

Question 2

(a) There were some good answers to this question. Many candidates need to improve their understanding of the differences between a Free Software licence and an Open Source licence.

(b) (i) Many candidates found this question challenging. Some candidates showed a good understanding of the limitations of a file based approach to storing data. There was some confusion with the storage of files on an external device, with descriptions of directory structures and file access techniques.

(ii) A minority of candidates answered this question well. Candidates need to improve their understanding of database schema.

(c) (i) Many candidates correctly identified the type of relationship.

(ii) Many candidates correctly described the implementation of the relationship. The most common error was to give a generic answer rather than an answer which used the table definitions given in the question.

(iii) There were some excellent answers to this question. The most common reason for error here was the omission of the `GROUP BY` clause. Some candidates need to improve their understanding of basic SQL programming.

Question 3

(a) (i) The question begins with the words 'Describe two advantages', so generic statements about library files do not answer the question. Candidates should understand that to answer this type of question it is insufficient to say, for example, 'Library files will have been used by a lot of people so will be well-tested', there needs to be a statement saying why this is an advantage. An example of a good answer would be, 'Library files will have been used by a large number of people so will be well-tested, which means that the program will be more robust'.

(ii) A minority of candidates answered this question well. Many answers repeated statements from the previous question with no reference to a Dynamic Link Library (DLL). Candidates need to improve their understanding of a DLL.

(b) (i) The majority of candidates correctly stated one benefit.

(ii) Many candidates were able to state that a compiler produces an executable file. Fewer candidates were able to correctly give two more benefits. Many answers stated how a compiler operates rather than saying why it would be used when the program was complete. There was some confusion between the use of a compiler and the use of an interpreter. A number of responses incorrectly stated that it was faster to run the compiler rather than that the executable file would probably run faster than interpreting the source code.

Question 4

(a) Almost all candidates were able to name three other principles of the ACM/IEEE Software Engineering Code of Ethics. Describing the principles was more challenging. Candidates need to be aware that if a question says, for example, 'Describe **three other** principles'; credit will not be given for including the example given in the question.

(b) Almost all candidates correctly explained how data backup could aid recovery of data. Many candidates found explaining disk-mirroring much more challenging. Many responses did not differentiate between backup and disk-mirroring. Candidates need to improve their understanding that disk-mirroring is the simultaneous use of two or more disks.

Question 5

(a) There were a small number of correct answers to this question. Many candidates need to improve their understanding of the operation of a two-pass assembler.

(b) There were a small number of excellent answers to this part question. Many candidates need to improve their understanding of processor instruction groups.

(c) Tracing of assembly language programs is improving steadily and this one was generally very well done, with many candidates providing completely correct solutions, neatly set out in the table provided. The most common error was the inclusion of the word `END` in the `OUTPUT` column after the final statement.

Question 6

(a) There were some very good answers to this question. A minority of candidates confused interlaced encoding with progressive encoding and vice versa. Some candidates need to improve their understanding of these two different methods.

(b) Many candidates need to understand that in order to answer this question; each statement needs to refer to both types of encoding. An example of a good answer is: 'Interlaced encoding halves the bandwidth requirements compared to progressive encoding which needs a high bandwidth'.

(c) There were a few very good answers to this question. Many candidates confused spatial and temporal redundancy and some candidates used some irrelevant statements about run length encoding. Most candidates need to improve their understanding of this topic.

(d) (i) The majority of candidates wrote that an analogue to digital converter would be used; few candidates described further how the analogue waveform would be converted to digital data. The question begins with the word 'Describe', so candidates need to include sufficient detail in their answers. A single statement is insufficient for full credit.

(ii) This question asks specifically for the effects of the sampling rate and sampling resolution on the size of the file. Candidates need to be aware that their answers should reflect this. While there were a few excellent answers to this question, many explanations gave great detail about sampling rate and sampling resolution, then omitted any reference to the effect that changing these would have on the size of the file

COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

Candidates need to use precise terms related to the subject at this level of study.

Each question needs to be read carefully. A question that begins with 'Describe' requires a different type of answer from one that begins with 'Explain'.

Candidates are required to respond appropriately in the given context of the question. Generic answers that apply to any situation will not gain full credit.

General comments

In general, candidates performed well on low level languages question, as well as the question on logic gates and truth tables. Many candidates found the questions about hardware and Database Management Systems (DBMS) more challenging.

Some candidates write their answer first in pencil and then over-write in ink. These candidates must ensure that the pencil marks are completely removed.

Comments on specific questions

Question 1

(a) (i) There were many correct answers to this question. The majority of candidates were able to give a valid IPv4 address. The most common incorrect response either used a colon as a separator, or included values above 255.

(ii) This question asked 'why' there is a need for IPv6 addressing. Many of the answers did not answer this question, but described IPv6 addressing in general terms. Some candidates need to understand that IPv6 is required because IPv4 addresses are running out.

(iii) Most candidates recognised that the address given would be invalid as an IPv4 address because the value in each group of numbers was out of range. Some candidates need to understand that the maximum value used in IPv4 addressing is 255, not 256. A common incorrect statement was that IPv4 addresses cannot be represented in Hexadecimal notation.

(b) (i) A minority of candidates answered this question well. Candidates need to improve their understanding of the transmission of data using a Public Switched Telephone Network (PSTN). Many responses just re-worded the stem of the question by saying that switches were used in this type of transmission, which is insufficient for credit at this level of study.

(ii) Many of the benefits listed were very appropriate and the majority of candidates answered this part well. Some candidates need to be more precise in their suggestions for drawbacks. Answers such as, 'It is expensive' are not detailed enough for credit at this level. There needs to be a reason why it is more expensive. An example of a good answer is, 'A drawback would be that a leased line is expensive to install and maintain'.

(c) This question asked for the role of routers and gateways in a network. There were some very good answers explaining both the similarities and the differences between routers and gateways. Other candidates incorrectly described the role of a router in connecting a single device to the Internet, rather than the role of a router in a network.

(d) Almost all candidates could identify three other types of server. The most common incorrect answer was a client-server.

Question 2

(a) (i) Almost all candidates correctly stated the purpose of a language translator.

(ii) Many candidates were able to state that easier de-bugging is a benefit of using an interpreter during program development. Fewer candidates were able to correctly state a second benefit. Many answers described how an interpreter operates rather than stating benefits of use. There was some confusion between an interpreter and a compiler.

(iii) Almost all candidates correctly named an assembler.

(b) This question begins with 'Explain why', so a generic description of library files does not answer the question. Candidates should understand that to answer this type of question it is insufficient to say, for example, 'Library files will have been used by a lot of people so will be well-tested'; there needs to be a statement as to why this fact aids software development.

Question 3

(a) Most candidates answered this question well. A small number of candidates need to take care when drawing the arrows in the statements to ensure that they point in the right direction.

(b) There were a small number of excellent answers to this part question. Many candidates need to improve their understanding of processor instruction groups.

(c) Tracing of assembly language programs is improving steadily and this one was very well done, with many candidates providing completely correct solutions, neatly set out in the table provided. The most common error was the inclusion of the word `END` in the `OUTPUT` column after the final statement.

(d) (i) Almost all candidates correctly converted the denary value to 8-bit binary.

(ii) Almost all candidates correctly converted the denary value to Hexadecimal.

(iii) Most candidates answered this question well. The most common error was the omission of the two leading zeros on the final answer i.e. 0044.

Question 4

(a) (i) A minority of candidates provided a good response to this question. Other responses demonstrated considerable confusion between the internal operations of the microphone, that is, how the microphone captures the sound, and the encoding of the analogue sound to digital form.

(ii) Many candidates found this part question challenging. There was considerable confusion between the operation of a laser printer and the operation of a scanner. Candidates need to improve their understanding of how a laser printer works. Many answers incorrectly stated that the laser engraved the text onto the paper, or that the laser melted the toner ink onto the paper.

(b) Some candidates were able to describe how all the data for a single frame was recorded or displayed at the same time. Many candidates need to improve their understanding of what is meant by progressive encoding. Some candidates confused progressive encoding with interlaced encoding and vice versa.

(c) There were a number of good answers to this question. Many candidates need to improve their understanding of SRAM and DRAM. The question begins with, 'Explain the differences'. Some candidates need to understand that in order to answer this question each statement needs to refer to both SRAM and DRAM. An example of a good answer is, 'SRAM is used in cache memory, while DRAM is used in main memory'. Answers such as, 'SRAM is more expensive than DRAM' or, 'SRAM is faster than DRAM' are not detailed enough for credit at this level. There needs to be a reason why it is more expensive and how it is faster. A better answer is, 'SRAM has faster access times than DRAM'.

(d) (i) There were a number of good answers regarding the idea of legal ownership or the need for permission to use. Some answers to this question did not give a clear definition of copyright. There was some confusion between copyright and plagiarism.

(ii) There were some excellent answers to this question where the responses were given in context. Many candidates gave a generic description of an open source licence and had not applied their answer to the context given in the question. This is a question where the precise use of terminology is important too. It is insufficient to say that Shazia does not want anyone to see the 'software'; it is the *source code* that she wants to keep private.

(iii) The most popular correct answers were Shareware and a Commercial licence, and the majority of candidates correctly stated these two types of licence. A small number of candidates responded with a name only, despite the question asking for a name **and** description.

Question 5

(a) (i) The majority of candidates correctly stated that data redundancy referred to the duplication of data.

(ii) Many responses included a statement about normalisation and there were a small number of excellent descriptions of the normalisation process. Some candidates need to improve their understanding of relational databases. There were many answers that mentioned multiple tables but few descriptions of how the data was stored and linked. This is another question where the precise use of terminology is important. There was considerable confusion between tables and databases, with the word *database* frequently being incorrectly used to refer to a table within the database.

(b) (i) Many candidates correctly explained the difference between security and integrity. A small number of candidates need to improve their understanding of this topic. It is also not enough, at this level of study, to simply repeat the wording of the question. It is not enough to say, 'security is keeping data secure'. There needs to be some explanation of the meaning of 'keeping data secure'.

(ii) This question is one that needs to be read carefully. It asks for two security features of a DBMS. There were some very good answers. Other responses referred to security measures such as firewall and anti-virus software that are not part of the DBMS.

(iii) The majority of candidates need to improve their understanding of the terms query processor and developer interface.

Question 6

(a) This question was answered very well. The majority of candidates drew a logic circuit that correctly represented the logic expression given. Many of the solutions demonstrated excellent understanding by using a 3-input OR gate as the final gate. There is an improvement in the standard of the diagrams over the previous year. There were far fewer instances where `AND` and `OR` gates could not easily be distinguished.

(b) Almost all candidates provided a completely correct truth table.

COMPUTER SCIENCE

Paper 9608/13
Written Paper

Key messages

Candidates need to use precise terms related to the subject at this level of study.

Each question needs to be read carefully. A question that begins with 'Describe' requires a different type of answer from one that begins with 'Explain'.

Candidates are required to respond appropriately in the given context of the question. Generic answers that apply to any situation will not gain full credit.

General Comments

In general, candidates performed well on low level languages question, as well as the question on utility software. Many candidates found the questions about hardware and Database Management Systems (DBMS) more challenging.

Some candidates write their answer first in pencil and then over-write in ink. These candidates must ensure that the pencil marks are completely removed.

Comments on specific questions

Question 1

(a) Many candidates correctly entered the missing management tasks. The majority of candidates found completing the two missing descriptions much more challenging. There was considerable confusion between memory management and the management of the hard disk, with frequent incorrect references to the disk as memory. Many candidates need to understand that answers that re-word the text given in the question such as, 'It provides a platform for software' are not enough to describe the provision of a software platform at this level of study.

(b) (i) The majority of candidates were able to describe one action performed by each of the two utility programs given. Many candidates need to improve their understanding of how the software operates. There were a significant number of responses that stated the reverse operation for the disk defragmenter; namely that it split the files up.

(ii) Almost all candidates correctly identified three other utility programs. The question asked for three other utility programs. Some candidates need to be careful not to include those already given.

Question 2

(a) Generally, this question was well answered, with many candidates able to correctly explain the difference between security and privacy. A small number of candidates need to improve their understanding of this topic. It is also not enough, at this level of study, to simply repeat the wording of the question. It is not enough to say, 'security is keeping data secure'. There needs to be some explanation of the meaning of 'keeping data secure'.

(b) Many candidates were able to name two non-physical methods for improving the security of computer systems. Many of these candidates found describing the methods much more challenging. Candidates need to improve their understanding of this topic. An example of a good answer is, 'A Firewall checks all the incoming and outgoing network traffic and blocks any signals that do not meet pre-determined requirements. It can also keep a log of the traffic in and out'.

(c) (i) Many candidates were able to describe how each byte would have horizontal parity and that the parity byte would also have vertical parity. Explaining how the parity block is used to check for incorrect parity after transmission was more challenging. A number of responses gave great detail about odd and even parity without answering the question.

(ii) The majority of candidates understood that there needed to be an even number of incorrect bits.

(d) The majority of candidates answered this question well. There were some excellent explanations of how Frankie could act in the best interests of his client. Some candidates need to understand that the question gave a specific context and that answers needed to refer to the context given.

(e) Many candidates were able to explain that a compiler produces an executable file. Fewer candidates were able to correctly give any further explanations. Many answers described how a compiler operates rather than explaining why it would be used when the program was complete. There was some confusion between the use of a compiler and the use of an interpreter. A number of responses incorrectly stated that it was faster to run the compiler rather than that the executable file would probably run faster than interpreting the source code.

Question 3

(a) (i) Many candidates found this part question challenging and need to improve their understanding of what is meant by a data dictionary.

(ii) There were a small number of very good answers to this question and most candidates understood that a query processor would resolve queries. A significant number of candidates were quite vague in their reference to searching for data and displaying results. Many candidates found describing the purpose of a query processor to be quite challenging.

(b) Some candidates provided good descriptions of the primary and foreign keys; the description of their forming a relationship was sometimes rather vague. There was considerable confusion for some candidates between tables and databases, with the word database frequently being incorrectly used to refer to a table, for example, the primary key in one database links to the foreign key in another database.

(c) Almost all candidates could correctly draw the one to many relationships between `DEPARTMENT` and `EMPLOYEE_DATA` and between `DEPARTMENT` and `DEPARTMENT_MANAGER`. The most common incorrect relationship was the one to one relationship between `DEPARTMENT_MANAGER` and `EMPLOYEE_DATA`.

(d) Almost all candidates could correctly give three reasons why the database was normalised.

(e) (i) There were many correct answers to this question. Some candidates need to read the question carefully. The question asked for the statement to create the database. The most common incorrect answer was a statement to create a table `EMPLOYEES`.

(ii) There were some excellent answers to this question. Many candidates provided completely correct responses. The most common error was defining the `Gender` field to be of data type Boolean,

when the data given in the question is text. Some candidates need to improve their understanding of basic SQL programming.

(iii) The most common reasons for error on this question was the omission of quotation marks around `Finance` and `Female` and not correctly joining the two tables. Some candidates need to improve their understanding of basic SQL programming.

Question 4

(a) The majority of candidates correctly described what happened when the two instructions were run.

(b) The majority of candidates correctly identifying the group to which each instruction belonged to.

(c) (i) Many candidates were able to correctly give an example of a hardware interrupt. The most common answer was 'printer out of paper'. A number of candidates found giving an example of a software interrupt more challenging. There were some good response, for example, 'division by zero'.

(ii) There were a small number of very good answers to this question. The majority of candidates found it more challenging. Almost all candidates could explain that the processor needed to detect any interrupts and that interrupts were handled by the Interrupt Service Routine (ISR). Many candidates found any further description more difficult. There was some confusion between the ISR and an ISP (Internet Service Provider) and a small number of candidates described the Fetch-Execute (FE) cycle with no reference to interrupts.

Question 5

(a) The majority of candidates correctly identified the minimum number of bits required for the maximum number of colours. The most common errors were joining 2 colours to 2 bits and 256 colours to 9 bits.

(b) (i) There were a few correct answers to this question. Many candidates need to understand that repeating the information given in the question is insufficient for credit at this level of study. It is not enough to say that 40fps means 40 frames per second.

(ii) There were a number of good answers to this question. Many candidates need to improve their understanding of interlaced and progressive encoding. The question begins with, 'Describe two differences'. Some candidates need to understand that in order to answer this question each description of a difference needs to refer to both interlaced encoding and progressive encoding. An example of a good response is, 'In progressive encoding the frames are not divided into fields. In interlaced encoding each frame is divided into two fields, one contains the data for the odd numbered rows and one contains the data for the even numbered rows'.

(c) A minority of candidates provided a good response to this question. Many candidates need to understand that repeating the information given in the question is insufficient for credit at this level of study. It is not enough to say that a sampling rate of 88.2kHz means that samples are taken at 88.2kHz or that a sampling resolution of 32 bits means that the samples have a resolution of 32 bits. Further description of sampling rate and sampling resolution is needed.

COMPUTER SCIENCE

Paper 9608/21
Written Paper

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to have developed these and need to be able to apply them to the scenarios presented to them to access the full range of marks.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and candidates should ensure that they use these correctly. It is also important that candidates writing program code use the correct syntax for their chosen language.

The understanding of fundamental programming concepts is essential. Examples of confusion include the difference between a literal and an identifier and the difference between `OUTPUT` and `RETURN`.

Candidates should read each question carefully before attempting to answer it. Questions may address individual topics in a number of different ways.

Candidates should attempt all questions and should avoid leaving blank responses. They should ensure their work is set out clearly, particularly where tables are used.

General comments

There were some excellent programming solutions. A significant number of candidates demonstrated skill levels that suggested they had little programming experience.

Candidates who offer solutions using Python needed to take care to maintain the correct indentation, as this is key to defining the program structure.

Comments on specific questions

Question 1

(a) (i) Many candidates gained full credit for correct constructs. The pseudocode examples were of a mixed standard. The most popular responses gave some good examples of iteration and selection. Some candidates did not give any response for the pseudocode examples.

(ii) Many candidates gave the correct response for the first pseudocode statement. Mixed responses were seen for the other statements.

Many responses identified the two `WRITEFILE` statements as Input rather than Output and the `READFILE` statement as Output instead of Input.

(b) (i) Many candidates answered this question well and gained at least partial credit. Candidates needed to have quotes around a string to clearly identify that the value is a string and not a variable, for example, 'tripod' here is a literal string value whereas tripod would be a variable. Credit could not be awarded where candidates did not include the quotes.

(ii) Many candidates answered this question fully correctly. The second variable `Version` gave an example value of `'C'`. Some candidates incorrectly gave this data type as a string rather than a character.

(c) Many candidates are familiar with the terms black-box and white-box testing and some good responses were seen. Other candidates often described black-box testing which did not answer the question and some confused the two types of testing. There were some incorrect responses that stated "random data would be used as the code is unknown". The input data must be known to be able to calculate the expected output.

Question 2

(a) There were many good responses which correctly identified a Conditional loop and gave a clear explanation. Some responses were too vague, stating a Boolean loop or a True and False loop. Stronger candidates used the correct terminology. Credit was given for those stating a While or Repeat Until loop.

(b) Many candidates correctly identified at least one feature of the program. Those that gave Input and Output, or Variables and Constants as two separate responses only gained partial credit. Stronger candidates were able to identify three different features.

(c) Many candidates recognised the correct type of control structure. Candidates were awarded credit for clearly describing the actions taken for each of the different values of `x`. Responses such as "different actions taken depending on the value of x" were considered too vague for credit.

Question 3

(a) Stronger candidates answered this question well using accurate pseudocode syntax. These candidates often accurately initialised the two integer variables and correct output of a concatenated message. Many candidates identified the need for a loop but sometimes this was incorrectly formed. Candidates needed to use the correct symbol for assignment ← rather than the = symbol. A number of candidates did not answer this question.

(b) Stronger candidates were able to recognise the difference between a function call and a procedure call. A function will always return a value to a variable whereas a procedure is simply called by stating the procedure name followed by any parameters. The response expected here was a procedure heading with three parameters and their data types. As the first two parameters needed to maintain their values after the procedure was finished, they needed to be passed by reference. Few candidates recognised the need for `ByRef` before the parameters `AverageValue` and `ZeroCount`. The `ArrayName` was of data type `Array`. Many candidates gave this data type as string.

Question 4

(a) (i) A few candidates correctly traced this code, and answered fully correctly. Many candidates gained credit for the Index column, recognising that the code would iterate twelve times, which was the length of the string parameter. As in previous questions using characters and strings, quotes needed to be included to distinguish between variables and literal values.

(ii) Candidates gaining full credit in the previous question mainly recognised the effect of the function. In addition, many of those achieving partial credit were able to recognise that the repeated space characters were removed.

(iii) A small number of candidates recognised the effect of initialising AfterSpace to TRUE and most of these gained full credit. When `AfterSpace` is true, the selection statement `IF AfterSpace = TRUE` would evaluate to true, then the `IF NextChar <> Space` would evaluate to false, ignoring the next two lines of code which add the spaces at the front, therefore not including them in the new string.

Many candidates incorrectly stated that it would have no effect on the outcome of `Result`.

(b) A variety of incorrect declarations of a 2D Array were seen. Some candidates gave two separate arrays, some used dimensions [4:500] and many based theirs using python type syntax [1:500] [1:4]. Stronger candidates were able to write the correct pseudocode syntax.

Many candidates correctly used a nested FOR loop and gained credit for this but some forgot to declare the extra variable needed for the inner loop. A few candidates assigned an empty string "" to the array instead of the literal string 'Empty' as stated.

(c) Only the strongest candidates were able to answer this question correctly. Many candidates gave no response.

Question 5

(a) Candidates need to recognise that responses such as 'saves time', 'more efficient, 'saves space' are too vague. More detail is expected for credit at this level. Those candidates who gained credit for this question most often referred to the fact that a built-in function is already written and tested which reduces chances of error.

(b) To gain credit for this question, candidates needed to be familiar with, and have experience of writing pseudocode. Many candidates did not attempt this question. However, there were many good solutions from candidates who were familiar with this. Most of these solutions recognised that an array would be the best way to record random values selected in the range 1 to 50 and declared an appropriate array. Some of these candidates then omitted the initialisation of the array which is needed before the data in it can be processed.

The `RAND` function needed for this solution was given in the appendix. To ensure that the value returned from `RAND` is between 1 and 50, the value returned needed to be incremented by 1 to prevent zero being returned. Many candidates tried to avoid the problem by using `RAND(51)` but this would include all values from 0 to 50 rather than 1 to 50. As this function returns a real number, candidates needed to convert the return value to an integer. The `INT` function required was also given in the appendix.

Stronger candidates understood that comparison statements must use the = symbol and not the assignment ← symbol.

Question 6

(a) This question was related to the scenario described in the start of the question and the responses should have been related to this scenario. Many candidates did not achieve credit as they gave general answers regarding the good practice of using program modules to construct a program. For example:
modules can be split between programmers
modules can be re-used in other programs.

These responses were not relevant in this situation which required a small program consisting of three modules.

As in **Question 5(a)**, some responses needed more detail. Responses such as 'easier to debug' should instead have stated "Each module can be tested separately, which makes it easier to identify any errors".

(b) This question required a programming solution. Candidates needed to have experience of writing code in one of the three recommended languages to enable them to answer this type of question. A few excellent responses were seen. A variety of solutions were given, mainly in Python, especially in the method used to read the lines from the file. Some of these read all the lines from the file into an array then extracted the elements from the array in blocks of three relating to the three items of data. Credit was given to these different correct solutions.

Many of those candidates not achieving full credit needed to ensure that they used the correct syntax when reading from a file in a loop. This question required the file to be read three times in each iteration in order to extract the CD data relating to a single CD. Some candidates had one read statement, which meant the loop would have to repeat three times in order to get all the data relating to one CD.

Many candidates did not attempt this question or were unable to write any recognisable program code.

COMPUTER SCIENCE

Paper 9608/22
Written Paper

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to have developed these and need to be able to apply them to the scenarios presented to them to access the full range of marks.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and candidates should ensure that they use these correctly. It is also important that candidates writing program code use the correct syntax for their chosen language.

The understanding of fundamental programming concepts is essential. Examples of confusion include the difference between a literal and an identifier and the difference between `OUTPUT` and `RETURN`.

Candidates should read each question carefully before attempting to answer it. Questions may address individual topics in a number of different ways.

General comments

Candidates who offer solutions using Python need to take care to maintain the correct indentation, as this is key to defining the program structure.

A significant number of candidates demonstrated skill levels suggesting they had little programming experience.

Comments on specific questions

Question 1

(a) (i) The majority of candidates indicated a good understanding of the concept of an algorithm as a sequence of steps. A frequent mistake was to refer to specific programming features.

(ii) Candidates responded well to the activity types. There were some mixed responses for the pseudocode example. A common mistake was to give `INPUT`, followed by a prompt, but without a variable name as in the following example:

```
INPUT "Enter a value"
```

A small but significant number of responses *described* the activity instead of giving an example. As stated in the Key Messages above, candidates need to read the question carefully to ensure they understand it before making their response.

(b) (i) Most candidates gained some credit on this question. A common fault was to omit the quotation marks for the final answer, which are needed to differentiate between a string and an identifier name.

(ii) This type of question has appeared relatively frequently, and virtually all candidates were able to achieve high marks. Common mistakes included `STRING` instead of `DATE`, and `STRING` instead of `CHAR`.

Question 2

(a) (i) The question used the key word 'algorithm'. The majority of responses gave answers that incorrectly related to the code itself.

(ii) Correct responses usually referred to reusable code and the ability to assign modules to different programmers or teams. A significant number of candidates gave answers that related to library routines rather than to subroutines in general.
A number of candidates used a response, such as "Easier to..." that only vaguely related to the question.

(iii) Most candidates correctly identified the module as a function and were able to give a correct justification. A relatively common mistake was to state that `DoSomething()` was a procedure because it "performed a task".

(b) At this level, candidates were required to use the correct technical terms when describing the activities. For example, just 'code' is too vague when referring to the source code. Additionally, candidates should appreciate that simply stating, "the Editor is used to edit" is not sufficient at this level.

Candidates who used the correct technical terms usually gained full credit. Translating source code to object code was a frequent and accurate response for Translator. A response such as "converting code to a language understandable by the computer" is too vague at this level of assessment.
A small number of responses suggested that low-level code would be translated into high-level.

Those achieving one mark often gained it for their response given for Debugger, with most identifying its use during program testing to detect errors.

(c) Most candidates identified the correct control structure. Few responses adequately described the function of the code. In many cases, responses simply contained copies of the statements from the question rather than descriptions of their actions.

Many candidates correctly stated that the code would only carry out the actions when `Result` is less than 20. Few mentioned the parameters being passed. Candidates often attempted to describe the terminating condition (i.e. `Result >= 20`) and it was common in these cases for the "or equals" part of the comparison to be omitted.

Question 3

(a) (i) Many candidates correctly used parameters A and B. A significant number of responses suggested a function rather than a procedure.
Only a small number of candidates recognised that parameter C was passed by reference.

A small but significant number of candidates were not able to answer this question.

(ii) Many candidates gained full credit on this question part. Candidates appeared to be much more familiar with functions than they are with procedures.

(b) 'Iteration' and 'Selection' were the most popular responses. Some responses lost marks for incorrect terminology such as stating that the structure chart contained `IF` statements.

A small number of responses repeated the features given in the question.

Question 4

(a) (i) Almost all candidates gained credit for the first column, and most also gave the correct characters for the second column. Many candidates did not gain credit for the second column due to not enclosing values in the column in quotation marks to identify them as characters. The third and

fourth columns were often correct, although many contained an erroneous leading zero. A mixture of responses was seen for the fourth column. The most common mistake was not restarting the string after each "number", and therefore ending up with a copy of the original parameter string.

(ii) There were a significant number of correct responses to this question part. Many other candidates gained this mark if they had correctly carried forward the final value for the variable `Selected` from their trace table.

(b) (i) Those candidates that gained credit, usually identified that there was no space at the end of the string. Fewer went on to explain that this meant that the final comparison would not be made.

A large number of weaker responses referred to either logical errors in the code, the use of invalid data types or incorrect type conversion. A common mistake was to attempt to add one to the loop count in an attempt to check the last number. Some responses simply repeated the phrase from the question, "the function did not return the largest value".

A number of responses gave pseudocode rather than a required explanation.

(ii) There were some very good responses that stated another selection statement should be added before the final value is returned, to compare `Selected` with the value of string `NewValue`.

Many candidates gained credit by stating that a space should be added to the end of the string. Often they did not specify where in the algorithm this should be done. A number of responses suggested that the function should be called with a different value string, which did not represent a change to the algorithm as the question required.

Many weaker responses simply suggested the use of white-box or black-box testing.

Question 5

(a) Many responses gained partial credit, usually for opening the file and looping until the End of File.

Answers often referred to the use of a counter variable. Few stated that it should be initialised to zero.

Many solutions included incrementing a counter within a loop. Reading a line from the file was frequently omitted.

Some solutions suggested the use of an existing function to perform the counting.

Response content varied widely, from simple descriptive prose to pseudocode or program code.

(b) A small number of candidates provided a full correct response. The majority of candidates recognised the need to open the file, perform a count-controlled loop, and finally make some form of output.

Many candidates correctly opened the file for reading. They often did not close the file.

A common mistake was to attempt to use `EOF()` and `CLOSEFILE()` without providing a file name, and as in **part 5 (a)** many solutions did not include reading the file or did not include a variable to hold the line read.

Although the question asked for a procedure, many solutions gave a function header. The requirements specified in the three bullets were often not satisfied, particularly regarding the provision of the file name as a parameter. Many solutions used a literal file name string in the procedure, in some cases, despite including a parameter in the module header.

Question 6

(a) Most responses included a function header and some form of loop. Many candidates gained some credit for declaring the local variables and for the prompt and input. Some responses used the

correct code for the final concatenation and return. Some candidates incorrectly used `OUTPUT` rather than `RETURN`.

Fewer candidates provided correct solutions for the tests in the loop. Many solutions did not include code to check the length.

Solutions often correctly included simple comparisons statements such as:

```
FirstChar >= 'A' AND FirstChar <= 'A
```

In some cases, the comparison operators and/or the logical operator were incorrect. Many solutions converted the character to its ASCII code value, making a more complex solution and often the values used for comparison were not the correct ones. Many candidates attempted to use functions which convert to upper case such as Python's `upper()`, resulting in many incorrect comparisons such as:

```
if ThisChar == upper(ThisChar):
```

Few solutions checked the last four characters correctly. Many solutions included an incorrect comparison such as:

```
LastChars >= "0000" AND LastChars <= "9999"
```

Many good responses included loops that extracted each individual character (four right characters) and tested each one individually. Many responses correctly used substring functions from the candidate's chosen language. In a significant number of cases, functions were from another language was used, for example, the attempted use of `MID()` in a Python solution.

Many answers seemed to suggest a failure to understand the key similarities and differences between subroutines and functions. For a good solution for **parts 6(a)** and **6(b)**, it was essential that candidates appreciated the detail given in the table in the question rubric for the three subroutines given.

The contents of many weaker responses bore little relation to the question and some candidates appeared to mix up the requirement of this question with that of **part 6 (b)**.

(b) Most good responses recognised the need for a procedure header and some form of loop.

A prompt and input was included in many responses. Some responses did not first check on the return value from `WriteInfo()`.

A minority of candidates recognised that `GetInfo()` and `WriteInfo()` were functions. The following construct was widely seen:

```
Call GetInfo()
```

Candidates appeared to be much more familiar with functions than they are with procedures.

Many solutions used native `INPUT` statements, sometimes in addition to `GetInfo()`.

Many correctly recognised the need to call function `WriteInfo()` in both cases. Often only a single parameter was passed.

A significant number of responses included native file handling operations, which were not required.

Many responses indicate a lack of programming experience and solutions that combined program code and pseudocode were common. A small number of very weak responses did not include any recognisable program code.

(c) The majority of candidates gained almost full credit for this question.

Some responses suggested that candidates do not understand what is meant by the ‘module declaration’, evidenced by the fact that many of these responses attempted to include all the code for the function.

COMPUTER SCIENCE

Paper 9608/23
Written Paper

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to have developed these and need to be able to apply them to the scenarios presented to them to access the full range of marks.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings. It is important that candidates use these correctly.

It is important that candidates writing program code use the correct syntax for their chosen language.

The understanding of fundamental programming concepts is essential. Examples include the difference between a literal and an identifier and the difference between `OUTPUT` and `RETURN`.

Candidates need to read each question carefully before attempting to answer it. Questions may address topics in a number of different ways and it is necessary to apply knowledge in a way that meets the requirement.

General comments

The majority of program code answers used Visual Basic or Python. These were some excellent solutions in both languages. Candidates who offer solutions using Python need to take care to maintain the correct indentation, as this is key to defining the program structure. A significant number of candidates demonstrated skill levels that suggested they had little programming experience.

If a candidates crosses out an answer, any 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.

Comments on specific questions

Question 1

(a) Many candidates gained full credit for this question. Correct symbols were used in most cases. Some candidates moved the loop test to the end of the conditional statements and where this was functionally equivalent the mark was awarded. In some cases, this resulted in an incorrect number of iterations. A common mistake was to write the pseudocode for a FOR loop in a single process box.

Weaker responses demonstrated basic errors such as incomplete flow lines or decision boxes with only a single output.

(b) (i) Most candidates gained between two and four marks, with many achieving full credit. The most common mistake was to omit the quotation marks around the string "postroom".

(ii) Most candidates were able to achieve high marks. A common mistake was to give `STRING` instead of `CHARACTER` for `Version`.

Question 2

(a) Most responses recognised the need for a loop of 100 iterations.

Answers often referred to the use of a counter variable. A minority of candidates stated that it should be initialised to zero.

Many vague descriptions were seen, such as ‘the count will add up all the elements’, which were too imprecise to gain credit.

Response content varied widely, from simple descriptive prose to pseudocode or program code.

(b) Many candidates gained one or two marks for the first two mark points. The third mark point for the objective of the process was rarely awarded.

References to the breaking down of a program did not gain a mark.

(c) The responses for the Mode were given correctly in many cases, and most of these candidates then went on to give a good description.

A small but significant number of responses incorrectly suggested Write mode.

A number of responses suggested that the question had not been understood. Among these were answers that described different file types or user permissions.

(d) Most responses gained a mark for an example of a feature of the Editor and many also gained a second mark for describing a debugger feature or for stating that a debugger is used to find errors. The mark for describing the ‘Translate’ operation was given less often.

Responses were sometimes vague, often simply stating ‘code is translated to a language understood by computer’ which is too vague at this level of assessment.

Question 3

(a) Many good responses were seen, often gaining 5 marks.

Many answers correctly showed the repetition arrow and module hierarchy was correct in virtually all cases.

The most common mistake was to not use a double-headed arrow to indicate the parameter that was passed by reference.

(b) The phrase 'transferrable skills' was only occasionally seen; most marks were gained for examples of program features that Albert could be expected to recognise.

Some answers simply listed features of a program with no explanation. These did not address the question and gained no credit.

The explanation that high-level languages were 'a lot like English' appeared in many answers.

Question 4

(a) Most candidates recognised the need for an array together with a count-controlled loop containing statements to assign values to each element. Many did not include the output prompt and often the student number was omitted. Many responses did not include declaration statements for either the array or the count variable.

(b) Many correct answers usually addressed the first mark point, describing an advantage based on the reduced amount of code. A smaller number of responses correctly described an advantage from having a single identifier name.

Incorrect answers usually mentioned reduced storage space or faster execution.

Question 5

(a) (i) Generally well answered, with most candidates addressing 'every path' in some way. Those losing the mark usually gave vague generalisations such as 'to make sure it works'.

(ii) Many correct answers seen. Incorrect answers often referred to testing methods using an IDE such as variable watch window.

A small number of answers bore little relevance to the question.

(b) (i) A significant number of responses gave an incorrect test string; usually containing three words, suggesting that candidates did not read the question carefully. As the mark points were separate, many of these were able to gain a subsequent mark for a correct explanation.

(ii) The marks for the correct outputs were given in most cases. Many explanations were vague. Some did not describe how the algorithm should be changed, describing instead why the output was incorrect or stating how the input string should be changed.

Question 6

(a) A wide range of responses was seen. Most included a function header and a loop together with some attempt at file handling. Several Python solutions read the entire file into a list and subsequently processed the lines from there.

A challenging part of the solution for many candidates appeared to be calculating the correct number of characters from the file data to compare with the search string, and few gained this mark point. The relatively simple approach of taking the length of the search string was often shunned in favour of performing a substring operation on the line read from the file. Various native substring functions were seen. In cases where an index value was obtained this was often not transferred correctly to the comparison operation. Several more elementary solutions were seen, based on extracting the name one character at a time up to the '*' character. Credit was given for these working solutions.

Few gained full mark, although some excellent solutions were seen. Marks were regularly lost for incorrect syntax in the chosen language or from providing a mixture of pseudocode and program code. For example, VB solutions often included `RETURN String` in the function header rather than `AS String`.

Many answers seemed to suggest a failure to understand the key similarities and differences between subroutines and functions. For a good attempt at parts **6(a)** and **6(b)**, it was essential that candidates appreciated the detail given in the table in the question rubric for the three subroutines given.

The contents of many weaker responses bore little relation to the question and some candidates appeared to mix up the requirement of this question with that of part **6(b)**.

(b) Most candidates recognised the requirement for a count-controlled loop and also returned a count value at the end of the function. Many correctly incremented the count if the student name had not been not found. In many cases the count was declared but not initialised.

A significant number did not appreciate that `SearchFile()` was a function and therefore returned a value. Often a response would state '`Call SearchFile()`' This presented the candidate with the problem of what variable to compare to an empty string in the next part of the algorithm.

As in **6(a)**, simple errors such as the omission of and ENDFOR statement to mark the end of the loop often lost marks.

(c) Two marks were given to many responses for correct function name and the return of a Boolean. Many gave a literal string for the filename as the parameter rather than an array name.

Some responses suggested that candidates do not understand what is meant by the 'module declaration', evidenced by the fact that many of these responses attempted to include code for the function.

COMPUTER SCIENCE

Paper 9608/31
Written Paper

Key messages

Candidates need to show an in-depth study of the topics in the syllabus and make good use of appropriate technical terminology for this advanced theory paper. Candidates, who have studied the theory and have also practised the precise use of appropriate tools and techniques, were able to demonstrate successfully how they could be used to solve the problems set on the examination paper.

Candidates need to show good examination technique by responding to each question after carefully reading each in order to understand the exact requirements. Questions that ask the candidate to 'Explain how….' require a technical explanation of how the task described in the question is performed. Questions that ask the candidate to show their working require the intermediate steps in the calculation to be shown as well as the final answer.

General comments

Candidates need to read questions very carefully before attempting to write an answer. For example, candidates need to carefully consider the abstract concepts of feedback and protocols as part of a system.

Comments on specific questions

Question 1

(a) The majority of candidates calculated the denary value of the floating-point number correctly and showed their working. A common incorrect answer seen was to omit the minus sign.

(b) The majority of candidates calculated the normalised floating-point number correctly and showed their working.

(c) (i) The majority of candidates wrote the correct mantissa and exponent for the largest positive number.

(ii) The majority of candidates wrote the correct mantissa and exponent for the smallest non-zero positive number. A common incorrect answer was `000000000001` for the exponent.

(d) The majority of candidates correctly identified the effects of increasing the number of bits for the mantissa or the exponent. Fewer candidates went on to explain the trade-off between the two. An example of an explanation of trade-off could be 'increasing the precision of the exponent will lead to a decrease in the bits available for the mantissa thus decreasing the range of numbers that can be used'.

Question 2

(a) (i) Many candidates correctly drew a diagram of the star topology using the devices mentioned in the question; no extra devices were required. Candidates needed to show the data flow on their diagram.

(ii) Candidates could usually identify at least one benefit to the admissions department of using star topology. Candidates needed to answer this question in the context of the use by the admissions department given in the question.

(b) A minority of candidates gave the detailed technical description of CSMA/CD protocol required by the question. The answers seen were often imprecise and incorrectly referred to the protocol taking an active part in the process rather than providing the rules that need to be adhered to by the process.

(c) (i) A minority of candidates considered both internal and external network connections.

(ii) Some candidates correctly explained that the NIC provided a unique MAC address to identify the device on the network. Others incorrectly wrote about National Identity Cards. A small number of candidates did not attempt this part of the question.

(iii) A minority of candidates correctly identified the use of radio communication to access the network without the need of a physical connection.

Question 3

(a) The majority of candidates correctly identified a NOR gate.

(b) (i) The majority of candidates correctly completed the truth table.

(ii) The majority of candidates correctly identified a half adder. A common incorrect answer given was a full adder.

(iii) The majority of candidates correctly identified the two outputs.

(c) Many candidates correctly used some Boolean algebra to simplify the given expression. Many candidates were not able to perform the final simplification.

Question 4

(a) The majority of candidates correctly entered the symbols and their token values in the table. A common error in the type column was to incorrectly classify the symbol for `Cambridge` as a string rather than a constraint.

(b) The majority of candidates correctly completed the output from the lexical analysis.

(c) (i) Many candidates were able to write the equivalent code. A common error was to omit `STO 236` after `ADD 237`.

(ii) The majority of candidates showed some understanding of the need to optimise code.

Question 5

(a) The majority of candidates were able to enter the correct terms in the descriptions.

(b) Some candidates showed a good understanding of how a secure connection is established using SSL. Stronger responses included checking the server's digital certificate and making appropriate use of the public key for encryption of the session key.

(c) The majority of candidates named two correct malware types and two correct methods of prevention.

Question 6

(a) A minority of candidates were able to answer this part well. Many responses were often imprecise and incorrectly referred to feedback taking an active part in the process by turning on heaters or taking readings from sensors rather than enabling automatic adjustments to the process.

(b) A minority of candidates were able to answer this part well. Candidates needed to apply their knowledge of feedback to the control system for an indoor swimming pool.

(c) Many candidates correctly identified a suitable monitoring system and explained why there was no control incorporated into the system.

COMPUTER SCIENCE

Paper 9608/32
Written Paper

Key messages

Candidates need to show an in-depth study of the topics in the syllabus and make good use of appropriate technical terminology for this advanced theory paper. Candidates, who have studied the theory and have also practised the precise use of appropriate tools and techniques, were able to demonstrate successfully how they could be used to solve the problems set on the examination paper.

Candidates need to show good examination technique by responding to each question after carefully reading each in order to understand the exact requirements. Questions that state 'Explain how….' require a technical explanation of how to perform the task described in the question. Questions that state 'Show your working.' require candidate to show the intermediate steps in the calculation, as well as the final answer.

General comments

Candidates need to read questions very carefully before attempting to write an answer. For example, in **Question 5(c)(i)**, the instruction is 'Describe **two** vulnerabilities that a malware can exploit.', and not describe two possible effects of malware.

Comments on specific questions

Question 1

(a) (i) Many candidates calculated the denary value of the floating-point number correctly and showed their working. A common incorrect answer given was 6.875, provided by candidates who incorrectly assumed that the binary point was at the start of the mantissa.

(ii) The majority of candidates provided a correct reason.

(iii) The majority of candidates gave the correct normalised mantissa and exponent. A common incorrect answer was `0101` for the exponent.

(b) (i) Most candidates correctly converted +11.625 to binary. Normalisation proved more of a challenge with some candidates not showing the required working for obtaining the exponent.

(ii) This part of **Question 1** proved to be the most challenging. To gain full credit, candidates needed to show how –11.625 would be converted to a normalised binary number, either by calculating the binary values, or converting +11.625 to a negative number. A common error was not to mention the exponent in the conversion.

(c) Many candidates understood the problem of not being able to convert some denary numbers to exact binary values. Some candidates also correctly identified that the difference in value between the denary and the binary would increase, and become significant enough to be seen, if two such numbers were multiplied together and the answer output.

Question 2

(a) Most candidates were able to name a correct alternative method.

(b) Many candidates gave a suitable example of a situation where circuit switching would be used. Most candidates correctly identified the need for a dedicated circuit to be made available. A minority of candidates were able to successfully justify their choice.

Question 3

(a) (i) Most candidates were able to successfully complete the Karnaugh Map.

(ii) Many candidates correctly identified one group of four ones, $\overline{A}$, from the Karnaugh map. The second group of four ones, $\overline{B}$, proved more challenging for some candidates.

(iii) The majority of candidates correctly gave $\overline{A}$; a common wrong sum-of-products given was $\overline{A.B}$.

(b) (i) This question was generally answered well.

(ii) This question was generally answered well.

(iii) This question was generally answered well.

Question 4

(a) There were some excellent descriptions of the main steps in the evaluation of a Reverse Polish Notation (RPN) expression using a stack. Candidates who gave clear descriptions used the correct terminology of push and pop related to the process of evaluating the given expression. Common incorrect answers given by candidates included using more than one stack, or placing the operators on the stack.

(b) Many candidates correctly identified the contents of the stack at each stage of the evaluation and gained good marks. A few candidates did not demonstrate the process and gained zero or very few marks.

Question 5

(a) Understanding of why public and private keys were needed was shown by many candidates. Fewer candidates provided a technical explanation of how these keys were used to ensure that the email message remained private.

(b) Some candidates showed a good understanding of how asymmetric encryption was used to send a verified message. Candidates' explanations that included encrypting the message digest with the government department's private key and sending this well as the message, so that the sent message digest could checked against the message digest recreated by Sanjeet's computer gained full credit.

(c) (i) Many candidates were awarded full credit for this question. Some candidates incorrectly described the effects of malware rather than vulnerabilities that malware could exploit.

(ii) This question was generally well answered. The majority of candidates were awarded the mark.

Question 6

Many candidates correctly identified two other hardware devices. A common incorrect response was to include a device already mentioned in the question, for example, a sprinkler or a processor. The purpose of a sensor is to measure an identified property rather than sensing or detecting.

Question 7

(a) Many candidates correctly attributed at least four of the statements to a RISC or CISC processor.

(b) (i) Many candidates were awarded full credit for this question. Some candidates incorrectly identified parts of a computer.

(ii) Many candidates correctly identified that a large number of processors were required. Fewer candidates went on to describe that the processors worked collaboratively on the same task and they needed to communicate via a messaging interface.

Question 8

(a) Many candidates correctly identified two states for a computer process. Most candidates gave a description of the identified state. A minority of candidates provided full descriptions.

(b) This question proved challenging to many candidates. Those candidates who provided descriptions of how the use of the resource stated in the question could be maximised gained better marks.

COMPUTER SCIENCE

Paper 9608/33
Written Paper

Key messages

Candidates need to show an in-depth study of the topics in the syllabus and make good use of appropriate technical terminology for this advanced theory paper. Candidates, who have studied the theory and have also practised the precise use of appropriate tools and techniques, were able to demonstrate successfully how they could be used to solve the problems set on the examination paper.

Candidates need to show good examination technique by responding to each question after carefully reading each in order to understand the exact requirements. Questions that ask the candidate to 'Explain how….' require a technical explanation of how the task described in the question is performed. Questions that ask the candidate to show their working require the intermediate steps in the calculation to be shown as well as the final answer.

General comments

Candidates need to read questions very carefully before attempting to write an answer. For example, candidates need to carefully consider the abstract concepts of feedback and protocols as part of a system.

Comments on specific questions

Question 1

(a) The majority of candidates calculated the denary value of the floating-point number correctly and showed their working. A common incorrect answer seen was to omit the minus sign.

(b) The majority of candidates calculated the normalised floating-point number correctly and showed their working.

(c) (i) The majority of candidates wrote the correct mantissa and exponent for the largest positive number.

(ii) The majority of candidates wrote the correct mantissa and exponent for the smallest non-zero positive number. A common incorrect answer was `000000000001` for the exponent.

(d) The majority of candidates correctly identified the effects of increasing the number of bits for the mantissa or the exponent. Fewer candidates went on to explain the trade-off between the two. An example of an explanation of trade-off could be 'increasing the precision of the exponent will lead to a decrease in the bits available for the mantissa thus decreasing the range of numbers that can be used'.

Question 2

(a) (i) Many candidates correctly drew a diagram of the star topology using the devices mentioned in the question; no extra devices were required. Candidates needed to show the data flow on their diagram.

(ii) Candidates could usually identify at least one benefit to the admissions department of using star topology. Candidates needed to answer this question in the context of the use by the admissions department given in the question.

(b) A minority of candidates gave the detailed technical description of CSMA/CD protocol required by the question. The answers seen were often imprecise and incorrectly referred to the protocol taking an active part in the process rather than providing the rules that need to be adhered to by the process.

(c) (i) A minority of candidates considered both internal and external network connections.

(ii) Some candidates correctly explained that the NIC provided a unique MAC address to identify the device on the network. Others incorrectly wrote about National Identity Cards. A small number of candidates did not attempt this part of the question.

(iii) A minority of candidates correctly identified the use of radio communication to access the network without the need of a physical connection.

Question 3

(a) The majority of candidates correctly identified a NOR gate.

(b) (i) The majority of candidates correctly completed the truth table.

(ii) The majority of candidates correctly identified a half adder. A common incorrect answer given was a full adder.

(iii) The majority of candidates correctly identified the two outputs.

(c) Many candidates correctly used some Boolean algebra to simplify the given expression. Many candidates were not able to perform the final simplification.

Question 4

(a) The majority of candidates correctly entered the symbols and their token values in the table. A common error in the type column was to incorrectly classify the symbol for `Cambridge` as a string rather than a constraint.

(b) The majority of candidates correctly completed the output from the lexical analysis.

(c) (i) Many candidates were able to write the equivalent code. A common error was to omit `STO 236` after `ADD 237`.

(ii) The majority of candidates showed some understanding of the need to optimise code.

Question 5

(a) The majority of candidates were able to enter the correct terms in the descriptions.

(b) Some candidates showed a good understanding of how a secure connection is established using SSL. Stronger responses included checking the server's digital certificate and making appropriate use of the public key for encryption of the session key.

(c) The majority of candidates named two correct malware types and two correct methods of prevention.

Question 6

(a) A minority of candidates were able to answer this part well. Many responses were often imprecise and incorrectly referred to feedback taking an active part in the process by turning on heaters or taking readings from sensors rather than enabling automatic adjustments to the process.

(b) A minority of candidates were able to answer this part well. Candidates needed to apply their knowledge of feedback to the control system for an indoor swimming pool.

(c) Many candidates correctly identified a suitable monitoring system and explained why there was no control incorporated into the system.

COMPUTER SCIENCE

Paper 9608/41
Written Paper

Key messages

Candidates need to demonstrate their programming skills and algorithmic thinking in a variety of different ways. They need to show their ability to write object-oriented code and to use the methods appropriate to complete the algorithms described.

Candidates also need to show their understanding of different abstract data types, both by demonstrating how items were added and/or removed from them, as well as writing code to implement the given structures.

General comments

Many candidates were able to clearly demonstrate their understanding of abstract data types and were able to implement these appropriately.

Some candidates found the object-oriented programming questions more challenging, with some candidates showing a lack of experience in writing constructors, get and set methods, and using these methods to write programs.

Candidates need to read the questions carefully to make sure they have met the required criteria, for example, where a question asks for a value to be returned, this should not be output instead. If a question asks candidates to use specific identifiers and names of parameters, then candidates should make sure they are using these in their answers.

Comments on specific questions

Question 1

(a) (i) (ii) This question was answered well by many candidates who were able to follow the operations and complete the current contents of the stack.

(b) Many candidates were able to describe the core principle of a queue as a first in, first out data structure. Fewer candidates could expand on this and describe further attributes of a queue. Most often candidates mentioned the use of head and tail pointers, but it was rare for candidates to mention the linear and circular queue.

Question 2

(a) (i) There were mixed responses to this question. Many candidates were able to add the data to the tree in the correct order; some candidates attempted to enter them in numerical order as opposed to the order given. Some candidates recognised the need to identify null pointers and completed these appropriately.

(ii) This question was answered well by many candidates; those who produced an incorrect tree in **part (i)** were often able to create a suitable table for their tree. Some candidates did not use appropriate null pointers, some left them blank whilst others used the index 0 as a null pointer; this is invalid because there is data in index 0 and is therefore not pointing to null.

(b) Many candidates were able to complete some of the sections appropriately. A common error was in the first gap, where candidates identified the variable `CurrentPointer` as being returned as opposed to the data type. Candidates should be familiar with the pseudocode guide and should be able to read and interpret this correctly.

(c) **(i)** Candidates struggled to describe an imperative programming paradigm. The most common response involved procedural programming, but few candidates could explain this further and some candidates described the declarative paradigm instead.

(ii) Candidates found this question more straightforward and there were some good descriptions of object-oriented programming. Most candidates were aware of the use of classes and objects, but fewer candidates could use these terms appropriately and some candidates just listed key words.

(d) **(i)** This question required candidates to make use of their language specific constructor a indicated in the design diagram at the start of the question. This design also stated the attributes and their initial values within the constructor i.e. `Score` is `0`, and `Category` is `"Not Qualified"`. Many candidates did not read this introduction carefully and performed other actions, such as attempting to input the score and category within the constructor.

(ii) This question required an understanding of the purpose and the function of get methods. As with the constructor, the introduction to **part (d)** stated what each of these functions had to do. Many candidates attempted to set a value within the method, or output the value instead of returning it.

(iii) Candidates were required to write a validation rule within a set method. Many candidates were able to define the procedure and ask for the appropriate input. Fewer candidates were able to create a procedure that looped until a valid ID was entered, with many using one `IF` statement to perform the check. Candidates must check the question carefully and identify all of the component parts that are required in their answer.

(iv) Candidates needed to write a method to check whether the parameter was valid. Some candidates did not make use of a parameter as required. They attempted to input a value into the procedure; other candidates did both. They sent the data as a parameter and then overwrote it with an input. It is important that candidates read the question carefully and where specific identifiers are required; these should be used in the code. This question required a value to be returned to the main program. Some candidates attempted to output the value, or wrote the correct value to a variable, but did not then return this in their code.

The method should have only set the value to `Score`, if it was valid. Another common error was to write this value at the start of the algorithm, before checking whether it was valid or not.

(v) The set method in this question required an `IF` statement to compare the attribute to the parameter values. Some candidates attempted to read in both the value for `Score` and the value for `Grade;` these were both properties from the original question as stated in the introduction to this question part. When writing expressions such as comparisons, candidates must make sure they are using their language-operators and not mathematical operators i.e. >= instead of ⩾.

(vi) This question part required candidates to use the methods they had written in previous questions. Candidates were expected to write a procedure to read in the values, create an instance of the class they had been producing, and then use the set methods to assign the correct values. Most candidates were able to input the correct values. Some candidates missed the requirement for prompts for these inputs.

Fewer candidates were able to create an instance of the object in their chosen language. Candidates should have experienced creating classes and then create objects of these classes using their chosen language.

Some candidates recognised the need to deal with returned values from `SetScore()`.

(e) This question required candidates to identify example data that could be used to test each of the conditions. This required an understanding of the comparison operators and how these affect boundaries. Some candidates struggled with this question, with many were unable to distinguish between greater than, and greater than or equal to. Boundary data should be at the limit of what is

accepted, but many candidates used values that were over the boundary, which is regarded as invalid data.

(f) **(i) (ii)** These questions were answered well, with many candidates providing correct responses to both parts.

(iii) Many candidates were able to follow the algorithm and get most of the rows correct. Some candidates did not increment the index, and/or did not reset it after each iteration.

Question 3

(a) This question was answered well by candidates. Most candidates gave a feature as defined in terms of itself. A significant number of candidates recognised the need for a general case and a stopping condition.

(b) Candidates found this question challenging. Some candidates were able to describe the use of a stack in this process and how data is added to it, and then taken off, when the function unwinds. Some candidates attempted to give another definition of recursion, and some candidates discussed the features of a compiler without the emphasis on what happens specifically when recursion needs to be implemented.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

Candidates need to demonstrate their programming skills and algorithmic thinking in a variety of different ways. They need to show their ability to write object-oriented code and to use the methods appropriate to complete the algorithms described.

Candidates also need to show their understanding of different abstract data types, both by demonstrating how items were added and/or removed from them, as well as writing code to implement the given structures.

General comments

Many candidates clearly demonstrated their understanding of abstract data types and implemented these appropriately.

Some candidates found the object-oriented programming questions more challenging, with some candidates showing a lack of experience in writing constructors, get and set methods, and using these methods to write programs.

Candidates need to read the questions carefully to make sure they have met the required criteria, for example, where a question asks for a value to be returned, this should not be output instead. If a question asks candidates to use specific identifiers and names of parameters, then candidates should make sure they are using these in their answers.

Comments on specific questions

Question 1

(a) (i) Many candidates were able to add the nodes in the correct places, and many added the appropriate null pointers. Those candidates, who put the nodes in the incorrect places, often gained credit for putting in the correct null pointers. Some candidates did not follow the order of the data given and instead attempted to enter them numerically which produced an incorrect tree structure.

(ii) Many candidates answered this question well. Those who gave an incorrect tree structure for part **(i)**, were often able to produce the correct table for their tree. Some candidates incorrectly used the data values as pointers instead of using the array indexes. Many candidates gave appropriate null pointer values. One common error was to give index the value 0 as a null pointer. The number zero cannot be used as a null pointer because there is a data item with the index 0, therefore those nodes would link to this data item.

(b) **(i)** Candidates were required to demonstrate their knowledge of constructors in their chosen language. Many candidates understood the need to use their own constructor as described in the class diagram. Some incorrectly used a create method. It is important that candidates read the requirements of the question correctly, and they refer back to the information at the start of the question. In the class diagram, candidates were told that the `Grade` should be initialised to "`Fail`" and that the `FinalMark` should be initialised as `0`. Some candidates attempted to read data input by the user within the diagram, instead of storing the data required. The question also required the constructor to take the centre number and candidates number as parameters and then concatenate these; some candidates attempted to take the whole `PaperID` as one parameter, or take this as an input from the user, which did not meet the specification provided.

(ii) Many candidates were able to demonstrate a good understanding of the purpose of get and set methods. Fewer candidates were able to explain how they supported security and integrity, with many candidates providing generic description of these. A common error was for candidates to state that the methods could not be accessed out the class; this is inaccurate because the values can still be accessed. They cannot be accessed directly, and set and get methods need to be used.

(iii) A significant number of candidates were aware of the purpose of get methods and were able to write at least part of them accurately. Common errors included the output of values within the methods instead of returning them. Another included declaring a procedure to return a value instead of a function returning the value. Some candidates seemingly confused get and set methods, by reading in values from the user and attempting to assign these to the attributes.

(iv) In this question, candidates needed to return either true or false; these were often output instead, or assigned to a variable that was not then returned within the function. This question required a straight forward `IF` statement making use of the parameter as given in the question. Some candidates were unable to use an `IF` statement correctly, or did not make use of the parameter instead attempting to input a value into the function. Candidates must make sure they are using appropriate operators for their language, for example the use of <= instead of the mathematical operator $\leqslant$.

(v) As with part **(iv)** this question required candidates to make use of the parameter values within an `IF` statement. Candidates needed to revisit the original specification at the beginning of **part (b)**. This question required the setting of the `Grade` attribute based on the value of the attribute `FinalMark`, and not the return of a value. Some candidates did not make use of this attribute. They attempted to read in a value to the method. A common error was for candidates to attempt to redefine the class's attributes within the set method, i.e. defining a new `Grade` and new `FinalMark` instead of using the attributes.

(vi) Candidates were required to use the methods they had defined in the earlier questions. Candidates were expected to create a main procedure to input values and then create an instance of their class. Many candidates were able to create an instance of an object in their chosen language. Fewer candidates were able to put together the information from the previous questions to send the appropriate data to the constructor. Some candidates were unable to use the get and set methods correctly in their language, for example, they attempted to send the object to the get and set methods as a parameter.

(c) Many candidates were able to demonstrate a clear understanding of the differences between linked lists and hash tables, and applied this appropriately to the context. The most common responses involved identifying that records could be directly accessed; this was quicker than a linked list. Fewer candidates were able to expand on this, for example, they explained how it was slower to follow the pointers and perform a linear search on the linked list. Some candidates confused the use of a hash table with hashing for encryption and discussed the security of the data. A common answer involved the use of memory, with candidates stating that a linked list required more memory because of the pointers, but they did not take into account the number of empty spaces that require declaring for a hash table to allow for sufficient space for all of the data.

Question 2

(a) The majority of candidates correctly identified that the first statement described a stack data structure.

(b) (i) (ii) There were mixed responses to this question. Some candidates correctly identified the stack structure as last in first out, but did not apply this to the given stack.

(iii) Candidates were given the requirements in the question i.e. the parameter that was required, and the values to return. It is important that candidates refer to back to the question whilst writing their answers to make sure they are meeting all the requirements. Some candidates attempted to read a value from the user instead of using a parameter, and many candidates did not actually return a value; either outputting the result, or assigning it to a variable, such as `Valid`, and then not returning it.

Candidates found the use of the `Top` global variable challenging, and often compared the top array element to `Top` instead of using it as the index.

A significant number of candidates were able to use the appropriate comparison to check the top of the array, using either its length, or the values assigned from previous questions to identify the size.

A number of candidates correctly assigned the new value to the top of the array, but did not then increment the `Top` counter.

Question 3

(a) This question was answered well by many candidates, who gave suitable features that they had used within their own programming experience. The most common responses included the use of colour coded text, auto-indent and auto-correct. Some candidates identified the use of indentation, and others, the use of comments as being features of an editor – whereas these are features provided by the programming language itself and not the editor.

(b) Many candidates identified the three types of test data using a variety of different names for each. The most common error was to identify testing strategies such as alpha and beta testing instead of types of test data.

Question 4

(a) (i) Answers were mixed to this question. The most common correctly identified error was line 26 (or the swapping of lines 28 and 30). Fewer candidates were able to correctly identify the other errors, giving a mixture of attempted corrections. When answering these questions, candidates should test the algorithm e.g. with example data, so they can trace each step and then see where and when the errors occur. If candidates have time, it is also recommended that they test their corrected algorithm.

(ii) The majority of candidates correctly identified a different searching algorithm.

(b) (i) Many candidates were able to gain marks for the completion of the trace table. The final mark was awarded least, with candidates stopping after the iteration and not providing the final iteration when the `FOR` loop runs for the second time. Candidates should refrain from writing '-' in spaces where values are not changed because these indicate that that value '-' is being assigned to the variables.

(ii) Many candidates found this question challenging. The question required the `FOR` loop in the algorithm to be converted into a `WHILE` loop. No other parts of the original algorithm required changing. Many candidates attempted to combine the `WHILE` loop with the `REPEAT UNTIL` loop by using two comparisons, and some candidates replaced the `IF` statement with the `WHILE` loop instead. Of those candidates who correctly used a while loop, often the criteria was incorrect, e.g. only using less than 4 instead of less than or equals to. Some candidates put the initialisation of the counter inside the `WHILE` loop which stopped the loop from working, and some candidates only incremented it when the IF statement condition was true. As with the error detection question, candidates should be testing their algorithms as they write them. They had already created a test table for the original algorithm, therefore could use the same data to test their amended algorithm.

(iii) Many candidates correctly identified the algorithm as a bubble sort. A significant number identified it as a selection sort.

(iv) This question was answered well with many candidates being able to give a different sorting algorithm to their first answer to **part (iii)**.

COMPUTER SCIENCE

Paper 9608/43
Written Paper

Key messages

Candidates need to demonstrate their programming skills and algorithmic thinking in a variety of different ways. They need to show their ability to write object-oriented code and to use the methods appropriate to complete the algorithms described.

Candidates also need to show their understanding of different abstract data types, both by demonstrating how items were added and/or removed from them, as well as writing code to implement the given structures.

General comments

Many candidates were able to clearly demonstrate their understanding of abstract data types and were able to implement these appropriately.

Some candidates found the object-oriented programming questions more challenging, with some candidates showing a lack of experience in writing constructors, get and set methods, and using these methods to write programs.

Candidates need to read the questions carefully to make sure they have met the required criteria, for example, where a question asks for a value to be returned, this should not be output instead. If a question asks candidates to use specific identifiers and names of parameters, then candidates should make sure they are using these in their answers.

Comments on specific questions

Question 1

(a) (i) (ii) This question was answered well by many candidates who were able to follow the operations and complete the current contents of the stack.

(b) Many candidates were able to describe the core principle of a queue as a first in, first out data structure. Fewer candidates could expand on this and describe further attributes of a queue. Most often candidates mentioned the use of head and tail pointers, but it was rare for candidates to mention the linear and circular queue.

Question 2

(a) (i) There were mixed responses to this question. Many candidates were able to add the data to the tree in the correct order; some candidates attempted to enter them in numerical order as opposed to the order given. Some candidates recognised the need to identify null pointers and completed these appropriately.

(ii) This question was answered well by many candidates; those who produced an incorrect tree in **part (i)** were often able to create a suitable table for their tree. Some candidates did not use appropriate null pointers, some left them blank whilst others used the index 0 as a null pointer; this is invalid because there is data in index 0 and is therefore not pointing to null.

(b) Many candidates were able to complete some of the sections appropriately. A common error was in the first gap, where candidates identified the variable `CurrentPointer` as being returned as opposed to the data type. Candidates should be familiar with the pseudocode guide and should be able to read and interpret this correctly.

(c) **(i)** Candidates struggled to describe an imperative programming paradigm. The most common response involved procedural programming, but few candidates could explain this further and some candidates described the declarative paradigm instead.

(ii) Candidates found this question more straightforward and there were some good descriptions of object-oriented programming. Most candidates were aware of the use of classes and objects, but fewer candidates could use these terms appropriately and some candidates just listed key words.

(d) **(i)** This question required candidates to make use of their language specific constructor a indicated in the design diagram at the start of the question. This design also stated the attributes and their initial values within the constructor i.e. `Score` is `0`, and `Category` is `"Not Qualified"`. Many candidates did not read this introduction carefully and performed other actions, such as attempting to input the score and category within the constructor.

(ii) This question required an understanding of the purpose and the function of get methods. As with the constructor, the introduction to **part (d)** stated what each of these functions had to do. Many candidates attempted to set a value within the method, or output the value instead of returning it.

(iii) Candidates were required to write a validation rule within a set method. Many candidates were able to define the procedure and ask for the appropriate input. Fewer candidates were able to create a procedure that looped until a valid ID was entered, with many using one `IF` statement to perform the check. Candidates must check the question carefully and identify all of the component parts that are required in their answer.

(iv) Candidates needed to write a method to check whether the parameter was valid. Some candidates did not make use of a parameter as required. They attempted to input a value into the procedure; other candidates did both. They sent the data as a parameter and then overwrote it with an input. It is important that candidates read the question carefully and where specific identifiers are required; these should be used in the code. This question required a value to be returned to the main program. Some candidates attempted to output the value, or wrote the correct value to a variable, but did not then return this in their code.

The method should have only set the value to `Score`, if it was valid. Another common error was to write this value at the start of the algorithm, before checking whether it was valid or not.

(v) The set method in this question required an `IF` statement to compare the attribute to the parameter values. Some candidates attempted to read in both the value for `Score` and the value for `Grade;` these were both properties from the original question as stated in the introduction to this question part. When writing expressions such as comparisons, candidates must make sure they are using their language-operators and not mathematical operators i.e. >= instead of ⩾.

(vi) This question part required candidates to use the methods they had written in previous questions. Candidates were expected to write a procedure to read in the values, create an instance of the class they had been producing, and then use the set methods to assign the correct values. Most candidates were able to input the correct values. Some candidates missed the requirement for prompts for these inputs.

Fewer candidates were able to create an instance of the object in their chosen language. Candidates should have experienced creating classes and then create objects of these classes using their chosen language.

Some candidates recognised the need to deal with returned values from `SetScore()`.

(e) This question required candidates to identify example data that could be used to test each of the conditions. This required an understanding of the comparison operators and how these affect boundaries. Some candidates struggled with this question, with many were unable to distinguish between greater than, and greater than or equal to. Boundary data should be at the limit of what is

accepted, but many candidates used values that were over the boundary, which is regarded as invalid data.

(f) **(i) (ii)** These questions were answered well, with many candidates providing correct responses to both parts.

(iii) Many candidates were able to follow the algorithm and get most of the rows correct. Some candidates did not increment the index, and/or did not reset it after each iteration.

Question 3

(a) This question was answered well by candidates. Most candidates gave a feature as defined in terms of itself. A significant number of candidates recognised the need for a general case and a stopping condition.

(b) Candidates found this question challenging. Some candidates were able to describe the use of a stack in this process and how data is added to it, and then taken off, when the function unwinds. Some candidates attempted to give another definition of recursion, and some candidates discussed the features of a compiler without the emphasis on what happens specifically when recursion needs to be implemented.