COMPUTER SCIENCE

Paper 9608/11
Written Paper

Key messages

Candidates must read each question carefully and follow any instructions given when answering the questions. Questions may address topics in various ways and it essential that candidate answers reflect this. Some questions require candidates to apply their knowledge to a particular scenario. In these instances, it is essential that answers refer to the context for full marks to be awarded.

This is a technical subject and candidates should use the technical language associated with the subject in an appropriate way. There is considerable misuse of terminology especially between words such as *database* or *file* and *table*, or *memory* and *storage*. Candidates must also understand that if acronyms are expanded, the correct terms for each initial will be expected.

It is essential that candidates indicate clearly to the Examiner when an answer continues either on an additional sheet, or on a blank space within the question paper.

General comments

Overall, the standard of presentation was good. Most Candidates' responses were clear to read.

Converting between different representations of data and tracing a low-level language program were usually carried out successfully. Candidates found questions about the theory of databases and IP addressing more challenging.

Comments on specific questions

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

Question 1

Candidates generally answered this question well. The question asked candidates to draw one or more lines to link each language translator to the most appropriate statement or statements. Some candidates need to understand that while both a compiler and an interpreter will **detect** errors, neither program is usually able to **correct** the error.

Question 2

(a) This question required some application of knowledge. The question asked the benefits to the veterinary surgery of upgrading their database. Many answers were generic statements about the benefits of relational databases over file-based systems. There were few descriptions of the benefits to the surgery. Some candidates should understand that when asked for a benefit answers such as 'it is better' or 'it is easier' without any further explanation will not be given credit at this level of study.

(b) (i) This question also required application of knowledge to the given SURGERY database. Many answers were generic statements about the requirements of 3NF without any application to the given tables.

(ii) There were very specific instructions given in the stem of this question. Candidates were told that the appointment table did not need to change, yet some answers included a changed appointment

table. Candidates were also instructed to identify the primary key(s) in each table. Some otherwise correct answers had no primary keys identified in any way. Some candidates did not copy the attribute names correctly from the given tables. These candidates need to understand that if attribute names are given in a question they must be correctly copied when used in an answer.

(c) (i) There were many correct answers to this question. Some candidates should understand that SQL stands for **Structured** Query language.

(ii) Many candidates found writing the Data Definition Language (DDL) statement challenging. There is considerable confusion between the use of the `UPDATE` statement and the `ALTER` statement.

(iii) Many candidates correctly completed the `SELECT` clause and one part of the `WHERE` clause, that is, `StaffID = 'JK1'`. Completing the second part of the `WHERE` clause and the `ORDER BY` clause was more challenging. Some candidates should understand that the SQL keyword for a sort in descending order is `DESC`. Some candidates did not copy the attribute names correctly from the given tables. These candidates need to understand that if attribute names are given in a question, they must be correctly copied when used in an answer.

(d) (i) There was considerable confusion between verification and validation. Many candidates described two validation checks rather than two verification checks. Some candidates who did describe verification checks should understand that a description of double entry that just says 'enter the data twice' is not enough. There needs to be an indication of a comparison between the two entries.

(ii) Many candidates described verification rather than validation. This question also required some application of knowledge. Some candidates should understand that paraphrasing the question is not enough. An example of a good answer is, '*A format check could be used to validate that the time is entered in the format <digit><digit><colon><digit><digit>*'.

(e) (i) Many candidates correctly completed the description.

(ii) Many candidates were able to correctly identify two authentication techniques that could be used. By far the most popular choices were password and biometrics. Some candidates should understand that encryption is not an authentication technique.

Question 3

(a) (i) There were many interesting and imaginative responses to this question. Some candidates did not read the question carefully enough and gave generic answers not relating to the product.

(ii) There were also some good answers to this question. Some candidates need to understand that for two marks two distinct points must be made. Some answers about Ria and her team acting ethically towards their colleagues were the same point written in two different ways.

(b) The identification of the types of licences was answered very well. Some candidates found it more challenging to justify their selection.

Question 4

(a) Most candidates were able to successfully trace this code.

(b) There were many correct answers for this question.

(c) Candidates found tracing this code more challenging. Some candidates need to improve their understanding of indirect addressing.

(d) There were many correct answers to this question. The most common incorrect answer was the writing of Op codes as mnemonics rather than identifying the modes of addressing.

(e) Candidates generally answered this question well. The most common incorrect responses were to tick the input and output of data column for `STO 120` and `LDD 20` instead of the data movement column.

Question 5

(a) (i) Almost all candidates correctly identified the streaming method and there were some excellent reasons why this was the method used. Some candidates found justifying their choice more challenging. Statements such as 'because he is watching it at the same time as it is taking place' which simply repeat the wording of the question are not enough.

(ii) There were some excellent responses for this question. The most popular reasons for the difference were issues of connectivity, buffer size and hardware capabilities. Some candidates should understand that if the question asks for three reasons, all the reasons given must be different.

(b) (i) There was considerable confusion when answering this question between progressive encoding and compression methods. Many candidates described the use of Run Length Encoding (RLE) on the frames rather than describing how the frames would be encoded for transmission.

(ii) Many candidates recognised that some lines were the same in the two frames and were able to state correctly that these lines would not need to be retransmitted.

(iii) There were many correct answers to this question.

(iv) Candidates found this question challenging. The most frequent answer was a file type, usually one of those given in the question.

Question 6

(a) Some candidates found converting the given denary integer into a two's complement binary integer challenging, especially as the denary value was a negative number. Some candidates converted the value given using a sign and magnitude representation.

(b) There were many correct answers to this question. Some candidates correctly converted `F` in Hexadecimal to `15` in denary and zero in Hexadecimal to zero in denary, but then simply added these values together giving an incorrect answer of `15` instead of correctly multiplying the 15 by 16 before the addition. Some candidates did not read the question carefully and converted the Hexadecimal value to binary rather than denary.

(c) There were many correct answers to this question.

(d) There were many correct answers to this question. Some candidates correctly converted each nibble but then incorrectly added their answers together.

Question 7

This was another question that required the application of knowledge. Some candidates should understand that when a question asks for the differences, it is necessary to make clear which of the items is relevant. It is not enough to say, for example, 'One file will take a long time to download, the other will not'. A better answer is, '*The 98 kHz file will take longer to download than the 44.1 kHz file*'.

Question 8

(a) There was generally a good understanding that a private IP address gave the laptop greater security and some candidates were able to correctly state why the router had both a public and a private IP address. Some candidates need to improve their understanding of the difference between a private and a public IP address. Some answers incorrectly stated that the private IP address was used to allow the laptop to connect directly to the Internet.

(b) There were some excellent answers to this question. Some candidates need to improve their understanding of the operation of the Domain Name System (DNS).

(c) There were many excellent answers to this question. Some candidates need to understand that the range of values allowed in an IPv4 address is 0 – 255, not 0 – 256.

COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

Candidates must read each question carefully and follow any instructions given when answering the questions. Questions may address topics in various ways and it essential that candidate answers reflect this. Some questions require candidates to apply their knowledge to a particular scenario. In these instances, it is essential that answers refer to the context for full marks to be awarded.

This is a technical subject and candidates should use the technical language associated with the subject in an appropriate way. There is considerable misuse of terminology especially between words such as *database* or *file* and *table*, or *memory* and *storage*.

It is essential that candidates indicate clearly to the Examiner when an answer continues either on an additional sheet, or on a blank space within the question paper.

General comments

Overall, the standard of presentation was good. Most Candidates' responses were clear to read.

Converting between different representations of data and tracing a low-level language program were usually carried out successfully. Candidates found questions about the theory of databases more challenging.

Comments on specific questions

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

Question 1

(a) (i) There were many correct answers to this question. The most common error was converting the denary value `105` to the positive binary number `0110 1001` instead of converting to Binary Coded Decimal (BCD).

(ii) Some candidates found converting the two's complement binary integer into denary challenging, especially as the answer was a negative number. Some candidates correctly showed –128 + 63 but then gave the answer as `65`, without the minus sign. Others treated the value given as sign and magnitude and so gave `-63` as the answer. `191` was also a frequent incorrect answer, where the value had been assumed to be unsigned binary. Some candidates need to understand how to differentiate between the various representations of binary numbers.

(iii) There were many correct answers to this question. Some candidates correctly converted `A` in Hexadecimal to `10` in denary and `B` in Hexadecimal to `11` in denary, but then simply added these values together giving an incorrect answer of `21` instead of correctly multiplying the 10 by 16 before the addition. Some candidates did not read the question carefully and converted the Hexadecimal value to binary rather than denary.

(b) (i) Many candidates found this question challenging. The question asked how the computer uses ASCII codes to represent characters. Answers were often a description of the ASCII character set, or a description of how characters input via a keyboard are converted. There was also confusion between the number of bits used to store each character, 7 (or 8), and the total number of

combinations, 128 (or 256). A common incorrect answer was that each character takes 128 bits to encode.

(ii) Candidates had little difficulty converting the three given characters to their equivalent denary ASCII codes. Some candidates then inexplicably added the three denary values together giving an incorrect answer of 267. Some candidates inserted punctuation symbols as separators between the three values. On this occasion these were ignored, in future that will not be the case. Some candidates gave correct answers in binary or Hexadecimal.

(iii) Almost all candidates were able to successfully calculate the denary ASCII code for the character '`H`'.

Question 2

(a) This was a question that required the application of knowledge. Candidates were asked to apply a given method of file compression (run-length encoding) to a **text** file. Many candidates incorrectly described other coding methods of compression rather than RLE. Some candidates ignored the text file given in the question and described the compression of images, which was not required.

(b) Many candidates found this question challenging. There appeared to be little understanding that written text has few repeated sequences of the same character and hence storing a copy of the character and a count of the number of consecutive occurrences, especially when the count was often 1, would take more storage space than storing just the sequence of different characters.

Question 3

(a) Candidates generally this question well. The last four spaces were most often all correctly completed. The first space was the one most likely to be incorrect. Candidates should understand that the Von Neumann model for a computer system used the **stored** program concept, where both data and instructions are stored in memory.

(b) (i) There were many very good attempts at tracing this program. There were very few answers that did not score any marks. Some candidates misinterpreted the instruction at address 51 and added the denary value `80` into the accumulator instead of the contents of address 80, but then were able to get back on track from the instruction in address 53. Some candidates incorrectly included the word `END` in the output column.

(ii) This was a clear example of a question where many candidates had not followed the instructions. The command statement says tick one box in each column. Many candidates had ticked more than one box in both the first and second columns.

Question 4

(a) Many of the answers seen for this question were completely correct. Incorrect answers were most often due to the candidate misinterpreting the position of the brackets in the expression and giving the truth table for X = (A NOR B) AND ((C XOR A) OR B) instead of for the expression given in the question.

(b) Many candidates found it challenging to give complete descriptions of the two gates and hence the difference between them. Answers such as, '*An AND gate outputs 1 if both inputs are 1*' are not enough for the mark. While clearly a true statement, it does not show the exclusivity of the output. There is no indication that any other combination of inputs results in a zero output. Similarly, for the NAND gate, giving an output of zero when both inputs are 1. Answers stating that a NAND gate is the opposite of an AND gate are also not enough for a mark. There needs to be some indication that the NAND gate inverts the output of an AND gate for any given pair of inputs.

Question 5

(a) This question required some application of knowledge. The question asked the benefits **to the teacher** of using a relational database. Many answers were generic statements about the benefits of relational databases over file-based systems, but there were few descriptions of the benefits to the teacher.

(b) (i) Many candidates need to improve their understanding of the developer interface of a DBMS. There was some confusion with Integrated Development Environments (IDEs) and many answers repeated benefits from **part (a)** and explained how the DBMS helped rather than the developer interface.

(ii) There were some very good answers to this question. Some candidates need to understand that the contents of a Data Dictionary are metadata rather than data.

(c) (i) There were some very good answers to this question. The Foreign key row was the one most likely to be incorrectly completed. A few candidates reversed the field names and table names.

(ii) This question required application of knowledge to the given `RESULTS` database. Many answers were generic statements about the requirements of 1NF, 2NF or 3NF without any application to the given tables.

(iii) Many candidates correctly completed the `SELECT`, `FROM` clauses and the insertion of the keyword `WHERE`. Completing the `WHERE` clause was more challenging. Some candidates did not copy the attribute names correctly from the given tables. These candidates need to understand that if attribute names are given in a question they must be correctly copied when used in an answer.

(iv) Three methods of data validation were usually correctly identified. Many candidates found it more challenging to give a description of the method applied to the `RESULTS` database. Vague statements such as, '*a type check to check the type of the data*' is insufficient for credit at this level of study. An example of a good answer would be, '*Method: A range check Description: The teacher can use a range check to ensure that the mark input for a test result is between zero and the maximum marks for that test*'.

(d) (i) There were some excellent answers to this question. Many candidates understood the need to keep the database secure.

(ii) There was some confusion between data backup and disk mirroring, backup was better understood than disc mirroring. Some candidates need to improve their understanding of backup and that it is not enough to store the backed-up data on an external hard drive unless that external drive is moved to a different location. Some candidates correctly identified on-line storage as a suitable location. Some candidates also need to understand that paraphrasing the question will not be enough for credit. Statements such as 'data backup is taking a backup of the data' are not enough. The concept of disk mirroring proved challenging to many candidates and there needs to be an improved understanding that disc mirroring requires the simultaneous writing of data to two (or more) discs and that in case of disc failure operations can be immediately switched to the other disc with no loss of data as the second disc is an exact replica of the first one.

Question 6

(a) There were many interesting and imaginative responses to this question. Some candidates did not read the question carefully enough and explained ways in which Malika was acting ethically rather than the manager.

(b) There were also some good answers to this question. Some candidates need to understand that for two marks two distinct points must be made. Some answers about Malika acting ethically were the same point just written in two different ways.

(c) There were many interesting responses to this question too. Popular correct suggestions were about talking to Malika and her colleagues, or organising team building events.

Question 7

(a) Many candidates found this question challenging. These candidates need to understand that while IPv4 is usually shown in denary notation, it can also be shown in Hexadecimal notation. Some candidates need to improve their understanding of the number of bits required for an IPv6 address.

(b) There were some good answers to this question. Some candidates found it more challenging and there was considerable confusion between static and dynamic IP addresses and static and dynamic RAM.

COMPUTER SCIENCE

Paper 9608/13
Written Paper

Key messages

Candidates must read each question carefully and follow any instructions given when answering the questions. Questions may address topics in various ways and it essential that candidate answers reflect this. Some questions require candidates to apply their knowledge to a particular scenario. In these instances, it is essential that answers refer to the context for full marks to be awarded.

This is a technical subject and candidates should use the technical language associated with the subject in an appropriate way. There is considerable misuse of terminology especially between words such as *database* or *file* and *table*, or *memory* and *storage*.

It is essential that candidates indicate clearly to the Examiner when an answer continues either on an additional sheet, or on a blank space within the question paper.

General comments

Overall, the standard of presentation was good. Most Candidates' responses were clear to read.

Converting between different representations of data and tracing a low-level language program were usually carried out successfully. Candidates found questions about graphics and IP addressing more challenging.

Comments on specific questions

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

Question 1

Candidates generally answered this question well. Some candidates need to understand that a program that rearranges the data on a disk so that files are contiguous is **de**fragmentation software, not fragmentation software.

Question 2

There were many correct answers to this question. The most common incorrect link was from progressive encoding to the description that stated that only the changed pixels were transmitted.

Question 3

(a) There were some very good answers to this question. Many candidates found it challenging to describe in enough detail why an interpreter would be used during the writing of a program. Answers such as, '*the interpreter is used for debugging*', are not enough as this statement could equally apply to a compiler. A better answer is, '*an interpreter is used because debugging is made easier as the errors can be corrected in real time*'.

(b) There were many correct answers to this question. Some vague statements referred to not needing **the code** after compilation, rather than saying that **the source code** would not be needed **to run** the program.

(c) There were only a small number of correct answers to this question. Many candidates need to improve their understanding of the process of partial compilation to an intermediate code which is then interpreted. There was considerable confusion about the order of use of the translation programs. Many responses just repeated answers given in **parts (a)** and **(b)**.

Question 4

(a) (i) Many candidates need to improve their understanding of IP addresses. The question stated in the stem that the laptop was on a home network and that the laptop connected to the Internet through a router. A popular answer to this question was, '*the laptop has an IP address so that it can connect to the internet*'. This says little more than the question stem. A better answer is, '*the laptop has a (private) IP address so that it can be identified on the home network and so that the router can direct data from the internet to the laptop*'.

(ii) There were some good answers to this question. Most popular was the idea of better security as the laptop's IP address would not be universally visible.

(iii) Some candidates should understand that when a question asks for the differences, it is necessary to include definitive statements about both items under discussion. It is not enough to say, for example, '*IPv4 had four groups of digits, IPv6 has more*'. A better answer is '*IPv4 has four groups of digits, whereas IPv6 has 8 groups of digits*'. A popular misconception was that IPv6 used 6 bytes rather than 16 bytes.

(b) Many candidates correctly described the cell phone network as a communication system to support the Internet. Some candidates found describing a second method more challenging.

(c) (i) Many candidates were able to correctly describe the first two lines of PHP code. The last two lines proved to be more challenging. A frequent error in an otherwise correct final description was that `<br>` meant that the text was emboldened, rather than that the text was followed by a line break. Candidates should take care when copying variable names from the code. The `$` symbol was often omitted.

(ii) There were many correct answers to this question. JavaScript was easily the most popular correct answer.

(d) This question requires an explanation of how parity checks protect the integrity of the data. It does not ask for an explanation or a description of parity. Many candidates need to ensure that they read the question carefully. Descriptions of odd and even parity were often given, with little or no reference to how the use of parity checks protects data integrity.

Question 5

(a) (i) There were many correct answers to this question. Some candidates need to improve their understanding of the use of the brackets in register transfer notation.

(ii) In general, candidates correctly answered this question.

(b) There were many correct answers to this question. Some candidates need to improve their understanding of the buses are bidirectional.

(c) (i) There were many correct answers to this question.

(ii) There were many very good attempts at tracing this program, and although some candidates found it challenging there were very few answers that did not score any marks. Some candidates misinterpreted the instruction at address 504 and added the denary value 902 into the accumulator instead of the contents of address 902, but then were able to get back on track from the instruction in address 506.

(d) (i) Almost all candidates were able to successfully calculate the denary value of the unsigned binary integer.

(ii) Some candidates found converting the two's complement binary integer into denary challenging, especially as the answer was a negative number. Some candidates correctly showed –128 + 74 but then gave the answer as 54, without the minus sign. Others treated the value given as sign and magnitude and so gave –74 as the answer. Some candidates need to understand how to differentiate between the various representations of binary numbers.

(iii) There were many correct answers to this question. Some candidates correctly converted the two nibbles into 12 and 10 in denary then stated that these could not be Binary Coded Decimal (BCD) as each value was greater than or equal to **10** instead of greater than or equal to 9.

Question 6

(a) There were many interesting and imaginative responses to this question. Some candidates did not read the question carefully enough and explained ways in which Benedict could act ethically rather than Lara.

(b) There were some good answers to this question. Some candidates need to understand that for two marks two distinct points must be made. Some answers about Lara acting unethically were the same point just written in two different ways.

(c) (i) There were many correct answers to this question.

(ii) There were many correct answers to this question. A commercial licence and a shareware licence were the most popular correct answers.

(iii) There were many correct answers to this question. An open-source licence was the most popular correct answer.

Question 7

(a) There were a small number of excellent answers to this question. Many candidates found it challenging to describe the encoding of the vector graphic logo. There was a lot of confusion with bitmapped graphics and many answers described the encoding of a bitmap using pixels rather than the encoding of a vector graphic.

(b) (i) Many candidates correctly calculated an approximation of the file size of the scanned image. Some candidates need to ensure that they read the question carefully. The question asked for the answer rounded to the nearest MB.

(ii) Many candidates could identify a method of lossy compression for the scanned image. Some candidates found it challenging to then describe the method identified. There was some confusion with lossless compression and candidates describing how run-length encoding (RLE) would be used to compress the image.

Question 8

(a) There were many correct answers to this question.

(b) There were many correct answers to this question.

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 be able to apply them to the scenarios presented.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and it is important that these are used correctly. Answers should be as precise and specific as possible and contain enough detail. For example, simply answering 'easier to debug' is not enough when a question asks for the advantages of modular program design.

Some candidates showed a lack of programming experience. A common mistake was the inclusion of unnecessary features, for example, file handling operations in a task where none are required, or the inclusion of code fragments from other solutions such as 'element swap' statements from a bubble sort. Programming skills are best learnt through the practical process of designing, writing and testing programs.

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

Candidates need to read each question carefully before attempting to answer it. Questions may address topics in various ways, and it is often necessary to apply knowledge in a specific way.

General comments

Candidates who offer solutions using Python need to take care to maintain the correct indentation.

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

If candidates cross out any answers, they must write any new answers clearly. Many candidates used blank pages for rough work when preparing their final response. In these cases, it is extremely helpful if candidates cross out any rough work.

Comments on specific questions

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

Question 1

(a) Many candidates gained at least two marks, usually from the list 'Design' 'Code' and 'Test'. Several answers correctly identified a preliminary analysis stage. Imprecise names such as 'planning' were often used for this.

General terms such as 'Development' were common, as were terms such as 'execution', 'compiling' and 'editing'.

A small number of candidates simply responded with unrelated computing terms such as 'Iteration, Procedure and Execution' or 'Input Process Output'

(b) The majority of candidates correctly defined Corrective Maintenance, often in very simple terms such as 'fixing bugs in the program'.

Adaptive maintenance was not answered as well. Many answers referred to a change being made, but not to the reason for the change. Good responses often referred to change in specification or required functionality.

(c) Most candidates correctly stated that a transferrable skill would help when learning a new language or when trying to understand a program in an unfamiliar language. Candidates often omitted to state that it is the skill in the known programming language which leads to the ability to learn and/or recognise specific features in a different programming language.

Some answers incorrectly stated that it was the ability of a programmer to transfer their skills to other members of their programming team and a small number referred to the transfer of data between programs.

(d) Many candidates understood the problem but often their description was either imprecise or incorrect. Some referred to identifier naming rules, such as 'shouldn't start with a number' or the fact that names had to be 'unique', which these were.

Many candidates stated that it would be easy to confuse the variables, but far fewer went on to explain that this would make the program more difficult to understand.

(e) Most candidates evaluated the four expressions correctly.

Question 2

(a) A minority of responses presented a clear and concise structured English description of the task. A wide range of styles was seen, ranging from pure narrative text to complete program code. This suggested that the use of structured English to describe an algorithm is not well practised.

Many candidates correctly identified the main structure of the solution: the need for input, a loop including some basic processing and a final output, following a comparison of some kind. Despite the question specifying 'prompt and input', the output of a prompt was omitted by many.

Candidates generally offered imprecise descriptions of the steps required. For example, the statement 'check the file for the user name' is not adequate.

Often, an answer would state that a file needed to be opened but would not specify a mode or a close operation.
Many recognised a loop was needed but did not state that this needed to continue until the end of file was reached. Very few stated that the search should stop as soon as new user name is equal to the name read from the file.

A significant number of candidates had not clearly understood the requirements of the problem and assumed that if the user name input was not found then it was to be written to the file. Similarly, some felt that if the name was not found then the user was required to enter subsequent names until one was found.

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

Some candidates did not correctly count the number of characters needed using the `MID()` and `LEFT()` functions.

(c) Many candidates provided correct response for the two-line numbers answers, and for recognising one of the function calls.

Not many suggested the term 'count controlled loop' as given in the syllabus. The term 'conditional loop' was seen quite often.

Few candidates demonstrated an understanding of what was meant by the 'scope' of the variable `OutString`.

Question 3

(a) Many good responses centred around there not being enough lines in the file for the given parameter values, or the fact that the given file might not exist.

The question stated that this function had been tested and was known to work with the given data but still 'syntax error' was often suggested as a possible cause of the 'unexpected result'

Candidates offering 'logical error' as a reason needed to provide some additional explanation or description.

Many answers focussed on the code itself (such as parameter order and syntax statements) rather than what the procedure was expected to do.

(b) Many candidates exhibited a very basic understanding of how a text file is processed. A common mistake was to assume that it was possible to go directly to a required line in the file without understanding that the text file must be read line by line from the start.

Several solutions, usually programming in Python, read the whole file into an array then used a single FOR loop to print the required lines using array indexes, demonstrating good problem-solving skills by these candidates.

Most solutions included a count-controlled loop. However, often the statement giving the range was incorrect. Candidates who opted for the two-loop 'seek' solution usually had the first loop correct.

(c) (i) Most candidates answered this question correctly by describing either a logical or run-time error.

A significant number named and described a syntax error instead.

(ii) Most candidates correctly referred to 'white-box testing' or to carrying out a 'dry-run with the use of a trace table'.

The question asked for a technique that may be used to identify an error rather than just to detect the presence of an error and this limited the scope of acceptable answers.

(d) 'Tried and tested' and 'readily available' were the two most popular answers.

Many answers did not distinguish between the use of subroutines in general and the availability and use of library subroutines.

Many answers for this question were in complete. For example, stating that library routines 'save development time' without the pre-cursor that this is possible as they are pre-written and readily available.

Question 4

(a) Many candidates understood the requirements and gained good marks. Many candidates did not put quotation marks around string values.

The question clearly states the data types of each parameter but very often this seemed to have been overlooked. A common mistake was to suggest values other than `TRUE` or `FALSE` for parameter `AddComma`.

The question asked for two tests to address different aspects of the function. Validation was not a feature and so the use of 'erroneous' data values was not correct.

(b) There were some very good attempts at this question. However, there were many responses that indicated low level of knowledge and understanding. A significant number of candidates did not attempt this question.

Many candidates recognised the need and correctly used the `NUM_TO_STRING()` function. An occasional mistake was to assign the resultant string back to number, which was an Integer.

Several solutions successfully tested the number to see if it was greater than 999 (or checked the length) but some then changed the value of `AddComma` accordingly.

There were some good attempts and ingenious ways of splitting up the number string and many of these made correct use of the `LEFT()` and `RIGHT()` functions.

Better solutions successfully concatenated the prefix to the start of the return string A small number of these contained an unnecessary test, for example:

```
IF Prefix <> ''
   THEN
      OutString ← Prefix & OutString
ENDIF
```

Other issues noted were as follows:

- not testing `AddComma` to determine whether a thousand-separator was required
- not using `Num_To_String()` to convert the Integer parameter to a String
- not checking that number provided to the function contained more than 3 digits.

A common final mistake was to output instead of returning the formatted string.

Question 5

(a) Some excellent solutions were seen. Many candidates made no realistic attempt to answer this question.

This question asked for a straightforward linear search of a 1D array. Many solutions implemented an inefficient algorithm that would continue searching even after the required hashtag was found.

Many weaker solutions included unrequired prompt and input statements.

The use of an immediate RETURN from within a loop was condoned. A common search error when this technique was used was to return –1 immediately if the array element was not equal to the hashtag being searched for. For example:

```
WHILE Index <= 10000
   IF TagString[Index] = HashTag
      THEN
         Return Index
      ELSE
         Return -1
   ENDIF

(Contd.)
```

(b) Some excellent solutions were seen. Many candidates made no realistic attempt to answer this question.

Better solutions used a conditional loop which terminated if the nth occurrence was found. These solutions usually correctly incremented a count when a Hash character was found and then followed this with a conditional statement to check if this was the required hashtag (as given by the function parameter).

The use of the `MID()` function to extract a character from the message string was recognised by many. As a rule, solutions that gained this mark point (MP2) usually went on to also gain the next three.

Several solutions attempted to access an individual character using an index, as might be the case in Python.

A small number of solutions searched only for the third hashtag, as per the example given in the module definition in the question.

(c) There were many good solutions to this question. Some candidates displayed little or no grasp of programming. Some candidates stated a programming language but were unable to program in the programming language stated. Some solutions were purely pseudocode despite the question asking for pseudocode.

Many candidates using VB attempted to access characters in a string as if they were elements in an array.

Some correctly addressed the need to check if the character at the start position was a hash (MP3), usually enclosing the majority of the remainder of the code in a `THEN` clause.

Many solutions did not correctly identify the end of a hashtag (MP6). Many of these just checked for an empty string or a hash character.

Many solutions attempted to concatenate the characters to form the hashtag (MP7) and then return it (MP9).

Some used an incorrect start position when extracting individual characters from the string and testing for the termination condition. This resulted in testing the first character (a hash) which would immediately terminate the loop.

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 be able to apply them to the scenarios presented.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and it is important that these are used correctly. Answers should be as precise and specific as possible and contain enough detail. For example, simply answering 'easier to debug' is not enough when a question asks for the advantages of modular program design.

Some candidates showed a lack of programming experience. A common mistake was the inclusion of unnecessary features, for example, file handling operations in a task where none are required, or the inclusion of code fragments from other solutions such as 'element swap' statements from a bubble sort. Programming skills are best learnt through the practical process of designing, writing and testing programs.

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

Candidates need to read each question carefully before attempting to answer it. Questions may address topics in various ways, and it is often necessary to apply knowledge in a specific way.

General comments

Candidates who offer solutions using Python need to take care to maintain the correct indentation.

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

If candidates cross out any answers, they must write any new answers clearly. Many candidates used blank pages for rough work when preparing their final response. In these cases, it is extremely helpful if candidates cross out any rough work.

Comments on specific questions

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

Question 1

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

In cases where only one correct term was given this was usually 'Output'.

'Procedure' was occasionally given instead of 'process' and general computing terms were given in a small number of cases.

(b) Many candidates misunderstood the question and rather than stating what an algorithm represents, they described one of the methods given in the question. Candidates need to be sure they have read and understood the question before they begin to prepare their answer.

(c) A small number of candidates offered a description of each data type rather than an example data value.

A common error was to omit quotation marks around string or character data values.

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

(e) Most candidates evaluated the four expressions correctly.

A minority of candidates wrote a modified form of each expression rather than evaluating it fully.

Question 2

(a) Many candidates missed the requirement to ensure the values were all unique. Among those who did attempt this, often just a simple `IF…` was used for a single repeat of the input values and only relatively few solutions included a loop to repeat the input as long as the uniqueness test failed.

Two simple mistakes often made were:

- not declaring `Average` as a real value
- calculating the average using the expression `A + B + C / 3` rather than `(A + B + C )/3`

Most candidates made some attempt to find the largest value. In many cases, the logic implemented with a nested `IF…` was incorrect. Those who employed a simpler two-stage algorithm which first found the larger of `A` and `B` and then went on to compare this with `C` in a second `IF…` expression usually got the right answer.

(b) Most candidates correctly answered this question.

The most common mistake was not recognising the need to use the `LENGTH()` function. Some suggested the value `12`, ignoring the subsequent string parameter given in the expression.

`MOD()` was substituted for `DIV()` on a number of occasions.

Some candidates attempted to use the `RIGHT()` function for the second expression. In some of these cases the keyword `STRING` was used to provide the second parameter.

(c) Some students correctly answered the questions, but many gave answers relating to the benefits of using subroutines in programs rather than benefits of sub-tasks when designing algorithms.

Question 3

(a) Many responses provided some good solutions. Other responses indicated a low level of knowledge and understanding.

A significant number of candidates either did not attempt this question or gave a solution in pseudocode.

MP1 and MP2 (on the mark scheme) were missed by many. The use of `PROCEDURE` as a keyword in VB solutions was common.

Most solutions used four separate variables for the inputs rather than an array and many did not check the range for the values input. The average was usually correctly calculated but any form of appropriate tolerance was often missing or incorrect, with weaker solutions often relying on a comparison such as `'>10%'`:

Better solutions sometimes included the tolerance into the test on each wheel, such as:

```
IF WheelA > (Average * 1.1) OR WheelA < (Average * 0.9)
   THEN
      (contd.)
```

The use of four separate variables often led to repeated lines of essentially identical code.

Quite a few solutions use output instead of return and some omitted this last step completely.

(b) A common mistake was for the data for Test 1 to indicate a skid.

The clearest responses gave 4 identical (or very closely matched) values for Test 1 and 3 identical numbers and one very different one for Test 2.

Several answers indicated a lack of understanding of the scenario description. Few candidates produced obvious and easy values for and against skidding. Many sets of values were unnecessarily complicated.

Question 4

(a) Many candidates provided correct solutions.

There were a many fundamental flowchart errors. The absence of an `END` symbol was one example as well as the absence of the output prompt.

Candidates should be aware that flow lines must be present in a flowchart. Many decision diamonds had only one output, and where there were two outputs it was not uncommon for these to be unlabelled.

Imprecise descriptions were common in the decision boxes intended to implement steps 3 and 4 of the original algorithm.

A common mistake when testing the number input was the use of the incorrect logical operator leading to an impossible condition:

```
IF Number < 0 AND Number > 9
   THEN
      (Contd.)
```

(b) (i) This question split candidates into two distinct groups. Many good answers were seen, often leading to accurate and complete trace solutions. Many candidates found this challenging and offered little or no answer.

Many candidates did not use quotation marks to denote a character or string value.

(ii) Most candidates correctly answered this question.

(iii) and (iv) These were very challenging questions and were only successfully answered by a small number of candidates, who exhibited a well-developed ability to analyse the pseudocode in order to identify the problem. Most of these correctly stated that the problem was related to the presence of repeated characters in `String2`.

Many answers suggested that the problem was related to the use of unequal length strings, despite the question stating that the function was for use on two equal-length strings.

Many candidates offered no answer, and many others related to general aspects of testing, rather than to a specific problem with the given algorithm.

There were few suggestions for **part 4(b)(iv)**. Of those that did suggest a solution, many of these related to first comparing the lengths of the two strings.

(v) Most candidates answered this correctly.

Black-box testing was the most common incorrect answer.

(vi) Single stepping and breakpoints were both well-known features; a variable watch window perhaps less so.

There was general confusion between features that would be used when writing a program (such as a context-sensitive prompt) and features that would be useful for later debugging.

Many answers imprecisely referred to error detection, without mentioning that it would be syntax errors that would be detected and that these would be highlighted, or indicated in some way.

Confused phrases such as 'Dynamic Syntax Errors' were common, as were general testing terms.

Question 5

(a) There were some excellent solutions to this question. Many struggled to provide a sensible solution.

This question asked for a straightforward initialisation of a 1D array.

A significant number of candidates confused initialisation with declaration. Many solutions consisted only of declaration statements.

Many solutions omitted the procedure declaration and/or end statement.

The question asked for pseudocode, so square array index brackets were required and the use of a back-arrow symbol for assignment is also expected.

(b) There were some excellent solutions to this question. Many struggled to provide a sensible solution.

Many candidates omitted the function header.

File open and close errors commonly included omitting quotation marks around the literal filename, opening the file in a mode other than `WRITE`, and not closing the file.

Most candidates included some form of loop in their solution and many recognised that a simple `FOR…` loop which iterated through the 10000 elements in the array was all that was required. Solutions where the file had been opened in `READ` mode understandably often also included an incorrect conditional `WHILE NOT EOF()` loop.

A successful attempt at identifying an unused element (MP4) was included in many solutions. Many used incorrect array index brackets.

The formation of the string to write to the file (MP5) often lacked the use of the `NUM_TO_STRING()` function.

Many responses did not initialise the counter variable.

Many answered showed some confusion between `OUTPUT` and `RETURN`.

(c) There were some excellent solutions to this question. Many struggled to provide a sensible solution.

Statements for creating a function heading and declaring and initialising a variable (MP1 and MP2) were often absent.

File handling (MP 3 and MP5) was usually correct in Python solutions but less frequently in VB, where the syntax used for opening of files was often incorrect.

Most candidates who attempted this question appreciated the need for a loop that ran until the end of file was reached (MP4).

Many solutions employed some sort of `split()` method, although a few good solutions were based on a linear search for the comma separator, and subsequent sub-string operations. Many candidates using VB attempted to access characters in a string as if they were elements in an array.

The type conversion required before the count string could be written to the array was often missing (MP7).

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 be able to apply them to the scenarios presented.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and it is important that these are used correctly. Answers should be as precise and specific as possible and contain enough detail. For example, simply answering 'easier to debug' is not enough when a question asks for the advantages of modular program design.

Some candidates showed a lack of programming experience. A common mistake was the inclusion of unnecessary features, for example, file handling operations in a task where none are required, or the inclusion of code fragments from other solutions such as 'element swap' statements from a bubble sort. Programming skills are best learnt through the practical process of designing, writing and testing programs.

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

Candidates need to read each question carefully before attempting to answer it. Questions may address topics in various ways, and it is often necessary to apply knowledge in a specific way.

General comments

Candidates who offer solutions using Python need to take care to maintain the correct indentation.

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

If candidates cross out any answers, they must write any new answers clearly. Many candidates used blank pages for rough work when preparing their final response. In these cases, it is extremely helpful if candidates cross out any rough work.

Comments on specific questions

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

Question 1

(a) Most candidates correctly stated that the process involved breaking down a task into sub-tasks. A large number also explained that this would make the problem easier to solve.

A common mistake was to refer to breaking down a program, often stating that the process would make the code 'more efficient'.

(b) This question was answered correctly by most candidates.

Common problems were the omission of quotation marks around an example of a `CHAR` and offering `STRING` or `REAL` that had been given in the question and were therefore excluded. Candidates need to read the given question carefully.

(c) (i) Most candidates correctly answered this question and referred to a language feature such as a loop or selection statement.

(ii) Many candidates correctly identified the term. Some did not make the connection between the wording of the question and the term 'transferrable skill' as used in the syllabus and offered a general statement that was often little more than a variation on the stem of **part 1(c)(i)**.

Several candidates did not answer this question.

(d) The phrase 'identify and locate' in the question was missed by many and so 'Black-box testing' was a very popular incorrect answer.

Many candidates correctly identified an IDE feature. Many of these gave two other IDE features, which was not required.

A small but significant number of candidates offered no answer.

Question 2

(a) A minority of candidates provided a clear and concise structured English description of the task. A wide range of styles was seen, ranging from pure narrative text to complete program code. This suggests that the use of structured English to describe an algorithm is not well practised.

Many candidates correctly identified the main structure of the solution: initialisation of the total and count, a loop including the input of a student mark together with some basic processing, and a final output following a calculation of the average. Many recognised a loop was needed but did not specify that this needed to continue until all 30 marks had been input.

Some descriptions attempted to enter all 30 marks at once.

A small number of solutions included an output statement within the loop.

Candidates generally gave imprecise descriptions of the steps required. For example, the statement 'look in the list of 30 students' is not enough.

(b) Most candidates provided clear statements of each error.

There was no obvious pattern for the few incorrect answers, but the phrase '`COUNT` is not defined' was seen on several responses as a description of the fourth error.

(c) Most candidates, who provided an accurate solution to this simple algorithm fragment in their chosen program language.

Although algorithm mistakes were rare, the one seen most often involved replacing the initial `WHILE...` clause with an `IF` statement. The consequent loss of meaning of the algorithm usually resulted in the loss of at least two marks.

A common mistake in Python was to use on a single = in a conditional statement.

A small number of candidates gave a solution in pseudocode. In some cases, this followed the declaration of the proposed programming language.

Question 3

Many candidates exhibited a reasonable grasp of this standard algorithm.

Most identified the need for an inner and outer loop. In many cases these were both count-controlled, but often the outer loop was conditional and terminated if no swap operation had been performed during the inner loop. An occasional boundary error on the inner loop meant that the resulting comparison of element 'n+1' would be beyond the upper bound of the array.

The element comparison and the swap operation were both usually correct.

Several solutions included a reducing boundary in the outer loop.

A common problem when two count-controlled loops were used was to compare elements using two different indices, rather than comparing adjacent elements.

For example:

```
FOR J ← 1 TO 4999
   FOR K ← 1 TO Boundary
      IF ProdNum[J] > ProdNum[K]
(contd.)
```

Many simple syntax errors were seen, such as:

- Incorrect procedure heading (use of '`PROCEDURE`' in VB and missing colon in Python)
- Use of a single '=' in a Python comparison
- Incorrect brackets for array indices
- Use of backarrow for assignment
- Terminating a loop with `ENDFOR`.

Question 4

(a) There were some excellent solutions to this question. Many struggled to provide a sensible solution.

This question asked for a linear search through a 1D array, arguably one of the simplest algorithms.

Many solutions omitted the function declaration and/or end statement (MP1).

The declaration of `Index` was omitted by some (MP2).

Most candidates who attempted the question also identified the need for a loop (MP3). In most cases this was a count-controlled loop iterating from 1 to 100, where the lop counter was also used as the array index. Many implemented the loop termination by using an immediate `RETURN` if the search value was found. Conditional loops were less common. Most of these successfully tested an index counter and a Boolean value to indicate whether the search value had been found. These were generally correct, although for many, the Boolean operator used to link the two conditions was erroneous; usually an `OR` which should have been an `AND`.

A small number of candidates chose to output the result rather than return it.

The question asked for pseudocode, so square array index brackets were required and the use of a back-arrow symbol for assignment was expected.

(b) Most candidates correctly answered this question.

Many general computing terms were incorrectly given, and several candidates provided no answer.

(c) Most candidates correctly answered this question.

(d) This question was answered well by a minority of candidates.

A common mistake was for the first output to be given as 25.

Explanations were generally not adequate. Many simply described the operation in terms of 'takes the value and adds five'. In some cases, the explanation did not correspond to the given output value.

(e) Many candidates offered advantages that were specific to the use of library routines, which the question did not ask for.

A small number gave short generic answers that failed to map to the question. An example would be 'more efficient'.

Question 5

(a) There were some excellent solutions to this question. Many struggled to provide a sensible solution.

This question required a search through a 1D array for an unused element and, if found, the initialisation of the corresponding element in the two arrays.

Better solutions usually included the declaration of a local Integer variable for the array index and a Boolean value to indicate that an element had been found, or a second integer to save the index of that element. Solutions that attempted an immediate return from within the loop after an unused element had been found and initialised often did not require this second variable and, in these cases, dependent only upon the declaration of an array index integer.

Most candidates who attempted the question also identified the need for a loop. In many cases, this was a conditional loop which terminated either when an empty element was found, or when the end of the array was reached.

Many solutions successfully tested an individual element correctly (MP3) and most of those that did this went on to store the index and terminate the loop (MP4 and MP5).

A small number of candidates chose to output the Boolean rather than return it, and many omitted this step altogether.

(b) There were some excellent solutions to this question. Many struggled to provide a sensible solution.

This question asked for the given `GetStart()` function to be used to count the number of hashtags in a given string.

Statements for creating a function heading and declaring and initialising a variable (MP1 and MP2) were often absent.

Many candidates recognised the need to call `GetStart()` repeatedly until it returned −1. Often the return value was successfully used in the conditional test of the loop.

Many candidates were unable to make correct use of the `GetStart()` function. The return value was often not assigned to a variable which made a successful subsequent test impossible.

Many candidates confused `OUTPUT` with `RETURN`.

(c) There were some excellent solutions to this question. Many struggled to provide a sensible solution.

This question asked for a linear search of a 1D array; this time with a subsequent increment of the corresponding element from the `TagCount` array and written in the candidate's chosen programming language.

The best answers tended to come from users of Python with the majority opting for a straightforward `FOR...` loop with an immediate return when the hashtag was found and the corresponding `TagCount` element incremented.

Mistakes often included the use of only a single equal sign for a comparison in Python and occasional incorrect array brackets.

Weaker solutions often made the mistake of simply incrementing a single variable rather than an array element.

(d) There were some excellent solutions to this question. Many struggled to provide a sensible solution.

Many solutions correctly initialised a `Max` variable to zero before searching through the `TagCount` array (MP1).

Most solutions recognised the need for a loop to search through the array (MP2). Many solutions split the algorithm into two separate loops; the first to find the largest value and the second to count how many times this value occurred. In many cases, this approach was very successful.

A common mistake in a single-loop solution was to replace the `ELSE` in the pseudocode below with and `ENDIF` and so split the nested IF statements into two separate ones. The effect of this was to incorrectly increment the `Count` as soon as a new `Max` value was detected (MP6).

```
IF TagCount[Index] > Max
   THEN
      Max ← TagCount[Index]
      Count ← 1 // there is only one max value
      MostPopTag ← TagString[Index]
   ELSE
      IF TagCount[Index] = Max
         THEN
            Count ← Count + 1 // another max value
      ENDIF
ENDIF
```

The outputs were generally correct, but many candidates attempted to use the concatenation operator on an integer value without first using the `NUM_TO_STRING()` function. Solutions where the items were presented as a list in the `OUTPUT` statement obviously did not need this step (MP7 and MP8).

COMPUTER SCIENCE

Paper 9608/31
Written Paper

Key messages

Candidates need to show an in-depth study of the syllabus topics and make good use of the appropriate technical terminology required to answer questions this paper. Candidates, who have studied the theory and have also practised the precise use of these 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 ensure that they provide the information asked for in the question set. It is not enough to rewrite the information given in the question as an answer.

General comments

Candidates need to read examination paper questions carefully before writing an answer.

For example, in **Question 2(a)(i)**, the answer must include only data types shown in the pseudocode shown in the question. In **Question 3(a)(i)**, the answer must be a description of the TCP/IP protocol suite. In **Questions 4(a)** and **4(c)(ii)**, the answer must be shown as a sum of products. In **Question 7(c)(i)**, only the instructions set out in the question can be used in the answer.

Comments on specific questions

Question 1

(a) Most candidates found the correct denary value and showed their working. A common error was to move the binary point in the wrong direction.

(b) (i) Most candidates gave the correct answer for the mantissa; the exponent proved to be more of a challenge for many candidates. A common error was to write `0011` instead of `0010` for the exponent.

(ii) A minority of candidates provided a full description of a problem that could occur when floating-point numbers are not normalised. Many candidates identified the problem of lack of precision, a few candidates added that this was due to redundant leading zeros in the mantissa.

Question 2

(a) (i) Many candidates identified some composite and non-composite data types. Fewer candidates correctly identified those data types that were used in the pseudocode shown in the question.

(ii) This question was generally answered well.

(b) Some candidates wrote correct pseudocode assignment statements to store the details of the box given in the question. One common error was to not store the values in the first element of the array `myBox[]`; another common error was the incorrect use of quotation marks for the values in the assignment statements

Question 3

(a) (i) Some candidates gave a good description of the TCP/IP protocol suite. Describing the TCP/IP protocol suite proved to be more of a challenge for many candidates. A common error was to describe the operation of TCP and/or IP.

(ii) Most candidates correctly identified the protocol as BitTorrent, and many candidates gave an acceptable description of the way BitTorrent protocol works.

(b) (i) Most candidates correctly identified the method as packet switching. Many candidates provided a good description of this method.

(ii) Many candidates found the identification of a benefit and a drawback challenging. An example of an acceptable answer is 'Benefit: Packets of data from the movie can be rerouted if there is a problem with the route selected. Drawback: Packets can be delayed as different routes are taken.'.

Question 4

(a) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks. A common error was to incorrectly write $\overline{A.B}.C$ for $\overline{A}.\overline{B}.C$

(b) Many responses showed a correctly completed Karnaugh Map.

(c) (i) Nearly all responses showed correct grouping. The most common error was to incorrectly group the three products in the bottom row.

(ii) Most candidates provided a correct simplified sum-of-products for the answer to **part (c)**.

(d) Many candidates correctly identified the uses of X and Y. Some candidates identified the logic circuit. A common error was to give the answer for the logic circuit as a half adder and not a full adder.

Question 5

Many candidates correctly completed most of the statements about a virtual machine. A common error was to insert the word virtual in the gaps in the statements.

Question 6

(a) Many candidates gave some similarities and differences.

(b) Some candidates gave some similarities and differences. A common error was to state that a digital signature was a scan of a handwritten signature.

(c) Many candidates gave some similarities and differences. A common error was to confuse the terms phishing and pharming.

Question 7

(a) Many candidates stated the correct information stored in the addresses. A common error was not to use the placement of the binary point to find the number of litres of fuel.

(b) Many candidates correctly converted the information given to the correct binary form for storage. A common error was not to use the placement of the binary point in the conversion of the number of litres of fuel.

(c) (i) Some candidates wrote assembly language instructions chosen from the table shown on the examination paper to complete the task set. A common error was to use an instruction not given in the table. Some candidates did not attempt this part of **Question 7**.

(ii) Most candidates completed the assembly language program. A common error was to not ensure that the process was continuous and jump to another routine rather than the start of the program.

COMPUTER SCIENCE

Paper 9608/32
Written Paper

Key messages

Candidates need to show an in-depth study of the syllabus topics and make good use of the appropriate technical terminology required to answer questions this paper. Candidates, who have studied the theory and have also practised the precise use of these 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 ensure that they provide the information asked for in the question set. It is not enough to rewrite the information given in the question as an answer.

General comments

Candidates need to read examination paper questions carefully before writing an answer.

For example, in **Questions 4(a)** and **4(c)(ii)** the answer must be shown as a sum of products. In **Question 7(b)** only the items used when sending the message are required. In **Question 8(c)(i)** only the instructions set out in the question can be used in the answer.

Comments on specific questions

Question 1

(a) Most candidates found the correct denary value and showed their working. A common error was to move the binary point in the wrong direction.

(b) (i) Many candidates found the correct denary value and showed their working. A common error was to incorrectly assume that the position of the binary point had moved.

(ii) A minority of candidates were able to identify problems caused by the change in representation of the floating-point number. A common error was to incorrectly identify a decrease in precision as a problem.

Question 2

(a) Most candidates correctly classified at least four data types. A common error was to incorrectly classify integer as a composite data type.

(ii) Some candidates correctly identified the data type and the classification. A common error was to confuse type and classification.

(ii) Most candidates wrote correct pseudocode declaration and assignment statements.

Question 3

(a) Most candidates correctly identified that protocols were a set of rules. Few candidates included that the rules were for the transmission of data. A common error was to repeat the information about communication given in the question.

(b) Most candidates correctly identified two protocols used in the transfer of emails and stated the purpose of each protocol.

(c) (i) Most candidates correctly identified the method as circuit switching. Some candidates provided a good description of the method. A common error was incorrectly state advantages of circuit switching instead of a description.

(ii) Many candidates found the identification of a benefit and a drawback challenging. An example of an acceptable answer is 'Benefit: Manav and Miora can see and hear each other with better synchronisation. Drawback: extra time needed to set up dedicated circuit at the start of the video conversation.'

Question 4

(a) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks. A common error was to incorrectly write $\overline{A.B.C}$ for $\overline{A}.\overline{B}.\overline{C}$

(b) Many responses showed a correctly completed Karnaugh Map.

(c) (i) Most responses showed correct grouping. The most common error was to incorrectly group the three products in the bottom row of OUTPUT Y.

(ii) Most candidates provided a correct simplified sum-of-products for the answer to **part (c)(i)**. Some candidates did not attempt this part of **Question 4**.

Question 5

Many candidates correctly completed all the statements about flip-flops. A common error was to provide an incorrect number of stable states.

Question 6

(a) A minority of candidates provided acceptable descriptions of the roles of the four operating systems shown in the diagram.

(b) A minority of candidates provided acceptable descriptions of the role of the virtual machine software in testing MyApp.

(c) A minority of candidates provided an acceptable explanation of a benefit and drawback of this approach to testing MyApp. An example of an acceptable answer is 'Benefit: Only one set of hardware is required to test multiple operating systems, this reduces the cost of producing MyApp. Drawback: extra code is executed because of the hardware emulation; this means that accurate judgements about response time cannot be made.'

Question 7

(a) A minority of candidates provided an acceptable explanation of the process required to obtain a digital certificate. A common error was to provide an explanation of the production of a digital signature.

(b) Most candidates were able to identify two items included in a digital certificate; many candidates correctly chose items that would be used when Sam sent his message.

(c) Most candidates were able to identify two other items included in a digital certificate.

(d) Many candidates provided an explanation of the production of a digital signature. A common error was not to be specific about the key used to encrypt the message digest and produce the digital signature for Sam's confidential message; the key used must be Sam's private key.

Question 8

(a) Many candidates stated the correct information stored in the addresses. A common error was not to use the placement of the binary point to find the number of litres of fuel.

(b) Many candidates correctly converted the information given to the correct binary form for storage. A common error was not to use the placement of the binary point in the conversion of the number of litres of fuel.

(c) (i) Some candidates wrote assembly language instructions chosen from the table shown on the examination paper to complete the task set. A common error was to use an instruction not given in the table. Some candidates did not attempt this part of **Question 8**.

(ii) Most candidates completed the assembly language program. A common error was to not ensure that the process was continuous and jump to another routine rather than the start of the program.

COMPUTER SCIENCE

Paper 9608/33
Written Paper

Key messages

Candidates need to show an in-depth study of the syllabus topics and make good use of the appropriate technical terminology required to answer questions this paper. Candidates, who have studied the theory and have also practised the precise use of these 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 ensure that they provide the information asked for in the question set. It is not enough to rewrite the information given in the question as an answer.

General comments

Candidates need to read examination paper questions carefully before writing an answer.

For example, in **Question 2(a)(i)**, the answer must include only data types shown in the pseudocode shown in the question. In **Question 3(a)(i)**, the answer must be a description of the TCP/IP protocol suite. In **Questions 4(a)** and **4(c)(ii)**, the answer must be shown as a sum of products. In **Question 7(c)(i)**, only the instructions set out in the question can be used in the answer.

Comments on specific questions

Question 1

(a) Most candidates found the correct denary value and showed their working. A common error was to move the binary point in the wrong direction.

(b) (i) Most candidates gave the correct answer for the mantissa; the exponent proved to be more of a challenge for many candidates. A common error was to write `0011` instead of `0010` for the exponent.

(ii) A minority of candidates provided a full description of a problem that could occur when floating-point numbers are not normalised. Many candidates identified the problem of lack of precision, a few candidates added that this was due to redundant leading zeros in the mantissa.

Question 2

(a) (i) Many candidates identified some composite and non-composite data types. Fewer candidates correctly identified those data types that were used in the pseudocode shown in the question.

(ii) This question was generally answered well.

(b) Some candidates wrote correct pseudocode assignment statements to store the details of the box given in the question. One common error was to not store the values in the first element of the array `myBox[]`; another common error was the incorrect use of quotation marks for the values in the assignment statements

Question 3

(a) (i) Some candidates gave a good description of the TCP/IP protocol suite. Describing the TCP/IP protocol suite proved to be more of a challenge for many candidates. A common error was to describe the operation of TCP and/or IP.

(ii) Most candidates correctly identified the protocol as BitTorrent, and many candidates gave an acceptable description of the way BitTorrent protocol works.

(b) (i) Most candidates correctly identified the method as packet switching. Many candidates provided a good description of this method.

(ii) Many candidates found the identification of a benefit and a drawback challenging. An example of an acceptable answer is 'Benefit: Packets of data from the movie can be rerouted if there is a problem with the route selected. Drawback: Packets can be delayed as different routes are taken.'.

Question 4

(a) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks. A common error was to incorrectly write $\overline{A.B}.C$ for $\overline{A}.\overline{B}.C$

(b) Many responses showed a correctly completed Karnaugh Map.

(c) (i) Nearly all responses showed correct grouping. The most common error was to incorrectly group the three products in the bottom row.

(ii) Most candidates provided a correct simplified sum-of-products for the answer to **part (c)**.

(d) Many candidates correctly identified the uses of X and Y. Some candidates identified the logic circuit. A common error was to give the answer for the logic circuit as a half adder and not a full adder.

Question 5

Many candidates correctly completed most of the statements about a virtual machine. A common error was to insert the word virtual in the gaps in the statements.

Question 6

(a) Many candidates gave some similarities and differences.

(b) Some candidates gave some similarities and differences. A common error was to state that a digital signature was a scan of a handwritten signature.

(c) Many candidates gave some similarities and differences. A common error was to confuse the terms phishing and pharming.

Question 7

(a) Many candidates stated the correct information stored in the addresses. A common error was not to use the placement of the binary point to find the number of litres of fuel.

(b) Many candidates correctly converted the information given to the correct binary form for storage. A common error was not to use the placement of the binary point in the conversion of the number of litres of fuel.

(c) (i) Some candidates wrote assembly language instructions chosen from the table shown on the examination paper to complete the task set. A common error was to use an instruction not given in the table. Some candidates did not attempt this part of **Question 7**.

(ii) Most candidates completed the assembly language program. A common error was to not ensure that the process was continuous and jump to another routine rather than the start of the program.

COMPUTER SCIENCE

Paper 9608/41
Written Paper

Key messages

Candidates need to answer using one of the three approved languages; Python, VB.NET or Pascal. There were several candidates attempting to use Java which is not a permitted language for this 9608 syllabus.

Candidates need to prepare well for the written description, or explanation, questions as well as the programming and practical questions. Answers to these questions often lacked the required detail or precision.

Candidates need to read each question carefully to make sure they are answering all its requirements.

General comments

Candidates using object-oriented programming demonstrated a good understanding of the use of constructors and get methods. However, candidates found instantiation of objects challenging. They need to have sufficient experience of creating and using objects.

Candidates demonstrated a good understanding of declaration programming and assembly language programming, with many candidates giving accurate responses.

Comments on specific questions

Question 1

Many candidates had a good understanding of GANTT charts and were able to correctly complete the table. Some candidates did not appear familiar with the GANTT chart. Common errors included starting each task from week number 1 and not making use of dependencies. Another common error was not detailing all weeks when an event can take place, i.e. just showing the end week.

Question 2

(a) Candidates found this question challenging. The most common correct answer was given for beta testing. Few candidates gave all three ticks for integration testing, and acceptance testing was commonly given a tick under being tested in-house. Candidates should have a clear understanding of the different types of testing required in the syllabus and should be able to identify their purpose and who conducts these.

(b) This question was answered well many candidates giving a suitable method of testing; most commonly white-box or black-box testing. There were a range of possible answers and other notable ones included stub testing and module testing. Some candidates did not read the requirement for one ‘other’ method; this indicates that they cannot give an answer already given in the question i.e. integration, acceptance, alpha or beta.

Question 3

(a) More candidates appeared familiar with the purpose of constructors and the appropriate syntax for their chosen language. When answering questions where candidates need to write a program/algorithm to meet a criterion, they must ensure they are carefully checking the requirements. In this case, the constructor did not take any parameters, but set the attributes to

specific values. Some candidates did not read this requirement and therefore used parameters instead. A common error was not clearly showing an empty string, some candidates left a large space between their speech marks to indicate the string, which could have been read as a space symbol. Although the speech marks do not need to be directly adjacent, it must be able to be differentiated from a space.

(b) Many candidates demonstrated a good understanding of the purpose and function of a get method. Some common errors included sending a parameter to the function and then returning this parameter or overwriting the stored player ID before returning it. Where the chosen programming language differentiates between procedures and functions, candidates must make sure they select the appropriate one; in this case a function due to it returning a value.

(c) This question provided candidates with functions to use to return the requirements elements of the string, and most candidates made appropriate use of these. Some candidates used other functions that gave equivalent results.

Some candidates were not able to use the functions appropriately, for example using the comparison symbol incorrectly e.g. `length = newPlayerID(6)`.

The description of the algorithm required the parameter to set the `PlayerID` attribute. This was often left out by candidates, who returned a value but did not set the attribute to the value.

The algorithm also required a return value of `TRUE` or `FALSE`. A common error was to not return the value, but instead it was either output, or stored in a variable and then not used.

(d) (i) Candidates found this question challenging, with many finding it difficult to accurately explain containment. A common error was stating that a class contains another class in its definition; this is not precise enough because the entire class is not within its definition, an object in the class is of a different class. The whole class is not contained within another. Stronger responses made use of the example by defining the term through the application in this question.

(ii) Many candidates could give the name of another feature. Fewer were able to describe it suitably. The most common answer was inheritance; however, the description was often insufficient i.e. stating that one class has the attributes of another, without any reference to the use of super/sub classes.

(iii) The declaration of this array required three elements that were given in the question; the identifier, number of elements and data type. Most candidates were able to identify the correct number of elements and the identifier. Fewer gave the appropriate data type.

(iv) Candidates found this question challenging. They were required to instantiate an object and assign it to the first array element. There were three values that candidates had to pass to the constructor that were given in the question. Candidates had to consider the data type of the values and use these appropriately, i.e. strings were required to be in speech marks, and the 10 was an integer value. Candidates should be familiar with creating objects in their chosen language. Many were unable to reproduce this in their answers. The final part was to assign it to the array they had declared in the previous question, this required an understanding of arrays. A common error was attempting to use it is as a record e.g. `Quiz.QuizBank = … .`

Question 4

(a) Many candidates demonstrated a good understanding of declaration programming. They were able to correctly identify the required clauses and write them appropriately. Candidates should be familiar with the use of upper-case and lower-case characters, and that the clauses should all make use of lower-case letters; a common error was the use of a capital letter as the first letter for France.

(b) This question had a range of responses. Some candidates were able to accurately write the rule, using appropriate statements and logical operators. Some candidates attempted to add further elements beyond those required that often duplicated their previous statements, or at times contradicted them. A common error was writing Stilton and England within the rule e.g. `origin(stilton, england)` instead of using the given variables `X` and `Y`.

Question 5

(a) This question required an explanation of how an insertion sort works. Candidates found this challenging to explain. Some candidates described each element of an algorithm to perform the sort. This did not explain the principles of the sort, instead describing variables and how they changed as opposed to how they related to the processes involved. The stronger responses made use of an example to illustrate their explanation, for example, creating a short set of numbers and showing the movements of each stage within their explanation. This added clarity to their answers.

Some answers were not clear enough to differentiate them from other sorting algorithms, for example a description of comparing items in pairs to the end of the array could be read as a bubble sort. The importance difference in an insertion sort is the comparison of the item within the already sorted list.

(b) This question was answered well. Fewer candidates were able to identify the correct 'for loop' end value, with many candidates giving Counter as the end condition which would then loop too many times. Another common error was the assignment of values to `DataSwapped` which were often the wrong way around. The loop continues until there have been no swapped, therefore, each time there is a swap, `DataSwapped` needs to be assigned to `TRUE` because it is initialised as `FALSE` within the loop.

Question 6

(a) Candidates needed to read this question carefully because the value **E** needed to be added between **B** and **C**; a common error was attempting to add it after the value **D**. Candidates also needed to consider the use of the null pointer and where this would be appropriate in the new linked list. Another common error was not including the null pointer, and therefore not indicating where the linked list ends.

(b) This question was answered well with many candidates giving an appropriate response. A common error was stating that there is no data; however, the null pointer is data itself and therefore this was inaccurate.

(c) (i) Most candidates were able to follow this algorithm and completed at least some of the missing code. The returning of the `CurrentPointer` was most often correct. Candidates need to be careful with their use of upper case and lower case when answering these questions, as well as the use of spaces within identifiers; for example, a common error was `currentpointer` in place of `CurrentPointer`.

(ii) There were many good attempts at answering this question, often making use of **Question 6(c)(i)** to structure their answer in terms of the use of pointers and the linked list. The most common awarded mark points were for correctly following the pointers through the linked list to find either the end of the value. Few candidates were able to then correctly remove the node by changing the pointers appropriately. A common error was to not follow the pointers, but instead to start at pointer 0 and then increment this each time.

Question 7

This question was answered well with many candidates gaining at least several marks. There were a range of approaches, with some candidates using the labels for memory spaces (e.g. length) and others making use of the binary values instead. Where a binary value is given, it must be proceeded by the correct symbol e.g. B for binary.

Candidates need to be able to differentiate between the different modes of addressing and how these affect the code. For example, common errors included using STO where STX was required, and vice-versa.

Some candidates attempted to complete comment boxes that did not have any text in, this was not required by the question which asked candidates to complete the program i.e. the instructions.

Question 8

(a) This question was answered well with many candidates being able to correctly identify the line where the recursive call was made, and then justifying this choice.

(b) Candidates found tracing this recursive algorithm challenging. Most made good use of the trace table, but few were able to follow the algorithm to the end. The forwards runs were most commonly correct, but then candidates could not unwind the algorithm making use of the return values. Some candidates stopped when the algorithm reached its end and did not attempt to unwind it. Candidates could have completed this using the trace table and/or the working space depending on their preference, but the majority made use of the table.

(c) Candidates who had correctly traced the algorithm were often able to explain what the algorithm did. Some candidates did not give enough detail to be awarded the mark by making use of generic statements such as adding previous values, without the precision of adding the two previous values.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

Candidates need to answer using one of the three approved languages; Python, VB.NET or Pascal. There were several candidates attempting to use Java which is not a permitted language for this 9608 syllabus.

Candidates need to prepare well for the written description, or explanation, questions as well as the programming and practical questions. Answers to these questions often lacked the required detail or precision.

General comments

Candidates using object-oriented programming demonstrated a good understanding of the use of constructors and get methods. However, candidates found instantiation of objects challenging. They need to have sufficient experience of creating and using objects.

Candidates demonstrated a good understanding of declaration programming and assembly language programming, with many candidates giving accurate responses.

Comments on specific questions

Question 1

(a) Many responses to this question were correct, with bubble sort being the most common answer given. Some candidates were unable to identify any sorting algorithms, with a common error being searching algorithms given instead.

(b) (i) Candidates were usually able to either follow the algorithm correctly or were unable to follow any part of the algorithm. Common errors included incorrectly calculating the bounds and changing `ValueToFind` during the algorithm. Some candidates need to follow the algorithm without making assumptions about its function at the start. This will allow them to take each line one at a time, implement it and update the trace table before moving to the next line.

(ii) Candidates who were able to trace the algorithm were usually able to give the correct answer.

(iii) Candidates found this question challenging. The question required candidates to consider when the loop condition was executed; this did not mean that the code inside the loop itself would run, but only that the condition would be checked. This led to some candidates giving one too few as a result.

(iv) As with the previous question candidates often considered the number of times the whole loop ran as opposed to just when the loop condition was executed, leading to a value one too high being given as the answer.

(v) This question required an understanding of when the loop terminated, and which items were being checked each time. Some candidates gave responses that were not precise enough, for example; to check all values in the list, or to make sure the value is found. These did not relate sufficiently to the code and the position of the elements. Some candidates may have found it useful to run the

algorithm several times with different values in the list of elements to identify when this final check is used.

(vi) Many candidates were able to correctly identify that the data items needed to be in order, for the algorithm to work. Few were able to explain why this is the case; i.e. why does it need to be in order.

Question 2

(a) Responses to this question were mixed, with many candidates being able to correctly produce a suitable GANTT chart. Some candidates did not appear familiar with the concept of a GANTT chart and were unable to make use of the dependencies. It is important that candidates complete all boxes where activities can take place; a common error was only identifying the end position of each activity and not showing where they start.

(b) (i) Candidates found this question challenging and few were able to correct identify all the activities that form the critical path, with often additional tasks being given.

(ii) This question was answered well with many candidates being able to correct identify three activities that can run in parallel.

(c) Many candidates incorrectly identified computational thinking methods such as decomposition as a project planning technique.

Question 3

(a) Most candidates were able to demonstrate a satisfactory understanding of declarative languages and were able to give some correct clauses. Candidates should be familiar with the differences between upper-case and lower-case characters in these languages, and therefore should not be using capital letters when writing clauses. Some candidates attempted to add an additional clause that a gecko was an animal; this clause was already given in the question (clause number 07).

(b) This question was answered well with many candidates giving the appropriate answer.

(c) Many candidates were able to gain several marks in this question, making correct use of the variables. A common error included adding additional statements beyond those required that sometimes contradicted previous rules that they had given. Candidates should be familiar with the use of logical operators in declarative languages, and hence be able to use AND OR and NOT appropriately.

Question 4

(a) Many candidates were familiar with their language specific constructor and were able to give the appropriate code for this. Candidates need to make sure they are reading the program requirements carefully to determine whether values are passed to the constructor as parameters, and/or whether they are set to default values. This problem had a combination where one item was passed as a parameter whilst the others had default values. A common error was passing three elements to the constructor and hence not making use of the default values. When candidates are declaring an empty string, they must make sure there is no clear space between the values, because a space has a string value and therefore is not an empty string.

(b) Many candidates were also more familiar with the purpose of get methods and the syntax of their chosen language. Some candidates still incorrectly send a parameter to the get method and then return this value or override the attribute before returning.

(c) This question was generally answered well. Many candidates were able to give an appropriate selection statement to check the value. Fewer candidates passed the value being compared as a parameter. Candidates should be familiar with the need for unique identifiers; that if an attribute is declared as Calories then the parameter being passed should not have the same identifier.

The method required either `TRUE` or `FALSE` to be returned. Some candidates assigned the value to a variable but did not then return this value at any point in the algorithm.

The method also required the parameter value to be assigned to the attribute only when valid; some candidates either assigned this in both conditions, or did not assign the value at all. Some candidates assigned the value after returning the value; this code would not run because it would return before it could be executed.

(d) (i) Responses to this question were mixed. Many candidates were not precise enough; for example stating that private means the attributes cannot be accessed is incorrect, because they can be accessed but only through the methods. Similarly, saying they cannot be accessed outside the class is also inaccurate, because they can be accessed through the methods.

(ii) Most candidates understood the concept of inheritance. Fewer were able to express this accurately. A common response was that a class can 'inherit attributes from another class' – this is defining inheritance using the word inherit which does not explain what this mean. This response also just states it is from another class, without any reference to one class begin the parent/super class and the other the child/sub class.

Both **parts (i)** and **(ii)** required application to the given program, and candidates who referred to these classes from the start often gave better response through their explanation of what was inherited and what this meant for the two classes.

(iii) Candidates found the description of polymorphism challenging. Some candidates confused it with containment, or encapsulation. Those who demonstrate an understanding of polymorphism were often able to explain it appropriately. This question also required application to the problem and candidates who used this from the start often gave better responses.

(e) Candidates found it challenging to explain the concepts of imperative programming, with many giving examples of concepts that may be found within it; for example 'you can write functions', which you can do in other types of programming as well as imperative. The most common response was that it gives step-by-step instructions. Many candidates gave examples of imperative programming languages which was not answering the given question.

(f) (i) This question was answered well with many candidates giving an appropriate type of testing.

(ii) Many candidates incorrectly gave alpha or beta testing as a response. Alpha testing is done within the company and therefore not to prove to the customer it works and meets requirements. Beta testing is where potential end users use the software, and therefore it is not when it is shown to the customer to prove it works.

(iii) Candidates should have experience of writing and completing test plans for programs. Few were able to identify appropriate items that would appear in a test plan. Some candidates attempted to give precise examples of data that would be included as opposed to general items that are included.

Question 5

This question was answered well. Many candidates were able to give correct responses to multiple lines. There were a range of approaches, with some candidates using the labels for memory spaces (e.g. `LENGTH`) and others making use of the binary values instead. Where a binary value is given, it must be proceeded by the correct symbol e.g. B for binary.

Candidates need to be able to differentiate between the different modes of addressing and how these affect the code. For example, common errors included using LDM where LDR was required, and vice-versa.

Some candidates attempted to complete comment boxes that did not have any text in, this was not required by the question.

Question 6

(a) This question was often answered well with many candidates being able to correct remove node C. There were a range of approaches to the linking of the nodes which were all valid e.g. node C

having no points, node C having no pointer to it but still pointing to node D, or node C having a null pointer.

(b) Candidates found this question challenging. Many simply repeated **part (a)**, that it was removed from the list, without answering the question 'what happens to the node'. When a node is removed, it is moved to the free list.

(c) This question was answered well. Most candidates were able to identify that the null pointer signified the end of the list, or that there was no data after node D.

(d) Many candidates were able to write an at least partially appropriate algorithm for `FindValue()`. Most commonly candidates could compare the data suitably to check if the value was found, or if it was empty. Candidates need to check the requirements of the problem carefully, especially the values that need to be returned and must make sure all these are met.

(e) Most candidates were able to identify a suitable abstract data type. Many of these could give a suitable description of some features of the data type, but few were able to give enough information. Many stopped short, particularly when describing stacks and queues, where the candidates stated if they were LIFO or FIFO, and then expanded this, or repeated this in a different way; i.e. they stated the same point multiple times.

COMPUTER SCIENCE

Paper 9608/43
Written Paper

Key messages

Candidates need to answer using one of the three approved languages; Python, VB.NET or Pascal. There were several candidates attempting to use Java which is not a permitted language for this 9608 syllabus.

Candidates need to prepare well for the written description, or explanation, questions as well as the programming and practical questions. Answers to these questions often lacked the required detail or precision.

Candidates need to read each question carefully to make sure they are answering all its requirements.

General comments

Candidates using object-oriented programming demonstrated a good understanding of the use of constructors and get methods. However, candidates found instantiation of objects challenging. They need to have sufficient experience of creating and using objects.

Candidates demonstrated a good understanding of declaration programming and assembly language programming, with many candidates giving accurate responses.

Comments on specific questions

Question 1

Many candidates had a good understanding of GANTT charts and were able to correctly complete the table. Some candidates did not appear familiar with the GANTT chart. Common errors included starting each task from week number 1 and not making use of dependencies. Another common error was not detailing all weeks when an event can take place, i.e. just showing the end week.

Question 2

(a) Candidates found this question challenging. The most common correct answer was given for beta testing. Few candidates gave all three ticks for integration testing, and acceptance testing was commonly given a tick under being tested in-house. Candidates should have a clear understanding of the different types of testing required in the syllabus and should be able to identify their purpose and who conducts these.

(b) This question was answered well many candidates giving a suitable method of testing; most commonly white-box or black-box testing. There were a range of possible answers and other notable ones included stub testing and module testing. Some candidates did not read the requirement for one ‘other’ method; this indicates that they cannot give an answer already given in the question i.e. integration, acceptance, alpha or beta.

Question 3

(a) More candidates appeared familiar with the purpose of constructors and the appropriate syntax for their chosen language. When answering questions where candidates need to write a program/algorithm to meet a criterion, they must ensure they are carefully checking the requirements. In this case, the constructor did not take any parameters, but set the attributes to

specific values. Some candidates did not read this requirement and therefore used parameters instead. A common error was not clearly showing an empty string, some candidates left a large space between their speech marks to indicate the string, which could have been read as a space symbol. Although the speech marks do not need to be directly adjacent, it must be able to be differentiated from a space.

(b) Many candidates demonstrated a good understanding of the purpose and function of a get method. Some common errors included sending a parameter to the function and then returning this parameter or overwriting the stored player ID before returning it. Where the chosen programming language differentiates between procedures and functions, candidates must make sure they select the appropriate one; in this case a function due to it returning a value.

(c) This question provided candidates with functions to use to return the requirements elements of the string, and most candidates made appropriate use of these. Some candidates used other functions that gave equivalent results.

Some candidates were not able to use the functions appropriately, for example using the comparison symbol incorrectly e.g. `length = newPlayerID(6)`.

The description of the algorithm required the parameter to set the `PlayerID` attribute. This was often left out by candidates, who returned a value but did not set the attribute to the value.

The algorithm also required a return value of `TRUE` or `FALSE`. A common error was to not return the value, but instead it was either output, or stored in a variable and then not used.

(d) (i) Candidates found this question challenging, with many finding it difficult to accurately explain containment. A common error was stating that a class contains another class in its definition; this is not precise enough because the entire class is not within its definition, an object in the class is of a different class. The whole class is not contained within another. Stronger responses made use of the example by defining the term through the application in this question.

(ii) Many candidates could give the name of another feature. Fewer were able to describe it suitably. The most common answer was inheritance; however, the description was often insufficient i.e. stating that one class has the attributes of another, without any reference to the use of super/sub classes.

(iii) The declaration of this array required three elements that were given in the question; the identifier, number of elements and data type. Most candidates were able to identify the correct number of elements and the identifier. Fewer gave the appropriate data type.

(iv) Candidates found this question challenging. They were required to instantiate an object and assign it to the first array element. There were three values that candidates had to pass to the constructor that were given in the question. Candidates had to consider the data type of the values and use these appropriately, i.e. strings were required to be in speech marks, and the 10 was an integer value. Candidates should be familiar with creating objects in their chosen language. Many were unable to reproduce this in their answers. The final part was to assign it to the array they had declared in the previous question, this required an understanding of arrays. A common error was attempting to use it is as a record e.g. `Quiz.QuizBank = … .`

Question 4

(a) Many candidates demonstrated a good understanding of declaration programming. They were able to correctly identify the required clauses and write them appropriately. Candidates should be familiar with the use of upper-case and lower-case characters, and that the clauses should all make use of lower-case letters; a common error was the use of a capital letter as the first letter for France.

(b) This question had a range of responses. Some candidates were able to accurately write the rule, using appropriate statements and logical operators. Some candidates attempted to add further elements beyond those required that often duplicated their previous statements, or at times contradicted them. A common error was writing Stilton and England within the rule e.g. `origin(stilton, england)` instead of using the given variables `X` and `Y`.

Question 5

(a) This question required an explanation of how an insertion sort works. Candidates found this challenging to explain. Some candidates described each element of an algorithm to perform the sort. This did not explain the principles of the sort, instead describing variables and how they changed as opposed to how they related to the processes involved. The stronger responses made use of an example to illustrate their explanation, for example, creating a short set of numbers and showing the movements of each stage within their explanation. This added clarity to their answers.

Some answers were not clear enough to differentiate them from other sorting algorithms, for example a description of comparing items in pairs to the end of the array could be read as a bubble sort. The importance difference in an insertion sort is the comparison of the item within the already sorted list.

(b) This question was answered well. Fewer candidates were able to identify the correct 'for loop' end value, with many candidates giving Counter as the end condition which would then loop too many times. Another common error was the assignment of values to `DataSwapped` which were often the wrong way around. The loop continues until there have been no swapped, therefore, each time there is a swap, `DataSwapped` needs to be assigned to `TRUE` because it is initialised as `FALSE` within the loop.

Question 6

(a) Candidates needed to read this question carefully because the value **E** needed to be added between **B** and **C**; a common error was attempting to add it after the value **D**. Candidates also needed to consider the use of the null pointer and where this would be appropriate in the new linked list. Another common error was not including the null pointer, and therefore not indicating where the linked list ends.

(b) This question was answered well with many candidates giving an appropriate response. A common error was stating that there is no data; however, the null pointer is data itself and therefore this was inaccurate.

(c) (i) Most candidates were able to follow this algorithm and completed at least some of the missing code. The returning of the `CurrentPointer` was most often correct. Candidates need to be careful with their use of upper case and lower case when answering these questions, as well as the use of spaces within identifiers; for example, a common error was `currentpointer` in place of `CurrentPointer`.

(ii) There were many good attempts at answering this question, often making use of **Question 6(c)(i)** to structure their answer in terms of the use of pointers and the linked list. The most common awarded mark points were for correctly following the pointers through the linked list to find either the end of the value. Few candidates were able to then correctly remove the node by changing the pointers appropriately. A common error was to not follow the pointers, but instead to start at pointer 0 and then increment this each time.

Question 7

This question was answered well with many candidates gaining at least several marks. There were a range of approaches, with some candidates using the labels for memory spaces (e.g. length) and others making use of the binary values instead. Where a binary value is given, it must be proceeded by the correct symbol e.g. B for binary.

Candidates need to be able to differentiate between the different modes of addressing and how these affect the code. For example, common errors included using STO where STX was required, and vice-versa.

Some candidates attempted to complete comment boxes that did not have any text in, this was not required by the question which asked candidates to complete the program i.e. the instructions.

Question 8

(a) This question was answered well with many candidates being able to correctly identify the line where the recursive call was made, and then justifying this choice.

(b) Candidates found tracing this recursive algorithm challenging. Most made good use of the trace table, but few were able to follow the algorithm to the end. The forwards runs were most commonly correct, but then candidates could not unwind the algorithm making use of the return values. Some candidates stopped when the algorithm reached its end and did not attempt to unwind it. Candidates could have completed this using the trace table and/or the working space depending on their preference, but the majority made use of the table.

(c) Candidates who had correctly traced the algorithm were often able to explain what the algorithm did. Some candidates did not give enough detail to be awarded the mark by making use of generic statements such as adding previous values, without the precision of adding the two previous values.