COMPUTER SCIENCE

Paper 9608/11
Written Paper

Key messages

Candidates should be encouraged to write their answers clearly in the spaces provided on the examination paper. If there is insufficient space, candidates may use additional sheets or blank spaces within the examination paper, but there is a need for an obvious indication that the answer is continued somewhere else. It is also essential, particularly in questions that require a calculation, that candidates indicate in some way the answer which is to be marked.

Candidates should be aware of the need for the correct use of the appropriate technical terminology and for precision in answering questions. At this level of study, candidates will only be given credit for detailed and specific answers.

There were two questions where candidates were asked to write SQL statements. The best way to prepare candidates for questions on this topic is to expose them to some practical work using simple databases which they can query by writing straightforward SQL scripts. Setting up the query using a QBE grid and then examining the SQL code automatically produced by the database software is not advisable, this code is unnecessarily complex for the level of answers that candidates would be expected to provide on this paper. If suitable software is not available there are a number of excellent online resources that could be used.

Some candidates continue to answer in pencil and then overwrite their answers in ink which makes some responses very difficult to read when they have been electronically scanned as black and white text. The same applies to the use of fibre tipped pens when the ink soaks through to the other side of the page.

General comments

It is very important that the question stem is read carefully and the key words highlighted. Some of these key words will indicate the type of answer required, either a single statement or more extended prose, and others will indicate the context in which the question has been set. Identifying and understanding these key words will help candidates to give more appropriate answers to the questions on the examination paper. Several of the questions on this paper, for example, **question 4**, **question 6** and **question 8** required answers or examples in a particular context and generalised responses or responses in a different context were unacceptable.

There is considerable confusion between the terms *data* and *information* and between the terms *field* and *record*. Candidates need to be aware of the difference and make sure that they use the correct terminology when answering questions.

Comments on specific questions

Question 1

This is a clear example of an instance where many candidates did not read the question properly. The question says, '*Draw lines*' ... '*to the correct one or more definitions*'. Most candidates were able to correctly connect the assembler to its answer, but only a few candidates realised that there were two correct definitions for both the compiler and the interpreter.

Question 2

It is appreciated that candidates may need space to work out the answer to questions such as **Question 2** and so there is plenty of blank space available in the examination paper. Some candidates put their working in the answer space and so obscured the answer. The number that is to be marked should be clearly visible. Unless there is an indication to the contrary, examiners will mark the first answer seen.

When answers are given in binary, it is especially important that candidates show clearly whether a particular digit is a zero or a one. If candidates change their answer and overwrite, for example a one with a zero, the result frequently looks like this, , and it is not possible to tell what it is meant to be, so no credit can be given.

(a) The majority of the candidates were able to correctly convert 55 in denary to a binary number. A small number of candidates did not include the leading zeros, when the question asked for 8-bit binary.

(b) Many candidates correctly converted the given BCD value into denary. The most common incorrect answer was 131, where candidates had not recognised that the initial value was in BCD format and had completed a straightforward binary to denary conversion.

(c) Most candidates realised that the negative denary number would result in a one in the most significant bit of the answer, however, many mistakes were made in the actual conversion. It was particularly noticeable in this part question that candidates were over-writing digits. If an answer needs to be changed, candidates should clearly cross out the first answer and completely re-write the second answer.

(d) Many candidates correctly converted the hexadecimal value to denary. The two most common causes of error were to convert the hexadecimal value to binary and then not to continue to denary or to write 414.

Question 3

(a) Please see the comments in the **General** section of this report regarding careful reading of the question. This question asked *how* special purpose registers were used. This was missed by a considerable number of candidates who wrote excellent descriptions of the special purpose registers without making any reference to how they were used in the fetch-execute cycle and thus did not answer the question on the examination paper.

(b) There were a number of completely correct answers to this question, but many candidates need to improve their understanding of how interrupts are handled during the fetch-execute cycle.

Question 4

Please see the comments in the **General** section of this report regarding context. **Part (a)** and **part (b)(i)** of the question specifically mention sound, but a number of candidate responses talked about sampling images. Images are discussed in **part (b)(ii)**.

(a) There were a number of correct descriptions for sampling. Responses which simply re-worded the question stem were common, such as '*sampling is the taking of samples'*; this is much too imprecise as a definition. The question asked for a description so additional information was required. Some candidates realised that improving the sampling rate would result in a more accurate representation of the sound but statements were often given in terms of sound quality, which is too vague. Candidates also need to be aware that in a question such as this it is important to give an indication of, for example, the time interval between each sample, or the number of samples that would be taken per second.

(b) (i) The question stated that the sound files were to be emailed to the producer, yet the majority of answers given, simply described what was meant by either lossy or lossless compression without any reference at all to the requirement that the compressed file be sent via email. A small number of candidates did not indicate which method was chosen.

(ii) This question proved challenging for many candidates. There was considerable confusion between the terms *bit* and *byte*, and many responses spoke vaguely about patterns which repeat which is far too imprecise. Candidates need to be aware of the need for the repeating colour to be in adjacent pixels, and that this repeating string is then coded into two values.

(iii) This question required the application of the method described in **part 4(b)(ii)**. It also proved challenging to many candidates. The most common mistake was for candidates to simply count up the number of Bs and Ws in each row, rather than replacing the repeating strings with the colour code and the count of the number of occurrences.

Question 5

This question was very well answered with many candidates correctly identifying the most appropriate type of software for each description.

Question 6

The majority of candidates were able to correctly state three principles of the ACM/IEEE Software Engineering Code of Ethics, with many repeating the code verbatim. Candidates need to be aware that when a question is asked in a particular context, examples given in the answer must also refer to that context; in this case, a team of software engineers developing a new e-commerce program for a client. A few candidates gave examples in the correct context; most of the examples were too generic and could have applied to any situation. There was some confusion between the code of ethics and the Data Protection Act.

Question 7

(a) Many candidates were able to correctly expand this commonly used abbreviation, but there were a considerable number of candidates who need to improve their understanding in this area.

(b) Many candidates correctly identified the IP addresses that were valid and those that were invalid, but in many cases the reasons were incomplete and too imprecise. It is not enough to say, for example, '*the numbers are all in range*', the range needs to be stated. In cases where the IP address is invalid, the value which makes it invalid needs to be identified and the reason why this value makes the address invalid stated, including any range which is exceeded. The most common incorrect identification was the hexadecimal address A:78:F4:J8 which a significant number of candidates thought was valid because they overlooked the J as an incorrect hexadecimal digit.

(c) This question proved challenging for many candidates. Answers were generally much too vague and imprecise. It is not enough to say, for example, that '*public addresses can be seen by everyone*'. Candidates need to improve their understanding of the differences between public and private IP addresses, particularly in the context of the wider internet and not just in the situation where a home user connects to the internet via a domestic router.

Question 8

(a) (i) Many candidates were able to correctly expand this commonly used abbreviation.

(ii) A small number of candidates gave good answers to this question. There was considerable confusion between the security of the student data and the integrity of the student data, with candidates writing at length about verification and validation techniques which was not what was required. The question asked how the DBMS software could be used to ensure the security of the data, and so answers such as '*use a firewall*' did not answer the question.

(iii) Candidates need to improve their understanding of the use of a query processor, which is just one of the many features of a DBMS. Some candidates gave a good example of use, but only a very few candidates could describe the generic process of setting up search criteria in order to retrieve data.

(iv) Please see the comments in the **General** section of this report regarding careful reading of the question. This question asked '*how the DBMS has replaced software that used a file based approach*'. It is not enough to simply identify the problem with the file-based approach, as the question asked '*how*'. Further detail is needed describing the way in which the DBMS software overcomes the problem identified. Also, the question says '*replaced software that used a file-based approach*'. Many candidates overlooked the fact that the DBMS was replacing a different software package and wrote about the advantages of a DBMS over a paper-based filing system. This is also an area where the use of the correct terminology is essential. There was considerable confusion between the use of '*file*' and '*record'.*

(b) (i) This question was answered well. The majority of candidates understood how the relationship would be implemented and were able to describe it clearly.

(ii) This question was answered well. The majority of candidates understood that there was a many-to-many relationship between `CLASS` and `STUDENT` and that in order to implement this `CLASS-GROUP` would be needed, and that if the relationship between `CLASS` and `CLASS-GROUP` was one-to-many as given in the question, then the relationship between `CLASS-GROUP` and `STUDENT` would be many-to-one.

(iii) Some candidates found this question very challenging and quite a few did not attempt an answer. The best answers were those where straightforward SQL statements had been written, each statement beginning on a new line. Some candidates confused themselves by including numerous brackets which were not necessary and which often actually made the code incorrect. On this occasion minor errors in syntax and spelling were overlooked, here and in **part 8(b)(iv)**, this will not be the case in future examination series. The most common errors were the omission of quotation marks around the "`10B`", sorting the `LastName` descending instead of ascending and forgetting the final semi-colon.

(iv) As **part 8(b)(iii)**, some candidates also found this question very challenging and did not attempt an answer. Many of the remarks for **question 8(b)(iii)** also apply here. This was a slightly more complex query involving two tables, but is a standard technique and should have caused little difficulty to well-prepared candidates. The solution could have been written in a number of ways; any correct method of connecting the two tables was accepted. One of the most common mistakes was the omission of the quotation marks around the `ClassID`. Another frequent error was the reversal of the table name and field name, when using dotted notation, so for example, writing `LastName.STUDENT` when it should be `STUDENT.LastName`. The final semi-colon was often missing.

Question 9

(a) (i) There were many fully correct answers to this question. The vast majority of candidates understood what was meant by indexed addressing and were able to correctly show the contents of the accumulator and describe how the answer was obtained. Candidates who showed an incorrect value in the accumulator, had frequently correctly added the contents of the index register to the base address, but then, instead of looking at the contents of address 68, had converted 68 in denary to its binary equivalent and entered that binary value into the accumulator.

(ii) The question was answered well. Most candidates correctly decremented the value in the index register.

(b) Most candidates correctly completed the first three instructions and the incrementing of the index register and there were a good number of completely correct answers to this question. The most frequent causes of error were the omission of the instruction addresses in the first column and the incorrect interpretation of the OUT instruction, outputting 120 instead of 'x'.

This question was one where many candidates had initially answered in pencil and then had overwritten their answers in ink. When scanned electronically this creates a double image which is sometimes very difficult to read. There is plenty of blank space in the question paper and candidates should be encouraged to do their rough working on these blank pages so that the answers to be marked are clear and can be easily read.

COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

Candidates should be encouraged to write their answers clearly in the spaces provided on the examination paper. If there is insufficient space, candidates may use additional sheets or blank spaces within the examination paper, but there is a need for an obvious indication that the answer is continued somewhere else. It is also essential, particularly in questions that require a calculation, that candidates indicate in some way the answer which is to be marked.

Candidates should be aware of the need for the correct use of the appropriate technical terminology and for precision in answering questions. At this level of study, candidates will only be given credit for detailed and specific answers.

There were two questions where candidates were asked to write SQL statements. The best way to prepare candidates for questions on this topic is to expose them to some practical work using simple databases which they can query by writing straightforward SQL scripts. Setting up the query using a QBE grid and then examining the SQL code automatically produced by the database software is not advisable, this code is unnecessarily complex for the level of answers that candidates would be expected to provide on this paper. If suitable software is not available there are a number of excellent online resources that could be used.

Some candidates continue to answer in pencil and then overwrite their answers in ink which makes some responses very difficult to read when they have been electronically scanned as black and white text. The same applies to the use of fibre tipped pens when the ink soaks through to the other side of the page.

General comments

It is very important that the question stem is read carefully and the key words highlighted. Some of these key words will indicate the type of answer required, either a single statement or more extended prose, and others will indicate the context in which the question has been set. Identifying and understanding these key words will help candidates to give more appropriate answers to the questions on the examination paper. Several of the questions on this paper, for example, **question 4**, **question 6** and **question 8** required answers or examples in a particular context and generalised responses or responses in a different context were unacceptable.

There is considerable confusion between the terms *data* and *information* and between the terms *field* and *record*. Candidates need to be aware of the difference and make sure that they use the correct terminology when answering questions.

Comments on specific questions

Question 1

This is a clear example of an instance where many candidates did not read the question properly. The question says, '*Draw lines*' ... '*to the correct one or more definitions*'. Most candidates were able to correctly connect the assembler to its answer, but only a few candidates realised that there were two correct definitions for both the compiler and the interpreter.

Question 2

It is appreciated that candidates may need space to work out the answer to questions such as **Question 2** and so there is plenty of blank space available in the examination paper. Some candidates put their working in the answer space and so obscured the answer. The number that is to be marked should be clearly visible. Unless there is an indication to the contrary, examiners will mark the first answer seen.

When answers are given in binary, it is especially important that candidates show clearly whether a particular digit is a zero or a one. If candidates change their answer and overwrite, for example a one with a zero, the result frequently looks like this, , and it is not possible to tell what it is meant to be, so no credit can be given.

(a) The majority of the candidates were able to correctly convert 55 in denary to a binary number. A small number of candidates did not include the leading zeros, when the question asked for 8-bit binary.

(b) Many candidates correctly converted the given BCD value into denary. The most common incorrect answer was 131, where candidates had not recognised that the initial value was in BCD format and had completed a straightforward binary to denary conversion.

(c) Most candidates realised that the negative denary number would result in a one in the most significant bit of the answer, however, many mistakes were made in the actual conversion. It was particularly noticeable in this part question that candidates were over-writing digits. If an answer needs to be changed, candidates should clearly cross out the first answer and completely re-write the second answer.

(d) Many candidates correctly converted the hexadecimal value to denary. The two most common causes of error were to convert the hexadecimal value to binary and then not to continue to denary or to write 414.

Question 3

(a) Please see the comments in the **General** section of this report regarding careful reading of the question. This question asked *how* special purpose registers were used. This was missed by a considerable number of candidates who wrote excellent descriptions of the special purpose registers without making any reference to how they were used in the fetch-execute cycle and thus did not answer the question on the examination paper.

(b) There were a number of completely correct answers to this question, but many candidates need to improve their understanding of how interrupts are handled during the fetch-execute cycle.

Question 4

Please see the comments in the **General** section of this report regarding context. **Part (a)** and **part (b)(i)** of the question specifically mention sound, but a number of candidate responses talked about sampling images. Images are discussed in **part (b)(ii)**.

(a) There were a number of correct descriptions for sampling. Responses which simply re-worded the question stem were common, such as '*sampling is the taking of samples'*; this is much too imprecise as a definition. The question asked for a description so additional information was required. Some candidates realised that improving the sampling rate would result in a more accurate representation of the sound but statements were often given in terms of sound quality, which is too vague. Candidates also need to be aware that in a question such as this it is important to give an indication of, for example, the time interval between each sample, or the number of samples that would be taken per second.

(b) (i) The question stated that the sound files were to be emailed to the producer, yet the majority of answers given, simply described what was meant by either lossy or lossless compression without any reference at all to the requirement that the compressed file be sent via email. A small number of candidates did not indicate which method was chosen.

(ii) This question proved challenging for many candidates. There was considerable confusion between the terms *bit* and *byte*, and many responses spoke vaguely about patterns which repeat which is far too imprecise. Candidates need to be aware of the need for the repeating colour to be in adjacent pixels, and that this repeating string is then coded into two values.

(iii) This question required the application of the method described in **part 4(b)(ii)**. It also proved challenging to many candidates. The most common mistake was for candidates to simply count up the number of Bs and Ws in each row, rather than replacing the repeating strings with the colour code and the count of the number of occurrences.

Question 5

This question was very well answered with many candidates correctly identifying the most appropriate type of software for each description.

Question 6

The majority of candidates were able to correctly state three principles of the ACM/IEEE Software Engineering Code of Ethics, with many repeating the code verbatim. Candidates need to be aware that when a question is asked in a particular context, examples given in the answer must also refer to that context; in this case, a team of software engineers developing a new e-commerce program for a client. A few candidates gave examples in the correct context; most of the examples were too generic and could have applied to any situation. There was some confusion between the code of ethics and the Data Protection Act.

Question 7

(a) Many candidates were able to correctly expand this commonly used abbreviation, but there were a considerable number of candidates who need to improve their understanding in this area.

(b) Many candidates correctly identified the IP addresses that were valid and those that were invalid, but in many cases the reasons were incomplete and too imprecise. It is not enough to say, for example, '*the numbers are all in range*', the range needs to be stated. In cases where the IP address is invalid, the value which makes it invalid needs to be identified and the reason why this value makes the address invalid stated, including any range which is exceeded. The most common incorrect identification was the hexadecimal address A:78:F4:J8 which a significant number of candidates thought was valid because they overlooked the J as an incorrect hexadecimal digit.

(c) This question proved challenging for many candidates. Answers were generally much too vague and imprecise. It is not enough to say, for example, that '*public addresses can be seen by everyone*'. Candidates need to improve their understanding of the differences between public and private IP addresses, particularly in the context of the wider internet and not just in the situation where a home user connects to the internet via a domestic router.

Question 8

(a) (i) Many candidates were able to correctly expand this commonly used abbreviation.

(ii) A small number of candidates gave good answers to this question. There was considerable confusion between the security of the student data and the integrity of the student data, with candidates writing at length about verification and validation techniques which was not what was required. The question asked how the DBMS software could be used to ensure the security of the data, and so answers such as '*use a firewall*' did not answer the question.

(iii) Candidates need to improve their understanding of the use of a query processor, which is just one of the many features of a DBMS. Some candidates gave a good example of use, but only a very few candidates could describe the generic process of setting up search criteria in order to retrieve data.

(iv) Please see the comments in the **General** section of this report regarding careful reading of the question. This question asked '*how the DBMS has replaced software that used a file based approach*'. It is not enough to simply identify the problem with the file-based approach, as the question asked '*how*'. Further detail is needed describing the way in which the DBMS software overcomes the problem identified. Also, the question says '*replaced software that used a file-based approach*'. Many candidates overlooked the fact that the DBMS was replacing a different software package and wrote about the advantages of a DBMS over a paper-based filing system. This is also an area where the use of the correct terminology is essential. There was considerable confusion between the use of '*file*' and '*record'.*

(b) (i) This question was answered well. The majority of candidates understood how the relationship would be implemented and were able to describe it clearly.

(ii) This question was answered well. The majority of candidates understood that there was a many-to-many relationship between `CLASS` and `STUDENT` and that in order to implement this `CLASS-GROUP` would be needed, and that if the relationship between `CLASS` and `CLASS-GROUP` was one-to-many as given in the question, then the relationship between `CLASS-GROUP` and `STUDENT` would be many-to-one.

(iii) Some candidates found this question very challenging and quite a few did not attempt an answer. The best answers were those where straightforward SQL statements had been written, each statement beginning on a new line. Some candidates confused themselves by including numerous brackets which were not necessary and which often actually made the code incorrect. On this occasion minor errors in syntax and spelling were overlooked, here and in **part 8(b)(iv)**, this will not be the case in future examination series. The most common errors were the omission of quotation marks around the "`10B`", sorting the `LastName` descending instead of ascending and forgetting the final semi-colon.

(iv) As **part 8(b)(iii)**, some candidates also found this question very challenging and did not attempt an answer. Many of the remarks for **question 8(b)(iii)** also apply here. This was a slightly more complex query involving two tables, but is a standard technique and should have caused little difficulty to well-prepared candidates. The solution could have been written in a number of ways; any correct method of connecting the two tables was accepted. One of the most common mistakes was the omission of the quotation marks around the `ClassID`. Another frequent error was the reversal of the table name and field name, when using dotted notation, so for example, writing `LastName.STUDENT` when it should be `STUDENT.LastName`. The final semi-colon was often missing.

Question 9

(a) (i) There were many fully correct answers to this question. The vast majority of candidates understood what was meant by indexed addressing and were able to correctly show the contents of the accumulator and describe how the answer was obtained. Candidates who showed an incorrect value in the accumulator, had frequently correctly added the contents of the index register to the base address, but then, instead of looking at the contents of address 68, had converted 68 in denary to its binary equivalent and entered that binary value into the accumulator.

(ii) The question was answered well. Most candidates correctly decremented the value in the index register.

(b) Most candidates correctly completed the first three instructions and the incrementing of the index register and there were a good number of completely correct answers to this question. The most frequent causes of error were the omission of the instruction addresses in the first column and the incorrect interpretation of the OUT instruction, outputting 120 instead of 'x'.

This question was one where many candidates had initially answered in pencil and then had overwritten their answers in ink. When scanned electronically this creates a double image which is sometimes very difficult to read. There is plenty of blank space in the question paper and candidates should be encouraged to do their rough working on these blank pages so that the answers to be marked are clear and can be easily read.

COMPUTER SCIENCE

Paper 9608/1301
Written Paper

Key messages

Candidates should be encouraged to write their answers clearly in the spaces provided on the examination paper. If there is insufficient space, candidates may use additional sheets or blank spaces within the examination paper, but there is a need for an obvious indication that the answer is continued somewhere else. It is also essential, particularly in questions that require a calculation, that candidates indicate in some way the answer which is to be marked.

Candidates should be aware of the need for the correct use of the appropriate technical terminology and for precision in answering questions. At this level of study only detailed, specific and clear answers will be given credit.

There were three questions where candidates were asked to write SQL statements. The best way to prepare candidates for questions on this topic is by exposing them to some practical work using simple databases which they can query by writing straightforward SQL scripts. Setting up the query using a QBE grid and then examining the SQL code automatically produced by the database software is not advisable, this code is unnecessarily complex for the level of answers that candidates would be expected to provide on this paper. If suitable software is not available there are a number of excellent online resources that could be used.

Some candidates continue to answer in pencil and then overwrite their answers in ink which makes some responses very difficult to read when they have been electronically scanned as black and white text. The same applies to the use of fibre tipped pens when the ink soaks through to the other side of the page. Some candidate responses were unreadable.

General comments

It is very important that the question stem is read carefully and the key words highlighted. Some of these key words will indicate the type of answer required, either a single statement or more extended prose, and others will indicate the context in which the question has been set. Identifying and understanding these key words will help candidates to give more appropriate answers to the questions on the examination paper. Several of the questions on this paper, for example, **Question 3**, **Question 5(b)** and **Question 7** required answers or examples in a particular context and generalised responses or responses in a different context were unacceptable.

There is considerable confusion between the terms *data* and *information* and between the terms *field* and *record*. Candidates need to be aware of the difference and make sure that they use the correct terminology when answering questions.

Comments on specific questions

Question 1

The question required candidates to describe two differences between a compiler and an interpreter, which means that in order to be given credit each description must consider both programs. There was considerable confusion regarding the use of the source code in the compilation process, and a general misconception that the source code was not needed for the compiler to execute. Candidates need to improve their understanding of the compilation process and that it is only after compilation that the source code is no longer required in order to run the fully compiled program.

Question 2

It is appreciated that candidates may need space to work out the answer to questions such as **Question 2** and so there is plenty of blank space available in the examination paper. Some candidates put their working in the answer space and so obscured the answer. The number that is to be marked should be clearly visible. Unless there is an indication to the contrary, Examiners will mark the first answer seen.

When answers are given in binary, it is especially important that candidates show clearly whether a particular digit is a zero or a one. If candidates change their answer and overwrite, for example a one with a zero, the result frequently looks like this, , and it is not possible to tell what it is meant to be, so no credit will be given.

(a) The majority of the candidates were able to correctly convert the binary value to the corresponding denary number.

(b) Many candidates correctly converted the given denary value into BCD. The most common incorrect answer was 0101 0010, where candidates had not recognised that it was BCD format that was required and had completed a straightforward denary to binary conversion.

(c) Most candidates realised that they were dealing with a negative number and correctly completed the binary to denary conversion. Candidates need to take care that when they use the method of 'flipping the bits' that they remember to include the minus sign in their answer. A common error was to write 53 instead of –53.

(d) Many candidates correctly converted the denary value to hexadecimal and showed how they had worked out the answer. A common error was to first convert the denary value to binary but then not show how that binary value converted to the hexadecimal answer.

Question 3

Please see the comments in the **General** section of this report regarding context. The question clearly states that a company is evaluating two different options for new software to manage its accounts, and the two options under consideration are given in the question.

(a) A significant number of candidates were able to explain what is meant by open source software. Many candidates were not aware of the requirement for precision in the use of terminology, for example, it is not enough to say '*the customer can change the codes*'; it needs to be explicit that it is the source code of the program that can be changed, and phrases such as '*open source software is open to the public*' are too generic and lack precision to gain any credit.

(b) Candidates found this question more challenging. There was considerable confusion between commercial software and bespoke software, and a general misconception that purchasing the software meant that the company then owned the software, rather than owning a licence to use the software.

(c) As in part **3(b)**, there was considerable confusion between commercial software and bespoke software. The majority of candidates identified benefits to the company of purchasing bespoke software, rather than buying software off-the-shelf, instead of identifying the benefits of choosing commercial software over open source software.

Question 4

(a) This question was answered well. Most candidates correctly incremented the value in the index register.

(b) Most candidates correctly completed the first three instructions and the incrementing of the index register. There were a significant number of completely correct answers to this question. The most frequent causes of error were the omission of the instruction addresses in the first column and the incorrect interpretation of the OUT instruction, outputting 67 instead of 'C'.

This question was one where many candidates had initially answered in pencil and then had overwritten their answers in ink. When scanned electronically this creates a double image which is sometimes very difficult to read. There is plenty of blank space in the question paper and

candidates should be encouraged to do their rough working on these blank pages so that the answers to be marked are clear and can be easily read.

Question 5

Please see the comments in the **General** section of this report regarding context. Part **(b)** clearly states that a School stores a *large* amount of personal data, so candidate responses should reflect this.

(a) This is a clear example of an instance where many candidates had not read the question properly. The question says, '*Draw a line to match each feature with its description*'. There were three feature boxes in the question, so candidates should have drawn three lines, one from each box. Most candidates were able to correctly connect the data dictionary to its description, but there was considerable confusion here between data security and data integrity.

(b) Candidates found this question very challenging. The question asked for factors to be considered when planning a backup procedure, but many candidates answered in terms of securing the data from unauthorised access. Security of the backup is just one of a number of factors that should be considered. Many candidates need to improve their understanding of backing up large quantities of data.

(c) (i) Many candidates were able to draw an Entity-Relationship diagram between the tables. Any representation of an E-R diagram is acceptable, but the most common cause of error was the omission of the degree of the relationships on the diagram. It is essential that this is indicated in some way.

(ii) This question was answered very well. The majority of candidates understood that there was a many-to-many relationship between `CANDIDATE` and `QUALIFICATION` and that in order to implement this `CANDIDATE-QUALIFICATION` would be needed, and that the relationship between `CANDIDATE` and `CANDIDATE-QUALIFICATION` was one-to-many.

(iii) This question was also answered well. The majority of candidates understood how the relationship would be implemented and were able to describe it clearly.

(d) (i) Some candidates found this question very challenging and quite a few did not attempt an answer. Candidates need to improve their understanding of SQL in general and of DML in particular.

(ii) As part **5(d)(i)**, some candidates also found this question challenging and did not attempt to answer it. The best answers were those where straightforward SQL statements had been written, each statement beginning on a new line. Some candidates confused themselves by including numerous brackets which were not necessary and which often actually made the code incorrect. The most common mistake made by candidates who wrote partially correct SQL was the omission of quotation marks around the "`SC12`". On this occasion, minor errors in syntax and spelling were overlooked here and in parts **5(d)(i)** and **5(d)(iii)**. This will not be the case in future examination series. The most frequent errors of this type were to forget the final semi-colon and to put an `S` on the end of table names, for example, `CANDIDATES` instead of `CANDIDATE`.

(iii) As part **5(d)(ii)**, some candidates also found this question challenging and did not attempt to answer it. Many of the remarks for **Question 5(d)(ii)** also apply here. This was a slightly more complex query involving two tables, but is a standard technique and should have caused little difficulty to well prepared candidates. The solution could have been written in a number of ways; any correct method of connecting the two tables was accepted. Again one of the most common mistakes was the omission of the quotation marks around the `Grade`, another frequent error was the reversal of the table name and field name when using dotted notation, so for example, writing `LastName.CANDIDATE` when it should be `CANDIDATE.LastName`, and the final semi-colon was often missing.

Question 6

(a) Most candidates understood that the internet was a network, but there was little appreciation of the scale of the connection. Quite a few candidates confused the World Wide Web (WWW) with a sub-section of the internet.

(b) This question was very well answered with the majority of candidates correctly identifying which connection method best fitted each description.

(c) Overall, there was a good understanding of the differences between on-demand bit streaming and real-time bit streaming in general terms. There is a need for greater understanding of the more technical differences between them.

(d) This question was generally well answered by candidates.

(e) There were some excellent, detailed answers to this question and some candidates clearly understood exactly how a URL and the DNS are used to locate a resource. There was, however, considerable confusion about exactly what the DNS was returning. Candidates need to improve their understanding of how the DNS operates and that it is an IP address that is returned, not the actual resource.

Question 7

Please see the comments in the **General** section of this report regarding context. This question is set in the context of a bank holding personal and financial data about its customers, so candidate answers need to reflect this.

(a) There were many excellent answers to this question with candidates able to describe in detail the difference between security and integrity of data with many good examples, but some examples were not in the context of a bank holding personal data about customers. Candidates should be aware that, at this level of study, it is not enough to say, for example, '*data security ensures that data is secure*'; more explanatory wording is required.

(b) Most candidates were able to identify three security measures for protecting the bank's electronic data. Candidates should be aware that descriptions must be precise and in context. For example, it should be clear that when anti-virus software is used, the virus definitions must be kept up to date in order to detect, quarantine or remove any new viruses.

COMPUTER SCIENCE

Paper 9608/21
Written Paper

Key messages

In preparation for this examination, candidates were expected to have previously studied the pre-release material that had been circulated to Centres. This material provides suggested ways for candidates to practise their problem-solving and programming skills.

There were some excellent answers for the programming questions, but there were a significant number of scripts where programming skills were not very strong. Candidates need extensive practical programming experience prior to sitting this examination.

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.

The syllabus gives very detailed guidance on the key words to use when writing or completing a pseudocode algorithm. Candidates particularly need to appreciate when it is appropriate to use the assignment operator '←' instead of the '=' symbol.

General comments

Candidates and Centres are reminded that written papers are now scanned and marked on computer screens. This means that if a candidate writes the answer to a question on an additional page, they must indicate clearly where their revised answer is to be found.

If answers have been crossed out, the new answers must be written clearly so that the text can be easily read and candidates can be awarded the appropriate mark.

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

The majority of candidates used Visual Basic (console mode), closely followed by Python, with a small minority using Pascal. As in previous sessions, no marks were awarded for programming answers that did not use one of these three languages. It should be noted than Visual Basic (console mode) does not support either the `InputBox()` or `MsgBox()` function.

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

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

Comments on specific questions

Question 1

(a) (i) This was generally well answered with the majority of candidates scoring at least four marks. There was some indication that candidates confused the terms 'selection' and 'iteration'. A number of candidates thought that the third item represented 'selection', perhaps because the `MID` function selected a sub-string from `MyString`.

(ii) A wide range of marks was awarded. Marks were often lost due to inadequate explanations, for example, 'assign a value to the variable' is insufficient. It does not give the same level of information as 'assign `65` to variable `MyScore`'.

Many candidates lost marks due to the use of incorrect technical terminology. For example 'to end the program' is not equivalent to 'to end a While loop'.

Many candidates did not state that item 2 was part of a loop.

A very common mistake was stating that a message would be output in item 6.

The question asked candidates **not** to use mathematical symbols but these still appeared in many answers.

(iii) This was answered well. A small number of candidates missed the '`D`' off the start of the first string.

A small number of candidates gave an answer that included the use of `RIGHT` and `LEFT` functions.

A small number of candidates did not pay enough attention to the case of the letters; 'Ten' in place of 'ten' being the usual example.

Question 2

(a) The majority of candidates correctly identified the data types.

Many candidates gave inadequate explanations, often describing why the data type was appropriate rather than what the identifier would be used for. Candidates should perhaps imagine that they are using the identifier table to help describe the program to another programmer. In this case, the description 'used to store the value from the sensor' would be helpful whereas 'stores a whole number' would is not.

(b) Most candidates gained at least a couple of marks, with many achieving the maximum six. Most were able to produce the correct `IF – THEN – ELSE` structure.

The syllabus was taken as a guide to acceptable pseudocode content. This resulted in marks being lost if candidates did not follow the published style.

Many candidates incorrectly gave an answer either partly or completely in their chosen high-level language.

A significant number of candidates seemed to have an insufficient grasp of conditional statements, often splitting a '>=' comparison into two separate `IF` clauses.

Common errors included:

- The use of '=' instead of the assignment operator '←'.
- Assigning a non-Boolean value to the variable `AlarmState`.
- Omitting the `ENDIF`.

Question 3

(a) This attracted answers at either end of the spectrum.

Common mistakes included:

- The use of '=' instead of the assignment operator '←'.
- Missing or incorrect data types in the parameter list or the inclusion of the array data type.
- Initialising `OutString` to what appeared to be several `SPACE` characters rather than an empty string.
- Use of incorrect brackets for the array index.
- Use of `Len` instead of `Length`.
- A final `OUTPUT` rather than `RETURN`.

(b) (i) This was not answered well, with most candidates not gaining any marks. Two marks were rarely awarded.

Many candidates thought that there needed to be 26 elements in the array. Presumably this mapped to the 26 letters of the alphabet. The question clearly stated that the array contains an entry for each of the `ASCII` characters and that an `ASCII` value is represented by 7 bits. Candidates at this level should have a basic understanding of binary values and realise that 7-bits can represent values from 0 to 127 so leading to an array with 128 elements.

The question stated that the array contained an entry for each of the `ASCII` characters, but few candidates realised that this implied the array data type should be `CHAR`.

(ii) A simple task to implement a flowchart in the chosen high-level language.

A significant number of candidates did not declare their chosen programming language and on occasion it was not even possible to guess which one was being used.

Many candidates gained the marks for the input and output statements but fewer candidates gained the marks for assigning a value to the correct array element or for a correct loop.

Several candidates introduced an additional loop counter and often these solutions did not increment the variable used as the array index.

Question 4

(a) Many candidates gained one mark. The use of absolute rather than relative terms was common. For example, saying that debugging would be 'easy' rather than 'easier'. This was not acceptable. Popular correct answers related to testing/debugging or the dividing of work between different people.

(b) (i) Many candidates either gave no answer to this part, or added symbols other than those required. Although the use of these symbols was referred to in the pre-release, it is suspected that many Centres did not cover this topic. Those candidates that used the correct symbols tended to get full marks.

(ii) This was not well answered. Candidates did not appreciate that a 'Card Payment' module would simply need the details of the card and the amount to be paid, and would return a payment confirmation. 'Card details' was the most common correct answer.

Question 5

(a) (i) There were many incorrect answers giving reasons that would have applied equally well to a solution that held the data in an array or similar structure within the program. It appeared that many candidates do not know why a file is used. Perhaps this is in some way due to the rise of 'always on' computers such as tablets but the concept of a file is fundamental.

Common incorrect answers included:

- 'to know where the data is'
- 'to allow easy access to the data'
- 'to store all the data in one place'

(ii) Many candidates correctly described the problem of 'searching for a data item when you do not know where one begins and another ends'. Terms that were deemed as vague, such as 'muddled up' were not accepted.

A common mistake made when describing a possible solution was the addition of a space character between the data items, not appreciating that these already contained spaces. Some candidates first converted each data item into CamelCase (or similar) making the use of a space separator a workable solution. Only a small number of candidates explicitly referred to the selection of a separator that did not occur in the original strings.

(b) There were some excellent answers to this, with a small number of candidates gaining maximum marks, but there were a large number of candidates who could demonstrate only a low level of programming ability. Many candidates did not even include a loop in their answer.

With reference to the different mark points:

- Correct procedure heading and ending was uncommon, especially in VB solutions.
- The mark for the declaration of the three variables was often given.
- The open file statement often contained an incorrect filename or an invalid file method.
- The loop structure was often incorrect. A common mistake was not checking for "`##`" until after it had been written to the file. A significant number of candidates offered a solution based on an unconditional loop with an internal `BREAK` statement, which is not to be encouraged.
- The mark for the input of the three variables inside the loop was often given, as was the mark for concatenation, including the use of a separator.
- The syntax of file writing and closing statements was often incorrect.

Question 6

(a) This attracted extreme responses with some very good answers. The majority of candidates gained full marks but some did not know what to do. It is difficult to identify common mistakes, except to say that if marks were lost the tended to be towards the right-hand columns of the table.

(b) (i) Many candidates correctly described searching for `String2` within `String1`, but fewer candidates mentioned the function returning the index value. Many candidates referred incorrectly to the index value being output. Unclear descriptions were common, such as 'to see if the letters in `String 2` are found in `String1`' which does not require an exact substring match.

(ii) Correctly answered by only a few candidates. Several candidates gained the first mark but did not give a meaningful reason, suggesting that in some cases the first answer might have just been a lucky guess.
Many candidates gave the return value as '`f`' which was the identifier from the pseudocode.

(iii) Only a very few candidates demonstrated that they had understood the problem. Many showed some knowledge of run-time errors by referring to endless loops or less precisely to 'program crashes' but few were able to describe the problem with reference to the given pseudocode. Successful candidates referred to either the potential 'subscript out of range' problem or the endless loop when the end of `String2` is reached.

COMPUTER SCIENCE

Paper 9608/22
Written Paper

Key messages

In preparation for this examination, candidates were expected to have previously studied the pre-release material that had been circulated to Centres. This material provides suggested ways for candidates to practise their problem-solving and programming skills.

There were some excellent answers for the programming questions, but there were a significant number of scripts where programming skills were not very strong. Candidates need extensive practical programming experience prior to sitting this examination.

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.

The syllabus gives very detailed guidance on the key words to use when writing or completing a pseudocode algorithm. Candidates particularly need to appreciate when it is appropriate to use the assignment operator '←' instead of the '=' symbol.

General comments

Candidates and Centres are reminded that written papers are now scanned and marked on computer screens. This means that if a candidate writes the answer to a question on an additional page, they must indicate clearly where their revised answer is to be found.

If answers have been crossed out, the new answers must be written clearly so that the text can be easily read and candidates can be awarded the appropriate mark.

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

The majority of candidates used Visual Basic (console mode), closely followed by Python, with a small minority using Pascal. As in previous sessions, no marks were awarded for programming answers that did not use one of these three languages. It should be noted than Visual Basic (console mode) does not support either the `InputBox()` or `MsgBox()` function.

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

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

Comments on specific questions

Question 1

(a) (i) This was generally well answered with the majority of candidates scoring at least four marks. There was some indication that candidates confused the terms 'selection' and 'iteration'. A number of candidates thought that the third item represented 'selection', perhaps because the `MID` function selected a sub-string from `MyString`.

(ii) A wide range of marks was awarded. Marks were often lost due to inadequate explanations, for example, 'assign a value to the variable' is insufficient. It does not give the same level of information as 'assign `65` to variable `MyScore`'.

Many candidates lost marks due to the use of incorrect technical terminology. For example 'to end the program' is not equivalent to 'to end a While loop'.

Many candidates did not state that item 2 was part of a loop.

A very common mistake was stating that a message would be output in item 6.

The question asked candidates **not** to use mathematical symbols but these still appeared in many answers.

(iii) This was answered well. A small number of candidates missed the '`D`' off the start of the first string.

A small number of candidates gave an answer that included the use of `RIGHT` and `LEFT` functions.

A small number of candidates did not pay enough attention to the case of the letters; 'Ten' in place of 'ten' being the usual example.

Question 2

(a) The majority of candidates correctly identified the data types.

Many candidates gave inadequate explanations, often describing why the data type was appropriate rather than what the identifier would be used for. Candidates should perhaps imagine that they are using the identifier table to help describe the program to another programmer. In this case, the description 'used to store the value from the sensor' would be helpful whereas 'stores a whole number' would is not.

(b) Most candidates gained at least a couple of marks, with many achieving the maximum six. Most were able to produce the correct `IF – THEN – ELSE` structure.

The syllabus was taken as a guide to acceptable pseudocode content. This resulted in marks being lost if candidates did not follow the published style.

Many candidates incorrectly gave an answer either partly or completely in their chosen high-level language.

A significant number of candidates seemed to have an insufficient grasp of conditional statements, often splitting a '>=' comparison into two separate `IF` clauses.

Common errors included:

- The use of '=' instead of the assignment operator '←'.
- Assigning a non-Boolean value to the variable `AlarmState`.
- Omitting the `ENDIF`.

Question 3

(a) This attracted answers at either end of the spectrum.

Common mistakes included:

- The use of '=' instead of the assignment operator '←'.
- Missing or incorrect data types in the parameter list or the inclusion of the array data type.
- Initialising `OutString` to what appeared to be several `SPACE` characters rather than an empty string.
- Use of incorrect brackets for the array index.
- Use of `Len` instead of `Length`.
- A final `OUTPUT` rather than `RETURN`.

(b) (i) This was not answered well, with most candidates not gaining any marks. Two marks were rarely awarded.

Many candidates thought that there needed to be 26 elements in the array. Presumably this mapped to the 26 letters of the alphabet. The question clearly stated that the array contains an entry for each of the `ASCII` characters and that an `ASCII` value is represented by 7 bits. Candidates at this level should have a basic understanding of binary values and realise that 7-bits can represent values from 0 to 127 so leading to an array with 128 elements.

The question stated that the array contained an entry for each of the `ASCII` characters, but few candidates realised that this implied the array data type should be `CHAR`.

(ii) A simple task to implement a flowchart in the chosen high-level language.

A significant number of candidates did not declare their chosen programming language and on occasion it was not even possible to guess which one was being used.

Many candidates gained the marks for the input and output statements but fewer candidates gained the marks for assigning a value to the correct array element or for a correct loop.

Several candidates introduced an additional loop counter and often these solutions did not increment the variable used as the array index.

Question 4

(a) Many candidates gained one mark. The use of absolute rather than relative terms was common. For example, saying that debugging would be 'easy' rather than 'easier'. This was not acceptable. Popular correct answers related to testing/debugging or the dividing of work between different people.

(b) (i) Many candidates either gave no answer to this part, or added symbols other than those required. Although the use of these symbols was referred to in the pre-release, it is suspected that many Centres did not cover this topic. Those candidates that used the correct symbols tended to get full marks.

(ii) This was not well answered. Candidates did not appreciate that a 'Card Payment' module would simply need the details of the card and the amount to be paid, and would return a payment confirmation. 'Card details' was the most common correct answer.

Question 5

(a) (i) There were many incorrect answers giving reasons that would have applied equally well to a solution that held the data in an array or similar structure within the program. It appeared that many candidates do not know why a file is used. Perhaps this is in some way due to the rise of 'always on' computers such as tablets but the concept of a file is fundamental.

Common incorrect answers included:

- 'to know where the data is'
- 'to allow easy access to the data'
- 'to store all the data in one place'

(ii) Many candidates correctly described the problem of 'searching for a data item when you do not know where one begins and another ends'. Terms that were deemed as vague, such as 'muddled up' were not accepted.

A common mistake made when describing a possible solution was the addition of a space character between the data items, not appreciating that these already contained spaces. Some candidates first converted each data item into CamelCase (or similar) making the use of a space separator a workable solution. Only a small number of candidates explicitly referred to the selection of a separator that did not occur in the original strings.

(b) There were some excellent answers to this, with a small number of candidates gaining maximum marks, but there were a large number of candidates who could demonstrate only a low level of programming ability. Many candidates did not even include a loop in their answer.

With reference to the different mark points:

- Correct procedure heading and ending was uncommon, especially in VB solutions.
- The mark for the declaration of the three variables was often given.
- The open file statement often contained an incorrect filename or an invalid file method.
- The loop structure was often incorrect. A common mistake was not checking for "`##`" until after it had been written to the file. A significant number of candidates offered a solution based on an unconditional loop with an internal `BREAK` statement, which is not to be encouraged.
- The mark for the input of the three variables inside the loop was often given, as was the mark for concatenation, including the use of a separator.
- The syntax of file writing and closing statements was often incorrect.

Question 6

(a) This attracted extreme responses with some very good answers. The majority of candidates gained full marks but some did not know what to do. It is difficult to identify common mistakes, except to say that if marks were lost the tended to be towards the right-hand columns of the table.

(b) (i) Many candidates correctly described searching for `String2` within `String1`, but fewer candidates mentioned the function returning the index value. Many candidates referred incorrectly to the index value being output. Unclear descriptions were common, such as 'to see if the letters in `String 2` are found in `String1`' which does not require an exact substring match.

(ii) Correctly answered by only a few candidates. Several candidates gained the first mark but did not give a meaningful reason, suggesting that in some cases the first answer might have just been a lucky guess.
Many candidates gave the return value as '`f`' which was the identifier from the pseudocode.

(iii) Only a very few candidates demonstrated that they had understood the problem. Many showed some knowledge of run-time errors by referring to endless loops or less precisely to 'program crashes' but few were able to describe the problem with reference to the given pseudocode. Successful candidates referred to either the potential 'subscript out of range' problem or the endless loop when the end of `String2` is reached.

COMPUTER SCIENCE

Paper 9608/23
Written Paper

Key messages

In preparation for this examination, candidates were expected to have previously studied the pre-release material that had been circulated to Centres. This material provides suggested ways for candidates to practise their problem-solving and programming skills.

There were some excellent answers for the programming questions, but there were a significant number of scripts where programming skills were not very strong. Candidates need extensive practical programming experience prior to sitting this examination.

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.

The syllabus gives very detailed guidance on the key words to use when writing or completing a pseudocode algorithm. Candidates particularly need to appreciate when it is appropriate to use the assignment operator '←' instead of the '=' symbol.

General comments

Candidates and Centres are reminded that written papers are now scanned and marked on computer screens. This means that if a candidate writes the answer to a question on an additional page, they must indicate clearly where their revised answer is to be found.

If answers have been crossed out, the new answers must be written clearly so that the text can be easily read and candidates can be awarded the appropriate mark.

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

The majority of candidates used Visual Basic (console mode), closely followed by Python, with a small minority using Pascal. As in previous sessions, no marks were awarded for programming answers that did not use one of these three languages. It should be noted than Visual Basic (console mode) does not support either the `InputBox()` or `MsgBox()` function.

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

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

Comments on specific questions

Question 1

(a) (i) Many candidates scored full marks, but a large number of candidates scored less than two marks, indicating that they had low knowledge of basic programming constructs. There was some indication that candidates confused the terms 'selection' and 'iteration'.

(ii) A wide range of marks was awarded. Marks were often lost due to inadequate explanations, for example, 'assign a value to the variable' is insufficient. It does not give the same level of information as 'assign string "`Gordon`" to variable `MyName`'.

Many candidates lost marks due to the use of incorrect technical terminology. For example 'to end the program' is not equivalent to 'to end an `IF` statement'. Many candidates did not state that item 1 was part of a loop. A very common mistake was stating that item 3 changed the data type of the variable.

The question asked candidates **not** to use mathematical symbols but these still appeared in many answers.

(iii) This was generally answered well. Mistakes made by a small number of candidates included:

- Missing the '`P`' off the start of the first string.
- Giving an answer that included the use of `RIGHT` and `LEFT` functions.
- Not paying enough attention to the case of the letters; '`main`' in place of '`Main`' being the usual example.

Question 2

(a) Most candidates gained at least four marks. No real pattern of incorrect answers was observed, although `STRING` was used for various numeric values.

A small number of candidates gave '`CONSTANT`' as a data type for identifier, `NormalTemp`.

(b) Most candidates gained at least a couple of marks, with many achieving the maximum six. Most candidates were able to produce the correct `IF – THEN – ELSE` structure, many decided to implement three nested `IF` statements to check for the stopping condition via three separate single comparisons rather than one making use of the Boolean operator `AND` to check the stopping condition in a single statement.

The syllabus was taken as a guide as to acceptable pseudocode content. This resulted in marks being lost if candidates did not follow the published style.

Many candidates incorrectly gave an answer either partly or completely in their chosen high-level language.

A significant number of candidates seemed to have an inadequate grasp of conditional statements, often splitting a '>=' comparison into two separate `IF` clauses.

Common mistakes included:

- The use of '=' instead of the assignment operator '← '.
- The use of '←' as a comparison operator.
- Use of comma in place of AND to link comparison statements.
- Incorrectly comparing `EngineTemp` with a literal value.
- Assigning a non-Boolean value to the variable `EngineStop`.
- Omitting the `ENDIF`.

Question 3

(a) This attracted answers at either end of the spectrum; either very good or very poor.

Common mistakes included:

- The use of '=' instead of the assignment operator '←'.
- Missing or incorrect data types in the parameter list or the addition of the array data type.
- Use of incorrect brackets for the array index.
- Use of `ASC` rather than `CHR`.
- An incorrect comment for the line `Index ← 1` (e.g. 'Increment the index').
- An inadequate comment for the line `Index ← 1` (e.g. 'set index to 1', which is hardly useful).
- A final `RETURN` of something other than `OriginalChar`.

(b) This was a simple task to implement a flowchart in the chosen high-level language.

A significant number of candidates did not declare their chosen programming language and on occasion it was not even possible to guess which one they were using.

Many candidates gained the marks for the input and output statements but few candidates gained the marks for assigning values to `OriginalChar` or `CipherChar`, or for a correct loop.

Several candidates introduced an additional loop counter and often these solutions did not increment the variable used as the array index.

Question 4

(a) The small number of correct answers gained marks for mention of functions or procedures and the use of global variables.

It appeared that many candidates did not understand the question and answers referring to unrelated topics were common. References to stepwise refinement, IDE features and descriptions of fundamental programming concepts (i.e. non-modular) were seen frequently. Many candidates simply gave the name of two programming languages.

(b) (i) Many candidates either gave no answer to this part, or added symbols other than the one required. Although the use of these symbols was referred to in the pre-release, it appears that many candidates had covered this topic. Those candidates that used the required symbol tended to use it correctly.

(ii) Many candidates gained full marks, demonstrating an understanding of the use of parameters between modules.

Question 5

(a) (i) A number of candidates correctly described the benefit of fixed-length strings in terms of making it easier to search for a specific entry. Many more seemed to miss the point, referring to:

- making sure the data entered was valid
- making it easy to estimate the file size
- making the data easier to write.

Descriptions of a drawback were better, with candidates often quoting the unnecessary space taken up or that the original data items may need to be truncated.

(ii) This attracted extreme responses. The majority of answers were simply incorrect and often lacked any meaning, but many lost marks due to answers not being very clear. This was a question where correct terminology was important. For example, it is incorrect to say that 'all the old data will be output' when what is meant is 'all the old data will be overwritten'.

A small number of candidates gained full marks for a perfect description of the problem, effect and solution. Many candidates gained two marks only as their explanation of the problem and the effect

were blurred. A small number of candidates said that the file had to be read before it could be written. Several candidates attempted to identify non-existent syntax errors in the pseudocode.

(b) There were some excellent answers to this, with a small number of candidates gaining maximum marks. There were also a significant number of candidates who demonstrated only a low level of programming ability. Many candidates did not even include a loop in their answer, and a significant number of candidates offered no answer.

With reference to the different mark points:

- Correct procedure heading and ending was uncommon, especially in VB solutions.
- The mark for the declaration of the counter variable was often given.
- The prompt and input were usually correct; this was the most common mark.
- The open file statement often contained an incorrect filename or an invalid file method
- The loop structure was often not correct. Often the check for `EOF` was not done properly. A significant number of candidates offered a solution based on an unconditional loop with an internal `BREAK` statement, which is not to be encouraged.
- The mark for reading a line from the file was not commonly given. Often the solution would perform three separate read statements and assign one line to each variable.
- Mark point 7, 8 and 9 usually went together and were rarely given.
- The mark for incrementing a counter was commonly given.
- The mark for the final output was commonly given.
- The syntax of file close statement was often incorrect.

Question 6

(a) (i) This attracted extreme responses. The majority of candidates gained full marks, but some candidates did not know what to do. Some candidates paid insufficient attention to the case of individual characters.

(ii) The majority of candidates correctly described 'capitalising the first character of each word and converting all the others to lower case' for full marks. A number did not mention the conversion of subsequent characters to lower case and there were many vague descriptions of 'correcting the string'.

(b) (i) Only a small number of candidates correctly stated that the function would return an empty string. Very many candidates referred to code 'crashing' or 'endlessly looping' suggesting that this is the expected response based on past questions of this type.

(ii) The majority of candidates were able to define appropriate test strings, but few candidates were able to explain why they had been chosen. The most successful candidates described the following types of test strings:

- correct format (e.g. 'Big Ben')
- 'Opposite' format (e.g. 'bIG bEN')
- all capitals
- all lower case
- with spaces
- without spaces

A significant number of candidates suggested strings with non-alpha characters, such as numbers or symbols, giving as an explanation that these would be rejected or would cause an error. There is no suggestion in the pseudocode that this should happen.

A small number of candidates completely misinterpreted the question. Among these answers were:

- definitions of different types of test data (normal, boundary, abnormal)
- different stages of testing (alpha, beta)
- explanation of lines from the pseudocode

COMPUTER SCIENCE

Paper 9608/31
Written Paper

Key messages

Candidates must write their answer in the spaces provided. Candidates should specify where the answer has been written if they are not written in the spaces provided.

General comments

In general, the performance of candidates showed knowledge of all of the topics examined on the paper. In many cases, candidates did not apply this knowledge in the context given by the question. There was evidence of this in a number of questions on the paper. Many questions are designed in order for candidates to apply their knowledge, rather than just require them to state their knowledge.

Comments on specific questions

Question 1

There were a large minority of candidates who did not know the difference between a bus network and a star network. Many candidates believed that the server controlled the network. This was often demonstrated in descriptions of how collisions were handled and also how the network functioned when a switch was added.

(a) In general, there were three types of answer: the first type of answer showed a bus network with some indication of terminators at either end, the second type of answer showed the bus network with no indication of terminators, and the third type of an answer showed a star network. The second answer was the answer seen most often.

(b) Full marks to this question were rare. Most candidates scored some marks. Those candidates who gave a star network in **part (a)**, tended to given incorrect answers to the second and third statements.

(c) (i) The majority of candidates could identify a collision as the problem.

(ii) Good answers described how the two computers stop transmitting before waiting a random amount of time before trying to transmit again. These answers were in a minority. Many candidates believed, incorrectly, that the server was in charge of all transfers on the network. Consequently, in the scenario described, the server would have priority and would therefore transmit first. There were also many candidates who described the use of IP addresses and packets. This type of answer scored no marks.

(d) A good answer would have described the nature of a switch and how the direct connections to it result in the bus network becoming a star network. Although many answers gave the change in topology few went on to give a clear description of how this physical change would eliminate collisions. Many incorrectly wrote about the role of the server and how the switch used IP addresses for routing.

Question 2

Many answers given for data items present in a digital certificate were too vague. Despite much evidence that candidates had knowledge of the use of encryption they often did not make suitable choices for keys or could give worthwhile reasons for their choice. Candidates had difficulties to understand the difference between security and authentication when using encryption.

(a) Many answers were too vague. "Public key" and "digital signature" were often given as answers. It was necessary to state who owned the public key and who owned the digital signature. Terms such as "user" and "sender" were also treated as being too vague. There is only one owner of a digital certificate but there will be many users. Although the term "user" is used in **part (b)** it is clear that the user in that case is the person making the application for the digital certificate. Many answers incorrectly gave a private key as one of the data items on this public document.

(b) (i) Candidates were expected to give answers that fitted the context given in the question. For this part the key to be chosen is one that is going to be sent to the CA and eventually published. An answer worthy of full marks would have chosen "public" and justified this by stating either that a private key is always kept private or that a public key is one that can be known by others. Only a minority of candidates gave a correct justification.

(ii) The question clearly states that the key being used belongs to the Certificate Authority. An answer that argued that the user would not know the private key of the Certificate Authority or would only know the public key of the Certificate Authority would have achieved full marks.

(iii) In this part, the context justifies the use of the Certificate Authority's private key. Many candidates gave the correct key but were not able to give an adequate justification.

(c) (i) A majority of candidates gave the correct answer. "Cipher text" was a common but incorrect answer.

(ii) Many mentioned the use of Alexa's public key but often did not make it clear that the key was retrieved from Alexa's digital certificate. Many candidates also described a hashing process but were not able to explain how this generated hash was used to authenticate the received message.

(iii) This was not answered well. Many answers lacked detail and focused more on security rather than the required authentication.

Question 3

This question was not answered well. There was little evidence that candidates understood virtual machines and how they can be used for testing software.

(a) (i) This question was not answered well. Any marks awarded were usually for hardware emulation.

(ii) Many candidates gained two marks for correctly explaining a feature of the guest operating system and a different feature of the host operating system. Many answers referred to the host operating system as the "original OS" or the "main OS". These answers gained no credit.

(b) (i) This question was not answered well. Candidates did not make full use of the given scenario and focus their answers on the use of virtual machines to test possible new operating systems and possible new webserver software. Incorrect responses included: using a virtual machine to run the old operating system, running the virtual machines on the current web server system, and assuming that virtual machines equated to virtual memory.

(ii) Many answers showed candidates were aware that there were issues with regard to the performance of a system and the same system running in a virtual machine. Many answers were often not expressed very clearly.

Question 4

In **part (a)** few candidates demonstrated a full understanding of the relationships between file organisations and file access methods. In **part (b)** candidates could often give the correct file organisation but were not able to give an adequate justification.

(a) Very few candidates gave all four correct responses. The majority of candidates scored two or three marks.

(b) (i) As either sequential or serial were correct answers, the vast majority of candidates scored at least one mark. Many candidates found it difficult to provide an appropriate justification for their selection. A few candidates stated clearly that as all customers would be given a statement, every record in the file would be processed and as a result, order was not an issue. For those who selected sequential, many incorrectly wrote that the order would be alphabetical, based on the name of the customer.

(ii) Many candidates correctly identified random as the correct organisation but often did not give a suitable justification. Correct responses for the justification tended to refer to the faster access to the data that random organisation allows.

(iii) Candidate answers to this part tended to be better than for the other two parts. Appending a new record and the resulting chronological order of the records were often given as answers.

Question 5

The majority of candidates demonstrated that they could construct truth tables, but only a few with total accuracy. Most candidates showed some knowledge of flip-flops.

(a) The majority of candidates provided a correct truth table for the NAND gate.

(b) (i) Fully correct tables were rare as were totally incorrect tables. Candidates were expected to work systematically, using the truth table for the NAND gate produced in **part (a)**, to complete the truth table for the given logic circuit. There was some indication that, because the logic circuit was that for a SR flip-flop, some candidates thought the values for Q and $\overline{Q}$ should be stated rather than generated. This applied in particular to the final row entry where values were sometimes given as "undefined" or "--".

(ii) Candidates often showed awareness that a particular combination of S and R should not occur, but often did not explain adequately why this should be the case. Marks were given for stating that Q and $\overline{Q}$ should not have the same value, as they are complements of each other. Candidates who wrote about an "invalid state" were given no credit as the question already states that one combination "should not be allowed to occur".

(c) (i) Many candidates gave the correct answer of a clock pulse. The word "clock" was required as any possible logic circuit abbreviations were considered too vague.

(ii) Only a minority of candidates could give an advantage of the JK flip-flop.

(d) Many answers mentioned some type of memory or referred to data storage. Fewer answer scored the second mark, due to not stating that a flip-flop can be used to store a single bit.

Question 6

The overwhelming majority of candidates were aware that the system described performed a monitoring function but many were incorrect in assuming that the system also exhibited control functions. There was good awareness of the different types of sensor but the reasons for inclusion of the sensor were often inadequate. The majority of candidates found it difficult to dry run fully the assembly language code and also to state the changes required to the code.

(a) (i) Many candidates correctly identified the system described as a monitoring system.

(ii) In giving their justification, it was often evident that many candidates believed, incorrectly, that the loud alarm was evidence of the system exhibiting control. However, it was the lack of any element of control that made the system a monitoring system.

(b) Temperature, sound and light sensors were not given credit for this particular system. A mark for the reason needed to be more specific than statements such as “a motion sensor detects motion”.

(c) (i) There were some excellent attempts at this question gaining full marks. Many answers displayed correct entries for COUNT and VALUE, and the first two entries for ACC. The mixture of binary and denary integers, which was meant to be helpful to the candidates, did not seem to cause any problems. The number of rows in the dry run table is an indication of the maximum space required for a full answer. Candidates should only make a further entry into a column when the value changes.

(ii) The operand ‘#1’ was only given by a few candidates. Some candidates gave the full instruction rather than just the operand; these answers did not get any credit.

(iii) Many candidates identified the required instruction. Only the most able candidates were able to give the correct operand for the amended instruction.

COMPUTER SCIENCE

Paper 9608/32
Written Paper

Key messages

Candidates must write their answer in the spaces provided. Candidates should specify where the answer has been written if they are not written in the spaces provided.

General comments

In general, the performance of candidates showed knowledge of all of the topics examined on the paper. In many cases, candidates did not apply this knowledge in the context given by the question. There was evidence of this in a number of questions on the paper. Many questions are designed in order for candidates to apply their knowledge, rather than just require them to state their knowledge.

Comments on specific questions

Question 1

There were a large minority of candidates who did not know the difference between a bus network and a star network. Many candidates believed that the server controlled the network. This was often demonstrated in descriptions of how collisions were handled and also how the network functioned when a switch was added.

(a) In general, there were three types of answer: the first type of answer showed a bus network with some indication of terminators at either end, the second type of answer showed the bus network with no indication of terminators, and the third type of an answer showed a star network. The second answer was the answer seen most often.

(b) Full marks to this question were rare. Most candidates scored some marks. Those candidates who gave a star network in **part (a)**, tended to given incorrect answers to the second and third statements.

(c) (i) The majority of candidates could identify a collision as the problem.

(ii) Good answers described how the two computers stop transmitting before waiting a random amount of time before trying to transmit again. These answers were in a minority. Many candidates believed, incorrectly, that the server was in charge of all transfers on the network. Consequently, in the scenario described, the server would have priority and would therefore transmit first. There were also many candidates who described the use of IP addresses and packets. This type of answer scored no marks.

(d) A good answer would have described the nature of a switch and how the direct connections to it result in the bus network becoming a star network. Although many answers gave the change in topology few went on to give a clear description of how this physical change would eliminate collisions. Many incorrectly wrote about the role of the server and how the switch used IP addresses for routing.

Question 2

Many answers given for data items present in a digital certificate were too vague. Despite much evidence that candidates had knowledge of the use of encryption they often did not make suitable choices for keys or could give worthwhile reasons for their choice. Candidates had difficulties to understand the difference between security and authentication when using encryption.

(a) Many answers were too vague. "Public key" and "digital signature" were often given as answers. It was necessary to state who owned the public key and who owned the digital signature. Terms such as "user" and "sender" were also treated as being too vague. There is only one owner of a digital certificate but there will be many users. Although the term "user" is used in **part (b)** it is clear that the user in that case is the person making the application for the digital certificate. Many answers incorrectly gave a private key as one of the data items on this public document.

(b) (i) Candidates were expected to give answers that fitted the context given in the question. For this part the key to be chosen is one that is going to be sent to the CA and eventually published. An answer worthy of full marks would have chosen "public" and justified this by stating either that a private key is always kept private or that a public key is one that can be known by others. Only a minority of candidates gave a correct justification.

(ii) The question clearly states that the key being used belongs to the Certificate Authority. An answer that argued that the user would not know the private key of the Certificate Authority or would only know the public key of the Certificate Authority would have achieved full marks.

(iii) In this part, the context justifies the use of the Certificate Authority's private key. Many candidates gave the correct key but were not able to give an adequate justification.

(c) (i) A majority of candidates gave the correct answer. "Cipher text" was a common but incorrect answer.

(ii) Many mentioned the use of Alexa's public key but often did not make it clear that the key was retrieved from Alexa's digital certificate. Many candidates also described a hashing process but were not able to explain how this generated hash was used to authenticate the received message.

(iii) This was not answered well. Many answers lacked detail and focused more on security rather than the required authentication.

Question 3

This question was not answered well. There was little evidence that candidates understood virtual machines and how they can be used for testing software.

(a) (i) This question was not answered well. Any marks awarded were usually for hardware emulation.

(ii) Many candidates gained two marks for correctly explaining a feature of the guest operating system and a different feature of the host operating system. Many answers referred to the host operating system as the "original OS" or the "main OS". These answers gained no credit.

(b) (i) This question was not answered well. Candidates did not make full use of the given scenario and focus their answers on the use of virtual machines to test possible new operating systems and possible new webserver software. Incorrect responses included: using a virtual machine to run the old operating system, running the virtual machines on the current web server system, and assuming that virtual machines equated to virtual memory.

(ii) Many answers showed candidates were aware that there were issues with regard to the performance of a system and the same system running in a virtual machine. Many answers were often not expressed very clearly.

Question 4

In **part (a)** few candidates demonstrated a full understanding of the relationships between file organisations and file access methods. In **part (b)** candidates could often give the correct file organisation but were not able to give an adequate justification.

(a) Very few candidates gave all four correct responses. The majority of candidates scored two or three marks.

(b) (i) As either sequential or serial were correct answers, the vast majority of candidates scored at least one mark. Many candidates found it difficult to provide an appropriate justification for their selection. A few candidates stated clearly that as all customers would be given a statement, every record in the file would be processed and as a result, order was not an issue. For those who selected sequential, many incorrectly wrote that the order would be alphabetical, based on the name of the customer.

(ii) Many candidates correctly identified random as the correct organisation but often did not give a suitable justification. Correct responses for the justification tended to refer to the faster access to the data that random organisation allows.

(iii) Candidate answers to this part tended to be better than for the other two parts. Appending a new record and the resulting chronological order of the records were often given as answers.

Question 5

The majority of candidates demonstrated that they could construct truth tables, but only a few with total accuracy. Most candidates showed some knowledge of flip-flops.

(a) The majority of candidates provided a correct truth table for the NAND gate.

(b) (i) Fully correct tables were rare as were totally incorrect tables. Candidates were expected to work systematically, using the truth table for the NAND gate produced in **part (a)**, to complete the truth table for the given logic circuit. There was some indication that, because the logic circuit was that for a SR flip-flop, some candidates thought the values for Q and $\overline{Q}$ should be stated rather than generated. This applied in particular to the final row entry where values were sometimes given as "undefined" or "--".

(ii) Candidates often showed awareness that a particular combination of S and R should not occur, but often did not explain adequately why this should be the case. Marks were given for stating that Q and $\overline{Q}$ should not have the same value, as they are complements of each other. Candidates who wrote about an "invalid state" were given no credit as the question already states that one combination "should not be allowed to occur".

(c) (i) Many candidates gave the correct answer of a clock pulse. The word "clock" was required as any possible logic circuit abbreviations were considered too vague.

(ii) Only a minority of candidates could give an advantage of the JK flip-flop.

(d) Many answers mentioned some type of memory or referred to data storage. Fewer answer scored the second mark, due to not stating that a flip-flop can be used to store a single bit.

Question 6

The overwhelming majority of candidates were aware that the system described performed a monitoring function but many were incorrect in assuming that the system also exhibited control functions. There was good awareness of the different types of sensor but the reasons for inclusion of the sensor were often inadequate. The majority of candidates found it difficult to dry run fully the assembly language code and also to state the changes required to the code.

(a) (i) Many candidates correctly identified the system described as a monitoring system.

(ii) In giving their justification, it was often evident that many candidates believed, incorrectly, that the loud alarm was evidence of the system exhibiting control. However, it was the lack of any element of control that made the system a monitoring system.

(b) Temperature, sound and light sensors were not given credit for this particular system. A mark for the reason needed to be more specific than statements such as “a motion sensor detects motion”.

(c) (i) There were some excellent attempts at this question gaining full marks. Many answers displayed correct entries for COUNT and VALUE, and the first two entries for ACC. The mixture of binary and denary integers, which was meant to be helpful to the candidates, did not seem to cause any problems. The number of rows in the dry run table is an indication of the maximum space required for a full answer. Candidates should only make a further entry into a column when the value changes.

(ii) The operand ‘#1’ was only given by a few candidates. Some candidates gave the full instruction rather than just the operand; these answers did not get any credit.

(iii) Many candidates identified the required instruction. Only the most able candidates were able to give the correct operand for the amended instruction.

COMPUTER SCIENCE

Paper 9608/33
Written Paper

Key messages

Candidates must write their answer in the spaces provided. Candidates should specify where the answer has been written if they are not written in the spaces provided.

General comments

In general, the performance of candidates showed knowledge of all of the topics examined on the paper. In many cases, candidates did not apply this knowledge in the context given by the question. There was evidence of this in a number of questions on the paper. Many questions are designed in order for candidates to apply their knowledge, rather than just require them to state their knowledge.

Comments on specific questions

Question 1

There were a large minority of candidates who did not know the difference between a bus network and a star network. Many candidates believed that the server controlled the network. This was often demonstrated in descriptions of how collisions were handled and also how the network functioned when a switch was added.

(a) In general, there were three types of answer: the first type of answer showed a bus network with some indication of terminators at either end, the second type of answer showed the bus network with no indication of terminators, and the third type of an answer showed a star network. The second answer was the answer seen most often.

(b) Full marks to this question were rare. Most candidates scored some marks. Those candidates who gave a star network in **part (a)**, tended to given incorrect answers to the second and third statements.

(c) (i) The majority of candidates could identify a collision as the problem.

(ii) Good answers described how the two computers stop transmitting before waiting a random amount of time before trying to transmit again. These answers were in a minority. Many candidates believed, incorrectly, that the server was in charge of all transfers on the network. Consequently, in the scenario described, the server would have priority and would therefore transmit first. There were also many candidates who described the use of IP addresses and packets. This type of answer scored no marks.

(d) A good answer would have described the nature of a switch and how the direct connections to it result in the bus network becoming a star network. Although many answers gave the change in topology few went on to give a clear description of how this physical change would eliminate collisions. Many incorrectly wrote about the role of the server and how the switch used IP addresses for routing.

Question 2

Many answers given for data items present in a digital certificate were too vague. Despite much evidence that candidates had knowledge of the use of encryption they often did not make suitable choices for keys or could give worthwhile reasons for their choice. Candidates had difficulties to understand the difference between security and authentication when using encryption.

(a) Many answers were too vague. "Public key" and "digital signature" were often given as answers. It was necessary to state who owned the public key and who owned the digital signature. Terms such as "user" and "sender" were also treated as being too vague. There is only one owner of a digital certificate but there will be many users. Although the term "user" is used in **part (b)** it is clear that the user in that case is the person making the application for the digital certificate. Many answers incorrectly gave a private key as one of the data items on this public document.

(b) (i) Candidates were expected to give answers that fitted the context given in the question. For this part the key to be chosen is one that is going to be sent to the CA and eventually published. An answer worthy of full marks would have chosen "public" and justified this by stating either that a private key is always kept private or that a public key is one that can be known by others. Only a minority of candidates gave a correct justification.

(ii) The question clearly states that the key being used belongs to the Certificate Authority. An answer that argued that the user would not know the private key of the Certificate Authority or would only know the public key of the Certificate Authority would have achieved full marks.

(iii) In this part, the context justifies the use of the Certificate Authority's private key. Many candidates gave the correct key but were not able to give an adequate justification.

(c) (i) A majority of candidates gave the correct answer. "Cipher text" was a common but incorrect answer.

(ii) Many mentioned the use of Alexa's public key but often did not make it clear that the key was retrieved from Alexa's digital certificate. Many candidates also described a hashing process but were not able to explain how this generated hash was used to authenticate the received message.

(iii) This was not answered well. Many answers lacked detail and focused more on security rather than the required authentication.

Question 3

This question was not answered well. There was little evidence that candidates understood virtual machines and how they can be used for testing software.

(a) (i) This question was not answered well. Any marks awarded were usually for hardware emulation.

(ii) Many candidates gained two marks for correctly explaining a feature of the guest operating system and a different feature of the host operating system. Many answers referred to the host operating system as the "original OS" or the "main OS". These answers gained no credit.

(b) (i) This question was not answered well. Candidates did not make full use of the given scenario and focus their answers on the use of virtual machines to test possible new operating systems and possible new webserver software. Incorrect responses included: using a virtual machine to run the old operating system, running the virtual machines on the current web server system, and assuming that virtual machines equated to virtual memory.

(ii) Many answers showed candidates were aware that there were issues with regard to the performance of a system and the same system running in a virtual machine. Many answers were often not expressed very clearly.

Question 4

In **part (a)** few candidates demonstrated a full understanding of the relationships between file organisations and file access methods. In **part (b)** candidates could often give the correct file organisation but were not able to give an adequate justification.

(a) Very few candidates gave all four correct responses. The majority of candidates scored two or three marks.

(b) (i) As either sequential or serial were correct answers, the vast majority of candidates scored at least one mark. Many candidates found it difficult to provide an appropriate justification for their selection. A few candidates stated clearly that as all customers would be given a statement, every record in the file would be processed and as a result, order was not an issue. For those who selected sequential, many incorrectly wrote that the order would be alphabetical, based on the name of the customer.

(ii) Many candidates correctly identified random as the correct organisation but often did not give a suitable justification. Correct responses for the justification tended to refer to the faster access to the data that random organisation allows.

(iii) Candidate answers to this part tended to be better than for the other two parts. Appending a new record and the resulting chronological order of the records were often given as answers.

Question 5

The majority of candidates demonstrated that they could construct truth tables, but only a few with total accuracy. Most candidates showed some knowledge of flip-flops.

(a) The majority of candidates provided a correct truth table for the NAND gate.

(b) (i) Fully correct tables were rare as were totally incorrect tables. Candidates were expected to work systematically, using the truth table for the NAND gate produced in **part (a)**, to complete the truth table for the given logic circuit. There was some indication that, because the logic circuit was that for a SR flip-flop, some candidates thought the values for Q and $\overline{Q}$ should be stated rather than generated. This applied in particular to the final row entry where values were sometimes given as "undefined" or "--".

(ii) Candidates often showed awareness that a particular combination of S and R should not occur, but often did not explain adequately why this should be the case. Marks were given for stating that Q and $\overline{Q}$ should not have the same value, as they are complements of each other. Candidates who wrote about an "invalid state" were given no credit as the question already states that one combination "should not be allowed to occur".

(c) (i) Many candidates gave the correct answer of a clock pulse. The word "clock" was required as any possible logic circuit abbreviations were considered too vague.

(ii) Only a minority of candidates could give an advantage of the JK flip-flop.

(d) Many answers mentioned some type of memory or referred to data storage. Fewer answer scored the second mark, due to not stating that a flip-flop can be used to store a single bit.

Question 6

The overwhelming majority of candidates were aware that the system described performed a monitoring function but many were incorrect in assuming that the system also exhibited control functions. There was good awareness of the different types of sensor but the reasons for inclusion of the sensor were often inadequate. The majority of candidates found it difficult to dry run fully the assembly language code and also to state the changes required to the code.

(a) (i) Many candidates correctly identified the system described as a monitoring system.

(ii) In giving their justification, it was often evident that many candidates believed, incorrectly, that the loud alarm was evidence of the system exhibiting control. However, it was the lack of any element of control that made the system a monitoring system.

(b) Temperature, sound and light sensors were not given credit for this particular system. A mark for the reason needed to be more specific than statements such as “a motion sensor detects motion”.

(c) (i) There were some excellent attempts at this question gaining full marks. Many answers displayed correct entries for COUNT and VALUE, and the first two entries for ACC. The mixture of binary and denary integers, which was meant to be helpful to the candidates, did not seem to cause any problems. The number of rows in the dry run table is an indication of the maximum space required for a full answer. Candidates should only make a further entry into a column when the value changes.

(ii) The operand ‘#1’ was only given by a few candidates. Some candidates gave the full instruction rather than just the operand; these answers did not get any credit.

(iii) Many candidates identified the required instruction. Only the most able candidates were able to give the correct operand for the amended instruction.

COMPUTER SCIENCE

Paper 9608/41
Written Paper

Key messages

It is essential that candidates have practical experience of programming using a high-level procedural language. Candidates must choose one of the following: Pascal/Delphi (console mode), Visual Basic.NET (console mode) or Python

Programming and pseudocode questions from past examination papers (including from the previous syllabus, 9691) provide an ideal starting point for practical work.

General comments

Many candidates clearly show they have experience of programming in a high-level language. There are a significant number of candidates who do not appear to distinguish pseudocode and the programming language being used.

Some candidates use the ← symbol (assignment) when an equality symbol is required. Candidates need to understand the difference between variables and literals when writing code. Frequently, variables are used in code with quotes around them.

Comments on specific questions

Question 1

This question referred to the topic on linked lists.

(a) Most candidates could write a pseudocode declaration of the required record type. The more able candidates declared an array of this type. Most candidates followed the instructions in the question which clearly stated that the array was to be of type `LinkedList`.

(b) Most candidates were able to read the linked list and find the correct surname and pointer value.

(c) The more able candidates realised that the missing entry in the identifier table was `IsFound` and of Boolean data type. The description was often very general, even though examples of what was required were given. The gaps in the pseudocode were generally completed well. The more able candidates clearly understood the technique of accessing a linked list, which is different to a sequential search. Candidates need to understand the mechanism of following pointers, rather than incrementing an index.

Question 2

This question involved recursion.

(a) Most candidates were able to describe a recursively defined procedure. Candidates need to understand that inferring repetition through iteration is not correct. A recursively defined procedure is a procedure that calls itself. Most candidates gave the correct line number where the recursive call was made.

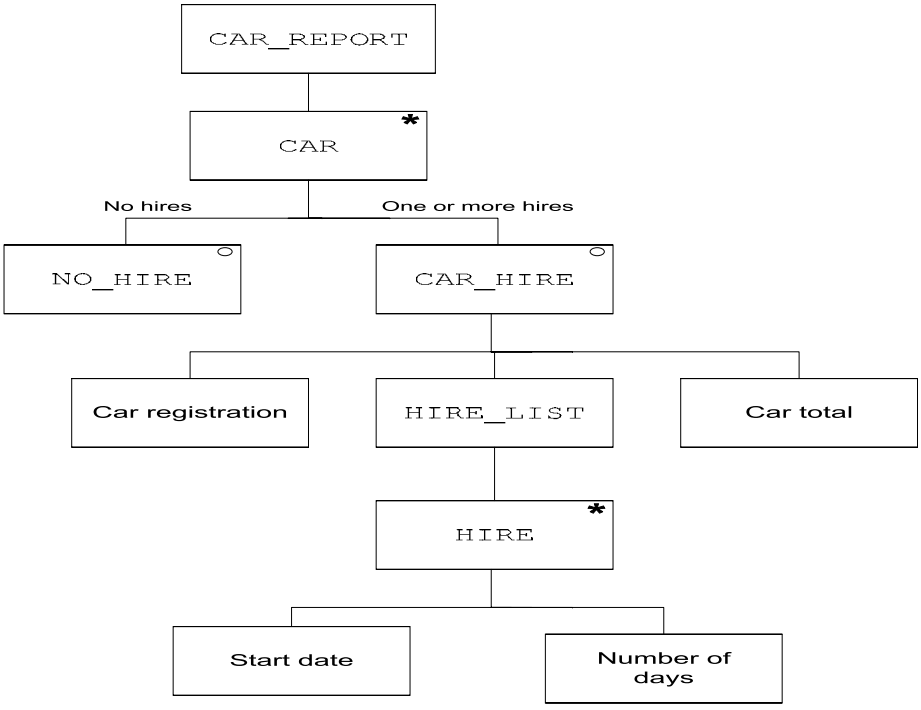
(b) Many candidates completed the trace table correctly. There is no need to enter values where these do not get re-assigned. This makes it easier to see what the effect of the algorithm is. Many candidates correctly stated the purpose of procedure x as removing the parameter value from the array. The more able candidates gave a more complete answer: the procedure deletes the first entry in the array that equals or is bigger than the parameter value. Some candidates gained credit for noting that the parameter value was overwritten by moving subsequent elements to the left.

Question 3

Jackson Structured Programming is a new topic. Many candidates made a good attempt in their responses.

(a) The data required to be added into this diagram was given in the question stem and most candidates arranged these in the correct hierarchy.

(b) Very few correct answers were seen. Many candidates found it difficult to select the correct data items needed for the required report. The question clearly stated that the report is to display the hire data for each car. Some candidates realised that the car registration needs to be only given once, whereas there may be more than one hire date and number of days hired. The more able candidates realised that the symbols for selection and repetition needed to be added to the diagram.



Question 4

This question was based on a knowledge base and logic programming.

(a) Most candidates correctly listed the values that would be returned for `DeadlineYear` of `2011`. Many candidates were able to interpret clause `22` as both tests needing to be passed.

(b) The more able candidates provided a suitable rule for this part of the question:

```
retestAllowed(ThisCar)
if (testBrakes(ThisCar, pass) and testTyres(ThisCar, fail))
or (testBrakes(ThisCar, fail) and testTyres(ThisCar, pass))
```

(c) (d) Many candidates were able to use the correct notation for lists and the empty list. Candidates need to understand that a list denoted `[X|Y]` will always produce a list for `Y`: the remainder of the list once the head, `A`, has been removed. The tail of a list with just one element is therefore an empty list, denoted `[]`

Question 5

Understanding of testing and exception handling was required for this question.

(a) Most candidates gave suitable example values for normal and extreme/boundary test data. The more able candidates realised that abnormal test data values (such as non-integer values or negative integers) should produce an error message. Many candidates incorrectly gave FAIL as an answer. This was the grade for any marks under 40, not an error message. Some candidates gave ranges for the marks and gained no credit. Candidates need to understand that such a test data table can be used for black box testing because the input and expected result is listed in the table and can be compared with the actual result when running the program with the given test data.

(b) Exceptions and exception handling are not well understood. This topic can be made more accessible if candidates are given the opportunity to use exception handling in practical programming exercises. It is not sufficient to state that an exception is an error situation. An exception is a run-time error (a program crash). This can be handled by providing code which is called when a run-time error occurs. This code may output an error message or take other action in order to avoid the program crashing.

(c) The more able candidates were able to state the possible exception errors that should be anticipated when a text file is read to populate an array. Some candidates did not appreciate that validation errors would not usually be caught using exception handling.

(d) The responses to this question part showed that a significant number of candidates did not have sufficient practical experience of programming exception handling routines. The more able candidates explained that Line `11` catches exceptions caused by the instructions between lines `05` and `10`. Line `11` stops the program from crashing and assigns the relevant exception type to ThisException. Line `12` then outputs the error message that is assigned to the type of exception caught by Line `11`.

Question 6

This question required candidates to write program code and the more able candidates provided some good answers. Those candidates who clearly had very little knowledge of how to write even straight-forward programming statements in their chosen programming language **part (b)** found this very challenging.

(a) Most candidates correctly added the missing transitions. The more able candidates also added the starting transition to represent that fact that WHITE always makes the first move.

(b) **Part (b)(i)** required candidates to declare and initialise a 2D array to represent the game board. Many candidates did not seem to understand and remember how an array is declared in their chosen programming language. This lack of understanding made **part (b)(ii)** a challenge as individual cells had to be tested to find out which moves were possible. Candidates need to understand that when presented with a question that involves some problem solving, it is important to think about the method of solution before embarking on writing down code. This can be a time-saving approach. The more able candidates clearly had lots of practical programming experience and remembered even the smallest detail of syntax of their chosen programming language. Some candidates did not notice that the values for `PieceColour`, `xCurrent` and `yCurrent` would be passed into the procedure as parameters and therefore wasted time writing lots of input statements to read this data in from the console. Here is an example solution written in Python:

```
def ValidMoves(PieceColour, xCurrent, yCurrent) :
    print("Possible moves are : ")
    if xCurrent != 1 :
        print("Moving LEFT . . .")
        i = xCurrent - 1
        NoFurther = False
        while i > 0 and NoFurther == False :
         if Board[yCurrent][i] == "E" :
               print(str(yCurrent) + " " + str(i))
               i = i - 1
           elif Board[yCurrent][i] != PieceColour :
               print(str(yCurrent) + " " + str(i) + " REMOVE PIECE")
               NoFurther = True
           else:
               NoFurther = True
    if xCurrent != 8 :
        print("Moving RIGHT . . .")
        i = xCurrent + 1
        NoFurther = False
        while i < 9 and NoFurther == False :
           if Board[yCurrent][i] == "E" :
               print(str(yCurrent) + " " + str(i))
               i = i + 1
           elif Board[yCurrent][i] != PieceColour :
               print(str(yCurrent) + " " + str(i) + " REMOVE PIECE")
               NoFurther = True
           else:
               NoFurther = True

Board = [["" for j in range(9)] for i in range(9)]
for Row in range(1, 3) :
       for Column in range(1, 9) :
              Board[Row][Column] = "B"
for Row in range(3, 7) :
       for Column in range(1, 9) :
              Board[Row][Column] = "E"
for Row in range(7, 9) :
       for Column in range(1, 9) :
              Board[Row][Column] = "W"
```

(c) This question part required candidates to analyse the scenario and suggest relevant classes and objects. The more able candidates suggested that `Piece` would be a suitable class, with objects of that class representing white and black pieces. Appropriate properties would therefore be colour and `x,y` position. Possible methods would be to move a piece to a given `x,y` position and analysing the possible moves from the current `x,y` position. **Part (b)(ii)** asked candidates to write the procedure `ValidMove`. Many candidates incorrectly suggested `ValidMove` could be a class and this shows a lack of basic understanding of object-oriented design. Procedures are methods.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

It is essential that candidates have practical experience of programming using a high-level procedural language. Candidates must choose one of the following: Pascal/Delphi (console mode), Visual Basic.NET (console mode) or Python

Programming and pseudocode questions from past examination papers (including from the previous syllabus, 9691) provide an ideal starting point for practical work.

General comments

Many candidates clearly show they have experience of programming in a high-level language. There are a significant number of candidates who do not appear to distinguish pseudocode and the programming language being used.

Some candidates use the ← symbol (assignment) when an equality symbol is required. Candidates need to understand the difference between variables and literals when writing code. Frequently, variables are used in code with quotes around them.

Comments on specific questions

Question 1

This question referred to the topic on linked lists.

(a) Most candidates could write a pseudocode declaration of the required record type. The more able candidates declared an array of this type. Most candidates followed the instructions in the question which clearly stated that the array was to be of type `LinkedList`.

(b) Most candidates were able to read the linked list and find the correct surname and pointer value.

(c) The more able candidates realised that the missing entry in the identifier table was `IsFound` and of Boolean data type. The description was often very general, even though examples of what was required were given. The gaps in the pseudocode were generally completed well. The more able candidates clearly understood the technique of accessing a linked list, which is different to a sequential search. Candidates need to understand the mechanism of following pointers, rather than incrementing an index.

Question 2

This question involved recursion.

(a) Most candidates were able to describe a recursively defined procedure. Candidates need to understand that inferring repetition through iteration is not correct. A recursively defined procedure is a procedure that calls itself. Most candidates gave the correct line number where the recursive call was made.

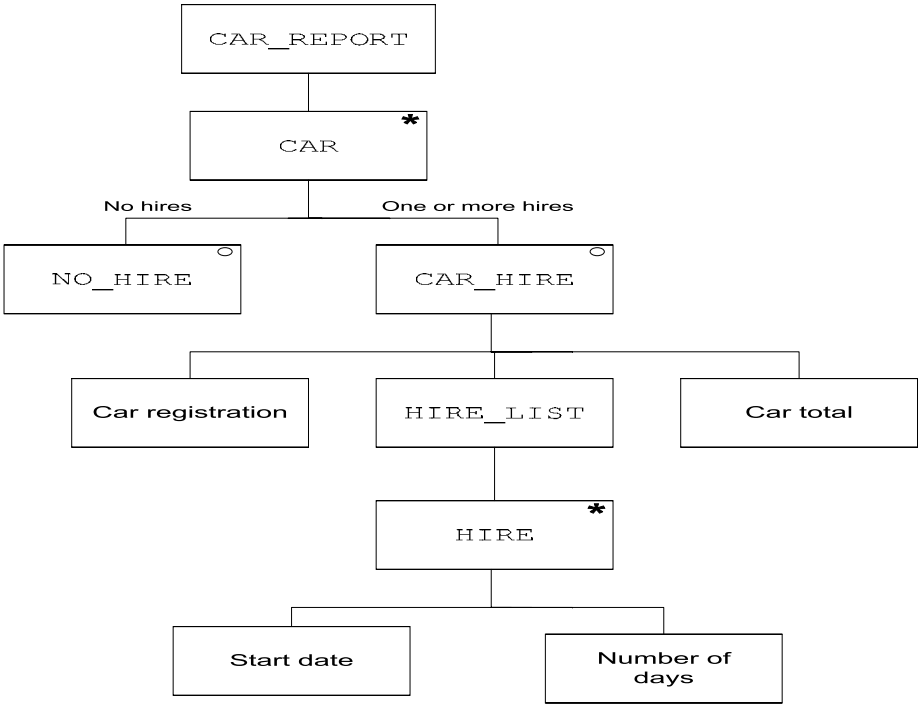
(b) Many candidates completed the trace table correctly. There is no need to enter values where these do not get re-assigned. This makes it easier to see what the effect of the algorithm is. Many candidates correctly stated the purpose of procedure x as removing the parameter value from the array. The more able candidates gave a more complete answer: the procedure deletes the first entry in the array that equals or is bigger than the parameter value. Some candidates gained credit for noting that the parameter value was overwritten by moving subsequent elements to the left.

Question 3

Jackson Structured Programming is a new topic. Many candidates made a good attempt in their responses.

(a) The data required to be added into this diagram was given in the question stem and most candidates arranged these in the correct hierarchy.

(b) Very few correct answers were seen. Many candidates found it difficult to select the correct data items needed for the required report. The question clearly stated that the report is to display the hire data for each car. Some candidates realised that the car registration needs to be only given once, whereas there may be more than one hire date and number of days hired. The more able candidates realised that the symbols for selection and repetition needed to be added to the diagram.



Question 4

This question was based on a knowledge base and logic programming.

(a) Most candidates correctly listed the values that would be returned for `DeadlineYear` of `2011`. Many candidates were able to interpret clause `22` as both tests needing to be passed.

(b) The more able candidates provided a suitable rule for this part of the question:

```
retestAllowed(ThisCar)
if (testBrakes(ThisCar, pass) and testTyres(ThisCar, fail))
or (testBrakes(ThisCar, fail) and testTyres(ThisCar, pass))
```

(c) (d) Many candidates were able to use the correct notation for lists and the empty list. Candidates need to understand that a list denoted `[X|Y]` will always produce a list for `Y`: the remainder of the list once the head, `A`, has been removed. The tail of a list with just one element is therefore an empty list, denoted `[]`

Question 5

Understanding of testing and exception handling was required for this question.

(a) Most candidates gave suitable example values for normal and extreme/boundary test data. The more able candidates realised that abnormal test data values (such as non-integer values or negative integers) should produce an error message. Many candidates incorrectly gave FAIL as an answer. This was the grade for any marks under 40, not an error message. Some candidates gave ranges for the marks and gained no credit. Candidates need to understand that such a test data table can be used for black box testing because the input and expected result is listed in the table and can be compared with the actual result when running the program with the given test data.

(b) Exceptions and exception handling are not well understood. This topic can be made more accessible if candidates are given the opportunity to use exception handling in practical programming exercises. It is not sufficient to state that an exception is an error situation. An exception is a run-time error (a program crash). This can be handled by providing code which is called when a run-time error occurs. This code may output an error message or take other action in order to avoid the program crashing.

(c) The more able candidates were able to state the possible exception errors that should be anticipated when a text file is read to populate an array. Some candidates did not appreciate that validation errors would not usually be caught using exception handling.

(d) The responses to this question part showed that a significant number of candidates did not have sufficient practical experience of programming exception handling routines. The more able candidates explained that Line `11` catches exceptions caused by the instructions between lines `05` and `10`. Line `11` stops the program from crashing and assigns the relevant exception type to ThisException. Line `12` then outputs the error message that is assigned to the type of exception caught by Line `11`.

Question 6

This question required candidates to write program code and the more able candidates provided some good answers. Those candidates who clearly had very little knowledge of how to write even straight-forward programming statements in their chosen programming language **part (b)** found this very challenging.

(a) Most candidates correctly added the missing transitions. The more able candidates also added the starting transition to represent that fact that WHITE always makes the first move.

(b) **Part (b)(i)** required candidates to declare and initialise a 2D array to represent the game board. Many candidates did not seem to understand and remember how an array is declared in their chosen programming language. This lack of understanding made **part (b)(ii)** a challenge as individual cells had to be tested to find out which moves were possible. Candidates need to understand that when presented with a question that involves some problem solving, it is important to think about the method of solution before embarking on writing down code. This can be a time-saving approach. The more able candidates clearly had lots of practical programming experience and remembered even the smallest detail of syntax of their chosen programming language. Some candidates did not notice that the values for `PieceColour`, `xCurrent` and `yCurrent` would be passed into the procedure as parameters and therefore wasted time writing lots of input statements to read this data in from the console. Here is an example solution written in Python:

```
def ValidMoves(PieceColour, xCurrent, yCurrent) :
    print("Possible moves are : ")
    if xCurrent != 1 :
        print("Moving LEFT . . .")
        i = xCurrent - 1
        NoFurther = False
        while i > 0 and NoFurther == False :
         if Board[yCurrent][i] == "E" :
               print(str(yCurrent) + " " + str(i))
               i = i - 1
           elif Board[yCurrent][i] != PieceColour :
               print(str(yCurrent) + " " + str(i) + " REMOVE PIECE")
               NoFurther = True
           else:
               NoFurther = True
    if xCurrent != 8 :
        print("Moving RIGHT . . .")
        i = xCurrent + 1
        NoFurther = False
        while i < 9 and NoFurther == False :
           if Board[yCurrent][i] == "E" :
               print(str(yCurrent) + " " + str(i))
               i = i + 1
           elif Board[yCurrent][i] != PieceColour :
               print(str(yCurrent) + " " + str(i) + " REMOVE PIECE")
               NoFurther = True
           else:
               NoFurther = True

Board = [["" for j in range(9)] for i in range(9)]
for Row in range(1, 3) :
       for Column in range(1, 9) :
              Board[Row][Column] = "B"
for Row in range(3, 7) :
       for Column in range(1, 9) :
              Board[Row][Column] = "E"
for Row in range(7, 9) :
       for Column in range(1, 9) :
              Board[Row][Column] = "W"
```

(c) This question part required candidates to analyse the scenario and suggest relevant classes and objects. The more able candidates suggested that `Piece` would be a suitable class, with objects of that class representing white and black pieces. Appropriate properties would therefore be colour and `x,y` position. Possible methods would be to move a piece to a given `x,y` position and analysing the possible moves from the current `x,y` position. **Part (b)(ii)** asked candidates to write the procedure `ValidMove`. Many candidates incorrectly suggested `ValidMove` could be a class and this shows a lack of basic understanding of object-oriented design. Procedures are methods.

COMPUTER SCIENCE

Paper 9608/43
Written Paper

Key messages

It is essential that candidates have practical experience of programming using a high-level procedural language. Candidates must choose one of the following: Pascal/Delphi (console mode), Visual Basic.NET (console mode) or Python

Programming and pseudocode questions from past examination papers (including from the previous syllabus, 9691) provide an ideal starting point for practical work.

General comments

Many candidates clearly show they have experience of programming in a high-level language. There are a significant number of candidates who do not appear to distinguish pseudocode and the programming language being used.

Some candidates use the ← symbol (assignment) when an equality symbol is required. Candidates need to understand the difference between variables and literals when writing code. Frequently, variables are used in code with quotes around them.

Comments on specific questions

Question 1

This question referred to the topic on linked lists.

(a) Most candidates could write a pseudocode declaration of the required record type. The more able candidates declared an array of this type. Most candidates followed the instructions in the question which clearly stated that the array was to be of type `LinkedList`.

(b) Most candidates were able to read the linked list and find the correct surname and pointer value.

(c) The more able candidates realised that the missing entry in the identifier table was `IsFound` and of Boolean data type. The description was often very general, even though examples of what was required were given. The gaps in the pseudocode were generally completed well. The more able candidates clearly understood the technique of accessing a linked list, which is different to a sequential search. Candidates need to understand the mechanism of following pointers, rather than incrementing an index.

Question 2

This question involved recursion.

(a) Most candidates were able to describe a recursively defined procedure. Candidates need to understand that inferring repetition through iteration is not correct. A recursively defined procedure is a procedure that calls itself. Most candidates gave the correct line number where the recursive call was made.

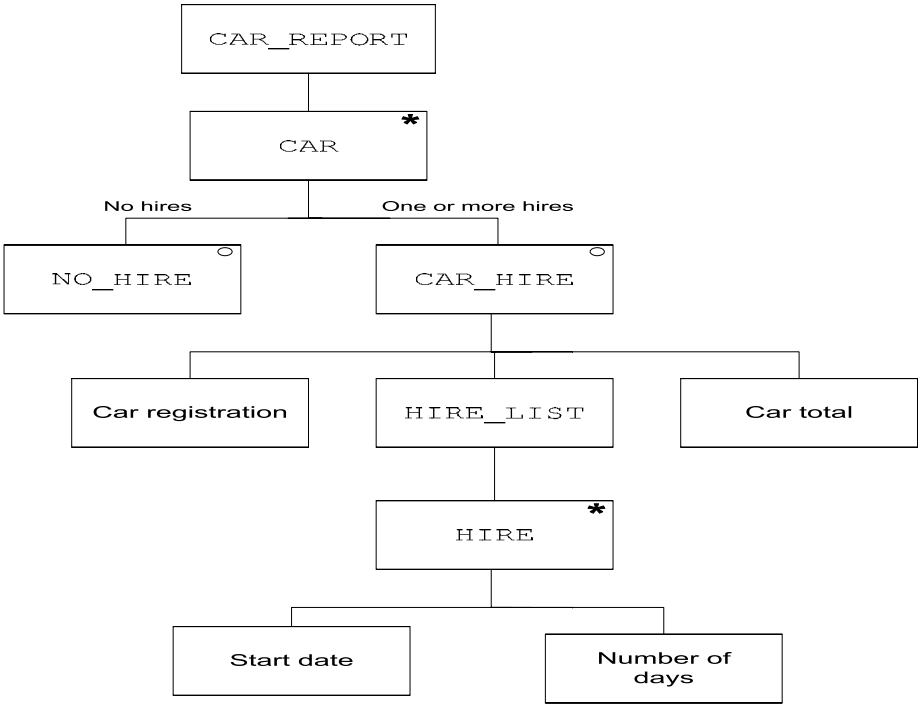
(b) Many candidates completed the trace table correctly. There is no need to enter values where these do not get re-assigned. This makes it easier to see what the effect of the algorithm is. Many candidates correctly stated the purpose of procedure x as removing the parameter value from the array. The more able candidates gave a more complete answer: the procedure deletes the first entry in the array that equals or is bigger than the parameter value. Some candidates gained credit for noting that the parameter value was overwritten by moving subsequent elements to the left.

Question 3

Jackson Structured Programming is a new topic. Many candidates made a good attempt in their responses.

(a) The data required to be added into this diagram was given in the question stem and most candidates arranged these in the correct hierarchy.

(b) Very few correct answers were seen. Many candidates found it difficult to select the correct data items needed for the required report. The question clearly stated that the report is to display the hire data for each car. Some candidates realised that the car registration needs to be only given once, whereas there may be more than one hire date and number of days hired. The more able candidates realised that the symbols for selection and repetition needed to be added to the diagram.



Question 4

This question was based on a knowledge base and logic programming.

(a) Most candidates correctly listed the values that would be returned for `DeadlineYear` of `2011`. Many candidates were able to interpret clause `22` as both tests needing to be passed.

(b) The more able candidates provided a suitable rule for this part of the question:

```
retestAllowed(ThisCar)
if (testBrakes(ThisCar, pass) and testTyres(ThisCar, fail))
or (testBrakes(ThisCar, fail) and testTyres(ThisCar, pass))
```

(c) (d) Many candidates were able to use the correct notation for lists and the empty list. Candidates need to understand that a list denoted `[X|Y]` will always produce a list for `Y`: the remainder of the list once the head, `A`, has been removed. The tail of a list with just one element is therefore an empty list, denoted `[]`

Question 5

Understanding of testing and exception handling was required for this question.

(a) Most candidates gave suitable example values for normal and extreme/boundary test data. The more able candidates realised that abnormal test data values (such as non-integer values or negative integers) should produce an error message. Many candidates incorrectly gave FAIL as an answer. This was the grade for any marks under 40, not an error message. Some candidates gave ranges for the marks and gained no credit. Candidates need to understand that such a test data table can be used for black box testing because the input and expected result is listed in the table and can be compared with the actual result when running the program with the given test data.

(b) Exceptions and exception handling are not well understood. This topic can be made more accessible if candidates are given the opportunity to use exception handling in practical programming exercises. It is not sufficient to state that an exception is an error situation. An exception is a run-time error (a program crash). This can be handled by providing code which is called when a run-time error occurs. This code may output an error message or take other action in order to avoid the program crashing.

(c) The more able candidates were able to state the possible exception errors that should be anticipated when a text file is read to populate an array. Some candidates did not appreciate that validation errors would not usually be caught using exception handling.

(d) The responses to this question part showed that a significant number of candidates did not have sufficient practical experience of programming exception handling routines. The more able candidates explained that Line `11` catches exceptions caused by the instructions between lines `05` and `10`. Line `11` stops the program from crashing and assigns the relevant exception type to ThisException. Line `12` then outputs the error message that is assigned to the type of exception caught by Line `11`.

Question 6

This question required candidates to write program code and the more able candidates provided some good answers. Those candidates who clearly had very little knowledge of how to write even straight-forward programming statements in their chosen programming language **part (b)** found this very challenging.

(a) Most candidates correctly added the missing transitions. The more able candidates also added the starting transition to represent that fact that WHITE always makes the first move.

(b) **Part (b)(i)** required candidates to declare and initialise a 2D array to represent the game board. Many candidates did not seem to understand and remember how an array is declared in their chosen programming language. This lack of understanding made **part (b)(ii)** a challenge as individual cells had to be tested to find out which moves were possible. Candidates need to understand that when presented with a question that involves some problem solving, it is important to think about the method of solution before embarking on writing down code. This can be a time-saving approach. The more able candidates clearly had lots of practical programming experience and remembered even the smallest detail of syntax of their chosen programming language. Some candidates did not notice that the values for `PieceColour`, `xCurrent` and `yCurrent` would be passed into the procedure as parameters and therefore wasted time writing lots of input statements to read this data in from the console. Here is an example solution written in Python:

```
def ValidMoves(PieceColour, xCurrent, yCurrent) :
    print("Possible moves are : ")
    if xCurrent != 1 :
        print("Moving LEFT . . .")
        i = xCurrent - 1
        NoFurther = False
        while i > 0 and NoFurther == False :
         if Board[yCurrent][i] == "E" :
               print(str(yCurrent) + " " + str(i))
               i = i - 1
           elif Board[yCurrent][i] != PieceColour :
               print(str(yCurrent) + " " + str(i) + " REMOVE PIECE")
               NoFurther = True
           else:
               NoFurther = True
    if xCurrent != 8 :
        print("Moving RIGHT . . .")
        i = xCurrent + 1
        NoFurther = False
        while i < 9 and NoFurther == False :
           if Board[yCurrent][i] == "E" :
               print(str(yCurrent) + " " + str(i))
               i = i + 1
           elif Board[yCurrent][i] != PieceColour :
               print(str(yCurrent) + " " + str(i) + " REMOVE PIECE")
               NoFurther = True
           else:
               NoFurther = True

Board = [["" for j in range(9)] for i in range(9)]
for Row in range(1, 3) :
       for Column in range(1, 9) :
              Board[Row][Column] = "B"
for Row in range(3, 7) :
       for Column in range(1, 9) :
              Board[Row][Column] = "E"
for Row in range(7, 9) :
       for Column in range(1, 9) :
              Board[Row][Column] = "W"
```

(c) This question part required candidates to analyse the scenario and suggest relevant classes and objects. The more able candidates suggested that `Piece` would be a suitable class, with objects of that class representing white and black pieces. Appropriate properties would therefore be colour and `x,y` position. Possible methods would be to move a piece to a given `x,y` position and analysing the possible moves from the current `x,y` position. **Part (b)(ii)** asked candidates to write the procedure `ValidMove`. Many candidates incorrectly suggested `ValidMove` could be a class and this shows a lack of basic understanding of object-oriented design. Procedures are methods.