COMPUTING

Paper 9691/11

Written Paper

General comments

Candidates seemed better prepared for this examination than in previous years. There were fewer very weak scripts. A number of candidates interpreted some questions in ways that were not expected; if the response was sensible, credit was given to the candidate. Many other candidates at times appeared to be answering questions that had been asked about similar, *but not the same*, topics from previous papers.

Comments on specific questions

Question 1

(a) This question was about the POS EFT interface in a typical supermarket checkout. This is a well known system and the question was answered accordingly, with many candidates gaining full marks. Those who gained less than full credit lost marks by giving incomplete reasons for the item of hardware they had given.

(b) This question was a typical one about printing a complete sales file at the end of the day. This was the scenario where candidates had to explain the use of a buffer and interrupts to make this process work. Candidate answers lost marks by being incomplete. For example, many candidates simply stated that data is sent to a buffer, but did not state *which* buffer. What they needed to say in this scenario is that data is sent to a printer buffer. Many answers simply defined a buffer and an interrupt, but did not relate any other part of the answer to the scenario and so did not answer the question. A small number of answers mixed up buffers and queues.

Question 2

(i) The three parts of this question were about utility software. Part **(i)** was about a disk formatter. Many candidates were correct by stating it is used to prepare a disk for use. Better responses also mentioned the preparation of tracks and sectors, though some candidates managed to get this the wrong way round saying that tracks were part of sectors. Only a very few mentioned the need to prepare the FAT or equivalent file system.

(ii) This question was about printer drivers and was the most poorly answered of the three parts. Most candidates mentioned the use in allowing communication between printer and computer, however, almost no one mentioned that it is necessary to translate the codes and characters sent from a computer into a form a printer can understand and use. This is simply an extension of the communication idea, and is the sort of thing that should be in an A Level answer.

(iii) For most candidates, the virus checker utility earned a lot of marks. Most knew that they checked for and removed or quarantined viruses, but few said it worked in the background or that it was a continuous process.

Question 3

This question asked candidates to match a series of statements to a series of computing terms. A large number of candidates gained full marks. However, many candidates mixed up broadband and baseband, and others mixed up packet switching with parallel communication.

Question 4

(a) This question was about monitoring pollution levels, and the candidates were asked to find appropriate sensors to monitor water pollution. There were only two sensible sensors but many candidates chose three or even four sensors from the list, some of which were not appropriate. For example, a humidity sensor would be of no use in monitoring water pollution.

(b) (i) This question asked for the benefits of using a dedicated line to send data from the sensors to a computer some distance away. Many candidates appeared to think that a dedicated line would allow no data corruption, which is obviously not true. The majority of reponses involved better security and the likelihood that data could be transferred faster than on a shared network. It was very unusual to see an answer that included the fact that a dedicated line would be always available.

(ii) This question was about serial and full duplex data transmission, it was very well answered. If a candidate did not get full marks they usually lost one because they forgot that serial transport uses only a single wire. It has been mentioned before that cable is *not* a wire, but this still appears far too often. Weak answers usually compared simplex and full duplex rather than the question that was asked.

(iii) This question continued the pollution scenario but asked about how a sensor could transmit data when monitoring pollution. Far too many candidates stated that data would be transmitted to a computer, when it should be transmitted to the processor. Almost all responses mentioned the need for ADC hardware *but* this must be stated with the detail that the sensor collected analogue data to make this necessary. Again, precision is required in an A Level answer but many answers were simply far too general. A rather large number of candidates also talked about data logging and data loggers but they were not involved in the scenario.

Question 5

(a) This question asked for benefits and drawbacks of three local area networks. This question has been asked before in various formats and this showed in the generally good answers provided by the candidates. Weak responses seemed to concentrate on the ease of spreading viruses through a network, and many others also confused security with privacy, which are really very different ideas. A number of candidates just described the topology and not the benefits or drawbacks.

(b) This was a very general question about the hardware found in a typical local area network. Candidates had to name and describe the items. Many however, only named the items or provided weak descriptions, which severely limited the number of marks available to them. Many candidates gave a modem as an item of hardware, but that would not be a sensible way to connect a LAN to the internet.

Question 6

(a) This question required the candidates to draw a logic circuit from a list of conditions given. Drawings produced were often very wrong. There were still far too many AND or OR gates that were drawn with just a single input. There were rather few answers between the maximum and minimum possible. Some candidates are still not drawing gates with the proper shaped symbol.

(b) This was a question about truth tables and is a skill that has consistently improved over the years. This was again the case on this paper. However, some candidates answered by putting 1's in all the output rows; this is not a way to answer such questions and could easily get a zero result.

Question 7

(a) This question was about completing a flowchart and as such has not appeared many times before. However, candidates did very well with the majority gaining full marks. Some did not answer the question properly since they did not put the appropriate statement number into the boxes provided. Some candidates managed to mix up statements 3 and 5, whilst others switched statements 6 and 4, and hence lost the marks available.

(b) This question was about the system life cycle but presented in a different way than is normally seen. It asked candidates to describe three stages that *follow* the design phase. Unfortunately, many candidates described stages that happen *before* the design phase and so lost all the marks available. Answers were often extremely vague and not up to the standard expected in an A Level answer. Many candidates also described things entirely unrelated to the stated scenario.

(c) The scenario from **part (b)** continued in this question, which gave one form of system maintenance and asked the candidates to state two others and describe how they are used. The majority of candidates gave the correct two kinds of maintenance and nearly all gave correct descriptions. Where marks were lost, it was for the usual reason that answers did not contain sufficient detail to gain credit.

(d) (i) This question asked candidates to convert a binary number to a denary value and was generally answered very well. Most candidates gained two marks but where answers were wrong they usually involved the second answer being just one different from the expected answer; possibly just an addition error.

(ii) Here candidates were asked to do the reverse, i.e. convert a denary number into binary, and this was answered even better with very few wrong answers.

Question 8

(a) (i) This whole question involved a chemical plant scenario and was essentially about interfaces, but presented in a very different way from previous years. The first part was simple observation and quite straightforward, where candidates had to say what two symbols on the screen were for. The vast majority got the answer correct either stating they were navigation buttons or by describing their action.

(ii) This part asked how a graphic could be used to show an open/closed valve. Unfortunately, many candidates described text-based answers and so gained no credit. Candidates who did describe graphic answers, gave many ideas that were both correct and very inventive, which were not on the mark scheme but could gain credit.

(iii) A further interface question asked the candidates to show how a changing pressure could be expressed as a graphic, and again there were many quite inventive and correct responses. Once again, however, others described a text-based answer that did not answer the question.

(b) This question asked why a mouse would not be a suitable interface device in the chemical plant control room. Many candidates did not answer the question, since they talked about why a touch screen would be better rather than why a mouse would *not* be a good idea. Where candidates did talk about the mouse, they often focused on the fact that a mouse would be a slow method of response, rather than reasons why it would not work, i.e. a potential lack of space or that this could be a dirty environment that could clog a typical mouse. A trailing cable for a mouse is, however, not normally a trip hazard.

(c) The final question required candidates to give two reasons why computer control of a chemical plant would be safer than manual control. Many candidates concentrated on the idea of keeping humans away from a dangerous situation, which was not our first thought even though it was correct. Good answers were common and usually revolved around the speed of response to dangerous situations being better, as well as the potential for constant 24 hour cover by a computer. Another possible answer was never given, i.e. that the computer could monitor far more parameters at once than a human could ever do. Many candidates stated that computers were more accurate, which is not necessarily true as it would depend on how well they were programmed.

COMPUTING

Paper 9691/12

Written Paper 1

General comments

Candidates seemed better prepared for this examination than in previous years. There appeared to be far fewer very weak scripts. There were a number of candidates who interpreted some questions in ways that were not expected. Many other candidates sometimes appeared to be answering questions that had been asked about a similar, *but not the same*, topic from a previous paper.

Comments on specific questions

Question 1

(a) (i) The types of generic software were well known by the majority of candidates but there were still many examples of brand-named software being used rather than the named type. It is clearly stated on the front cover of the Question Paper that brand names will gain no credit. The justifications for each item of software were not clearly described. Many candidates did not make specific reference to the science project in the justification and so made it much harder to gain credit for the answer.

(ii) Almost all the candidates gave the correct answer for the question. This is not surprising as the answer was stated in the stem of the question and all the candidate had to do was make an educated choice.

(iii) When asked for the benefits of “off the shelf” software many candidates gave answers that lacked sufficient detail to earn credit. Typically, answers stated “because it is cheaper” without giving the reason *why* it was cheaper, which is needed to gain credit; half a correct answer does not gain any marks.

(b) (i) Most candidates answered this part correctly. As before, this is not surprising since the answer was stated in the stem of the question and the candidate had to choose from one of two given answers.

(ii) When asked for the benefits of “custom written” software far fewer candidates gave answers that were worthy of credit. This was often because the candidates had not read the question properly and answered in terms of a school, rather than the examination board, which was clearly stated in the question stem. Other candidates lost marks by giving the same answer twice but worded in a different way.

Question 2

When asked for the difference between a command line interface (CLI) and a graphical user interface (GUI) some candidates did not appear to fully understand what a CLI was, whereas others performed far better when answering about a CLI than a GUI. The majority of candidates did not gain marks for the GUI because they only gave part of the expected answer. Other candidates answered in terms of the experience of the user and the ease of use (or not) of the appropriate user interface. Many did not state the need to click on icons to initiate processes with a GUI or that a CLI gives a user far more control over a computer system.

Question 3

This question was about the system life cycle. Candidates were given a diagram with some stages already entered and they were asked to complete the diagram by naming the appropriate stage. Some answers were word-perfect suggesting that the diagram might be available in a text book. However, many candidates really struggled to complete the diagram, typically naming stages in the wrong part of the diagram or even naming non-existent stages of the system life cycle. The two named stages given in the question should have given clear clues about where to enter the appropriate stages. It seems that learning the system life cycle is being neglected now that the practical project has been lost from the syllabus.

Question 4

(a) This question concerned the roles of ROM and RAM. It was presented in diagrammatic form and many candidates obtained full credit. Some candidates mixed up some of the descriptions. Other candidates drew two lines from one description to both ROM and RAM, one of which was correct and one of which was wrong; an Examiner cannot make a choice between these.

(b) This question asked for the roles of RAM, ROM and solid state memory in a typical digital camera. However, many candidates gave standard answers in terms of a computer not a camera; it is important to answer the question that has been asked rather than a question the candidate *thinks* has been asked. With regard to the solid state memory, very many candidates talked about pen drives rather than the "Compact Flash or SDHD" cards that would be used to store the images that were taken.

(c) The majority of the candidates knew the answer to how images could be transferred from the camera to a computer. However, many answers were incomplete and candidates failed to gain credit for partial answers. A typical example was "*to use a USB*", which was simply not enough to gain credit.

(d) (i)(ii) Part **(i)** asked what is meant by file compression. Many candidates simply said 'making them smaller', which is not accurate enough to gain marks. Other candidates used the intended answer for part **(ii)** (one reason why file compression is used) as the answer for part **(i)** and then tried to find something different for the part **(ii)** answer but could not do so.

(e) This question asked about the use of buffers and interrupts when transferring images from a camera to a computer. It was very poorly answered by the majority of candidates for a variety of reasons. Firstly, many candidates failed to say where the buffer was or which buffer it was; with modern sophisticated cameras it could be in the camera or the computer. Others simply defined what a buffer and an interrupt was and gave no other answer. There is still a common misconception that an interrupt is only generated in an error situation. Finally, many candidates think it is the buffer itself that generated the interrupt, which is definitely not the case.

Question 5

This question was presented in diagrammatic form, where a series of descriptions were to be matched to a series of computer terms. Many candidates produced responses that gained high marks; the majority earning from 3 to 5 marks. The most common mistake by candidates was to mix up formatter and operating system.

Question 6

(a) This question was about a computer controlled entry/security system. Candidates were given a scenario and asked how a computer system would decide whether to grant or refuse entry to the workplace. Most candidates appreciated that the photograph that was taken was somehow compared to another, either stored on a database or within the card itself. Others also mentioned about matching the other card details to pre-stored data. The strongest responses mentioned that both tests had to be passed before the worker was allowed to enter. What was almost entirely lacking in answers was the need for worker to enter a PIN to confirm the card was theirs. Unfortunately, many answers were vague and incomplete.

(b) This question concerned definitions for both full duplex and serial transmission. Generally this was well answered. However, there were a surprisingly large number that defined duplex and simplex, an answer to a question that was not asked; candidates should carefully read the question before answering. Most candidates defined duplex correctly far fewer did so for serial transmission.

(c) This question asked about the various parts of an expert system. This is a part of the syllabus that seems to be rarely answered well. It is a question that has been asked in various forms over the years but the answers given never seem to improve. Perhaps this part of the syllabus would benefit from greater coverage.

Question 7

(a) Questions involving creating a logic circuit have appeared often over the years and it appears that the skill of creating these diagrams has improved considerably. There were many completely correct diagrams, and even more pleasing was the fact that some candidates had correctly simplified the logic circuit and drew an alternative correct diagram. However, many candidates drew what was almost a correct diagram but omitted all the NOT gates and so lost the majority of the marks available. A few candidates did simplify the NOT A AND NOT C but used a NAND gate rather than a NOR gate. Finally others also simplified the pair of two input NOT gates into a single three input gate, which was a very good thing to do.

(b) There was a good number of completely correct answers for the truth table. Other candidates seemed to be guessing, by filling in the entire answer column with 1's; this time this did achieve some marks but is certainly not the way to properly answer this kind of question.

(c) (i) Converting a binary string to decimal is another skill that has improved over the years and this question further proved that point with the majority of answers being correct.

(ii) Converting a denary number back to binary is also a skill that has improved over the years. Some candidates however, did give an answer in denary with a different denary value.

(iii) This was a slightly more involved question requiring some thought rather than following a structured method. It asked for a binary string that would fit a defined fault condition of five consecutive 1's. Perhaps it was a misunderstanding of the word "consecutive" that caused some candidates to give answers that did not even contain five 1's. There were many correct forms of the answer, though most candidates chose to place the 1's at either the most significant or least significant ends of the string.

(c) (iv) Candidates were then asked for the minimum denary value that could signify the above error condition. This is another question that requires some thought rather than applying a rote answer. It is another answer that is either right or wrong but some candidates surprisingly answered in binary rather than in base 10.

Question 8

(a) This question was one of the more straightforward ones on the Question Paper as it was more about information handling. The complete answer was in the stem of the question but candidates had to apply a little knowledge to find it. The majority did so, and gained maximum marks in this part.

(b) This question was about computer interfaces but presented in a very different way from previous years. It was an almost free-form answer where marks could be awarded for many sensible ideas. Candidates often gave sensible suggestions, showing how inventive many candidates can be. Many gave only half the answer, since they mentioned how a graphic could be used but did not give a method of distinguishing which direction the train would be travelling.

(c) (i) Here the question asked about the term "refresh". Many candidates answered in terms of the system rather than updating the screen display, which we were expecting.

(ii) Candidates did not seem to appreciate why a dynamic ever-changing system would need to update the display, i.e. that the position, speed etc. of each train would be constantly changing.

(d) This final question appeared to be very hard for the majority of the candidates. Relatively few candidates gained full credit mainly because they could not think of appropriate specific software with which to supply the system. Many turned the question back on itself and simply described the scenario using different words, which though clever did not gain any credit. Many other responses involved talking about the interface rather than what was required to answer the question. Marks for the hardware part of the answer were easier to obtain but many candidates did not gain a mark since they used the singular sensor rather than sensors plural; a 150 km stretch of railway track could hardly be adequately covered using just one sensor. The answers were often generic and not applied to the particular scenario that the question asked for.

COMPUTING

Paper 9691/13

Written Paper

General comments

Candidates seemed better prepared for this examination than in previous years. There were fewer very weak scripts. A number of candidates interpreted some questions in ways that were not expected; if the response was sensible, credit was given to the candidate. Many other candidates at times appeared to be answering questions that had been asked about similar, *but not the same*, topics from previous papers.

Comments on specific questions

Question 1

(a) This question was about the POS EFT interface in a typical supermarket checkout. This is a well known system and the question was answered accordingly, with many candidates gaining full marks. Those who gained less than full credit lost marks by giving incomplete reasons for the item of hardware they had given.

(b) This question was a typical one about printing a complete sales file at the end of the day. This was the scenario where candidates had to explain the use of a buffer and interrupts to make this process work. Candidate answers lost marks by being incomplete. For example, many candidates simply stated that data is sent to a buffer, but did not state *which* buffer. What they needed to say in this scenario is that data is sent to a printer buffer. Many answers simply defined a buffer and an interrupt, but did not relate any other part of the answer to the scenario and so did not answer the question. A small number of answers mixed up buffers and queues.

Question 2

(i) The three parts of this question were about utility software. Part **(i)** was about a disk formatter. Many candidates were correct by stating it is used to prepare a disk for use. Better responses also mentioned the preparation of tracks and sectors, though some candidates managed to get this the wrong way round saying that tracks were part of sectors. Only a very few of candidates mentioned the need to prepare the FAT or equivalent file system.

(ii) This question was about printer drivers and was the most poorly answered of the three parts. Most candidates mentioned the use in allowing communication between printer and computer, however, almost no one mentioned that it is necessary to translate the codes and characters sent from a computer into a form a printer can understand and use. This is simply an extension of the communication idea, and is the sort of thing that should be in an A Level answer.

(iii) For most candidates, the virus checker utility earned a lot of marks. Candidates knew that they checked for and removed or quarantined viruses, but few said it worked in the background or that it was a continuous process.

Question 3

This question asked candidates to match a series of statements to a series of computing terms. A large number of candidates gained full marks. However, many candidates mixed up broadband and baseband, and others mixed up packet switching with parallel communication.

Question 4

(a) This question was about monitoring pollution levels, and the candidates were asked to find appropriate sensors to monitor water pollution. There were only two sensible sensors but many candidates chose three or even four sensors from the list, some of which were not appropriate. For example, a humidity sensor would be of no use in monitoring water pollution.

(b) (i) This question asked for the benefits of using a dedicated line to send data from the sensors to a computer some distance away. Many candidates appeared to think that a dedicated line would allow no data corruption, which is obviously not true. The majority of reponses involved better security and the likelihood that data could be transferred faster than on a shared network. It was very unusual to see an answer that included the fact that a dedicated line would be always available.

(ii) This question was about serial and full duplex data transmission, it was very well answered. If a candidate did not get full marks they usually lost one because they forgot that serial transport uses only a single wire. It has been mentioned before that cable is *not* a wire, but this still appears far too often. Weak answers usually compared simplex and full duplex rather than the question that was asked.

(iii) This question continued the pollution scenario but asked about how a sensor could transmit data when monitoring pollution. Far too many candidates stated that data would be transmitted to a computer, when it should be transmitted to the processor. Almost all responses mentioned the need for ADC hardware but this must be stated with the detail that the sensor collected analogue data to make this necessary. Again, precision is required in an A Level answer but many answers were simply far too general. A rather large number of candidates also talked about data logging and data loggers but they were not involved in the scenario.

Question 5

(a) This question asked for benefits and drawbacks of three local area networks. This question has been asked before in various formats and this showed in the generally good answers provided by the candidates. Weak responses seemed to concentrate on the ease of spreading viruses through a network, and many others also confused security with privacy, which are really very different ideas. A number of candidates just described the topology and not the benefits or drawbacks.

(b) This was a very general question about the hardware found in a typical local area network. Candidates had to name and describe the items. Many however, only named the items or provided weak descriptions, which severely limited the number of marks available to them. Many candidates gave a modem as an item of hardware, but that would not be a sensible way to connect a LAN to the internet.

Question 6

(a) This question required the candidates to draw a logic circuit from a list of conditions given. Drawings produced were often very wrong. There were still far too many AND or OR gates that were drawn with just a single input. There were rather few answers between the maximum and minimum possible. Some candidates are still not drawing gates with the proper shaped symbol.

(b) This was a question about truth tables and is a skill that has consistently improved over the years. This was again the case on this paper. Some candidates answered by putting 1s in all the output rows; this is not a way to answer such questions and could easily get a zero result.

Question 7

(a) This question was about completing a flowchart and as such has not appeared many times before. However, candidates did very well with the majority gaining full marks. Some did not answer the question properly since they did not put the appropriate statement number into the boxes provided. Some candidates managed to mix up statements 3 and 5, whilst others switched statements 6 and 4, and hence lost the marks available.

(b) This question was about the system life cycle but presented in a different way than is normally seen. It asked candidates to describe three stages that follow the design phase. Many candidates described stages that happen before the design phase and so lost all the marks available. Answers were often extremely vague and not up to the standard expected in an A Level answer. Many candidates also described things entirely unrelated to the stated scenario.

(c) The scenario from part **(b)** continued in this question, which gave one form of system maintenance and asked the candidates to state two others and describe how they are used. The majority of candidates gave the correct two kinds of maintenance and nearly all gave correct descriptions. Where marks were lost, it was for the usual reason that answers did not contain sufficient detail to gain credit.

(d) (i) This question asked candidates to convert a binary number to a denary value and was generally answered very well. Most candidates gained two marks but where answers were wrong they usually involved the second answer being just one different from the expected answer; possibly just an addition error.

(ii) Candidates were asked to do the reverse, i.e. convert a denary number into binary, and this was answered even better with very few incorrect answers.

Question 8

(a) (i) This whole question involved a chemical plant scenario and was essentially about interfaces, but presented in a very different way from previous years. The first part was simple observation and quite straightforward, where candidates had to state the purpose of the two symbols on the screen. The vast majority procided the correct answer, either stating that they were navigation buttons or by describing their action.

(ii) This part asked how a graphic could be used to show an open/closed valve. Many candidates described text-based answers and so did not gain credit. Candidates who did describe graphic answers, gave many ideas that were both correct and very inventive.

(iii) A further interface question asked the candidates to show how a changing pressure could be expressed as a graphic, and again there were many quite inventive and correct responses. Other candidates described a text-based answer that did not answer the question.

(b) This question asked why a mouse would not be a suitable interface device in the chemical plant control room. Many candidates did not answer the question, since they talked about why a touch screen would be better rather than why a mouse would not be a good idea. Where candidates did talk about the mouse, they often focused on the fact that a mouse would be a slow method of response, rather than reasons why it would not work, i.e. a potential lack of space or that this could be a dirty environment that could clog a typical mouse. A trailing cable for a mouse is, however, not normally a trip hazard.

(c) The final question required candidates to give two reasons why computer control of a chemical plant would be safer than manual control. Many candidates concentrated on the idea of keeping humans away from a dangerous situation, which was not our first thought even though it was correct. Good answers were common and usually revolved around the speed of response to dangerous situations being better, as well as the potential for constant 24 hour cover by a computer. Another possible answer was never given, i.e. that the computer could monitor far more parameters at once than a human could ever do. Many candidates stated that computers were more accurate, which is not necessarily true as it would depend on how well they were programmed.

COMPUTING

Paper 9691/21

Written Paper

Key messages

To succeed in this paper, it is essential that candidates have practical experience of programming using a high-level procedural language. It is recommended that candidates choose one of the following: Pascal, Visual Basic (console mode), or Python. Programming and pseudocode questions from past question papers provide an ideal starting point for the practical work.

General comments

There are a significant number of candidates who do not appear to have any programming experience. Some questions require an answer using a high-level programming language. Although minor discrepancies in syntax are ignored, candidates need to understand that giving pseudocode answers will not gain credit. Some questions ask for pseudocode solutions. Candidates giving answers using a real high-level programming language will be given credit for correct solutions.

Comments on specific questions

Question 1

(a) Many candidates could identify the correct data types but gave incorrect field sizes. A common error was to give a field size for an integer of 1. Candidates need to understand that a data type is of a fixed size. Most programming languages require 4 bytes to store an integer, 8 bytes to store a date, 4 bytes to store single and 8 bytes to store double real numbers, 8 bytes to store currency, 16 bytes to store decimal and 1 or 2 bytes to store a Boolean variable. Candidates need to understand that numeric codes starting with a zero have to be stored as strings.

(b) Most candidates were able to declare a record type in their chosen programming language. There appears to be some confusion between VB6 and VB.NET. Candidates should be aware of the programming language they are using and the keywords that are therefore appropriate.

(c) Many candidates stated that initialisation required to loop 50 times. Fewer candidates were then able to explain that a dummy value needed to be assigned to each field of every record. A common error was to assign NULL or the empty string to the whole array element.

(d) (i) Stronger responses were able to explain that the function `EOF()` returns a Boolean value depending on whether the marker at the end of the file was found at the next read attempt.

(ii) Only the stronger responses appeared to understand the concept of file handling. Many candidates were able to open and close the text file, but were not able to give the steps required to read the file and store the data in the array:

```
OPENFILE CourseData.DAT FOR READING
i ← 1
WHILE NOT EOF(CourseData.DAT)
     READ record from FILE
     Course[i] ← record
     i ← i + 1
ENDWHILE
CLOSEFILE CourseData.DAT
```

(e) Very few candidates recognised that the outer loop could be changed to a conditional loop and even fewer realised that the limit of the inner loop could be reduced to improve efficiency. Some candidates changed the `FOR` loop limits from the maximum of 50 to `NumberOfCourses`.

Question 2

(a) (i) Some candidates were able to write a `CASE` statement in their chosen programming language. Many candidates did not convert the pseudocode assignment operator ← to the relevant assignment symbol for their programming language (VB and Python use = and Pascal uses := for this). How a function returns a value is also not well known. Many candidates invented various input and output statements as part of their solution. Candidates need to understand that when converting a flowchart into program code, they need to translate only the steps that are given, not add other steps. Python programmers should use the `IF` … `ELIF` … construct for `CASE` statements.

(ii) Having just attempted to write program code for the flowchart, some candidates realised that `1` and `X` were invalid test data and the expected result should be `-1`. The majority of candidates correctly stated that G was normal input. A common error was to assume that one of those values must be borderline data.

(b) (i) Many candidates completed the trace table correctly. A common error was to output a value each time round the loop.

(ii) Many candidates correctly identified that line 8 needed correcting. Few candidates could give the correct statement:

```
Denary ← Denary * 10 + ThisNumber
```

(iii) The majority of candidates correctly stated that this was a logic error.

(iv) Many candidates correctly named two other types of error, syntax error and run-time error. The stronger responses were able explain that syntax errors were reported by translator diagnostics during compilation of the program, or when running an interpreted program, and run-time errors were found during execution when the program 'crashes' or 'freezes'.

(c) (i) The majority of candidates were able to list features found in the pseudocode such as meaningful identifiers, capitalisation of keywords, indentation and use of library/built-in functions. Just referring to white space is too vague.

(ii) Most candidates correctly identified comments as a feature that would have been useful to help the understanding of the pseudocode.

(iii) The requirement of this question was to implement the pseudocode given in **part (b)**. Candidates need to understand that they have to write program code for each line of pseudocode. Many candidates invented additional input and output statements or validation. To answer such questions correctly, candidates need to be able to convert the assignment symbol to the relevant symbol used in their programming language. Candidates also need to be able to convert pseudocode statements such as the FOR loop and the OUTPUT statement into the keywords used by their programming language. The syllabus requires that candidates know, in their chosen programming language, built-in functions including LENGTH and MID.

(iv) The stronger responses used the fact that the DenaryDigit function returns –1 for invalid characters. They suggested to use an IF statement to test for this, give an error message instead of the Denary assignment and exit the loop.

Question 3

(a) (i) The stronger repsonses correctly stated that a breakpoint could be set in program code and that execution would pause at this point. A common error was the misunderstanding that a breakpoint is the part of the program that stops working due to an error.

(ii) Many vague answers were seen here. Very few candidates could explain what stepping was in the context of debugging. Candidates need to have practical experience of using such debugging tools. Stepping is used to execute one statement at a time and program execution pauses after each statement.

(iii) The stronger responses were able to explain that the variable check allows a tester to choose which variables to watch. The variable watch window is then used to see how the variable contents change when stepping through a program.

(b) The majority of candidates were able to state that white box testing makes use of debugging tools.

COMPUTING

Paper 9691/22

Written Paper

Key messages

To succeed in this paper, it is essential that candidates have practical experience of programming using a high-level procedural language. It is recommended that candidates choose one of the following: Pascal, Visual Basic (console mode), or Python. Programming and pseudocode questions from past question papers provide an ideal starting point for practical work.

General comments

There are a significant number of candidates who do not appear to have any programming experience. Some questions require an answer using a high-level programming language. Although minor discrepancies in syntax are ignored, candidates need to understand that giving pseudocode answers will not gain credit. Some questions ask for pseudocode solutions. Candidates giving answers using a real high-level programming language will be given credit for correct solutions.

Comments on specific questions

Question 1

(a) Many candidates could identify the correct data types but gave incorrect field sizes. A common error was to give a field size for an integer of 1. Candidates need to understand that a data type is of a fixed size. Most programming languages require 4 bytes to store an integer, 8 bytes to store a date, 4 bytes to store single and 8 bytes to store double real numbers, 8 bytes to store currency, 16 bytes to store decimal and 1 or 2 bytes to store a Boolean variable. Candidates need to understand that codes such as an ISBN have to be stored as strings.

(b) Most candidates were able to declare a record type in their chosen programming language. There appears to be some confusion between VB6 and VB.NET. Candidates should be aware of the programming language they are using and the keywords that are therefore appropriate.

(c) Many candidates stated that initialisation required to loop 100 times. Fewer candidates were then able to explain that a dummy value needed to be assigned to each field of every record. A common error was to assign NULL or the empty string to the whole array element.

(d) The calculation of file size for storing the data of 10 books was generally well done. A minority of candidates did not show their working.

(e) Only the stronger responses appeared to understand the concept of file handling. Many candidates were able to open and close the text file, but were not able to give the steps required to save the array contents:

```
OPENFILE BookData.DAT FOR WRITING
FOR i ← 1 TO 100
   IF Book[i].Title > ""
      THEN
         WRITE record to FILE
   ENDIF
ENDFOR
CLOSEFILE BookData.DAT
```

(f) The stronger responses were able to explain that the function `EOF()` returns a Boolean value depending on whether the marker at the end of the file was found at the next read attempt.

(g) (i) Some candidates were able to check each book record in the array and store a pointer to the book with the best rating so far encountered. An example answer gaining full credit is:

```
BestSoFar ← 1
FOR i ← 2 TO 100
   IF Book[i].Rating > Book[BestSoFar].Rating
      THEN
         BestSoFar ← i
   ENDIF
ENDFOR
OUTPUT Book[BestSoFar].Title
```

Only the stronger responses were able to show the correct field notation when records are stored in an array.

(ii) Very few correct answers were seen for this part. Candidates need to realise that titles cannot be output until after the whole array has been searched to establish the highest rating. Then a second time checking the array for this highest rating allows all book titles with this highest rating to be output.

Question 2

(a) (i) Some candidates were able to write a `CASE` statement in their chosen programming language. Many candidates did not convert the pseudocode assignment operator ← to the relevant assignment symbol for their programming language (VB and Python use = and Pascal uses := for this). How a function returns a value is also not well known. Many candidates invented various input and output statements as part of their solution. Candidates need to understand that when converting a flowchart into program code, they need to translate only the steps that are given, not add other steps. Python programmers should use the `IF` … `ELIF` … construct for `CASE` statements.

(ii) Having just attempted to write program code for the flowchart, some candidates realised that `D`, `V` and `I` were normal test data and that there is no borderline test data. The majority of candidates correctly stated that `Y` was invalid input. However, only the stronger responses stated that the expected output would be `-1` in this case.

(b) Many candidates completed the trace table correctly. A common error was to output a value each time round the loop.

(c) (i) The majority of candidates were able to list the features found in the pseudocode: meaningful identifiers, capitalisation of keywords and use of library/built-in functions. Just referring to white space is too vague. A common error was to list indentation as a feature. There were no indented lines in the given pseudocode.

(ii) The requirement of this question was to implement the pseudocode given in **part (b)**. Candidates need to understand that they have to write program code for each line of pseudocode. Many candidates invented additional input and output statements or validation. To answer such questions correctly, candidates need to be able to convert the assignment symbol to the relevant symbol used in their programming language. Candidates also need to be able to convert pseudocode statements such as the `FOR` loop and the `OUTPUT` statement into the keywords used by their programming language. The syllabus requires that candidates know, in their chosen programming language, built-in functions including `LENGTH` and `MID`.

(d) Most candidates correctly stated that the Roman number IIII should return the value `4` and the reason for choosing this number is because it was four identical letters. Many candidates also correctly stated that `IV` would be chosen because it should also give the result `4` but as the shortened Roman number version. Only the stronger responses realised that `XIV` should return `14` and the individual symbols were in order of: higher value, then lower value, then higher value. The final input string `XY` was rarely labelled as invalid data and that it should return an error.

(e) (i)(ii) The stronger responses were able to dry-run the pseudocode successfully with each input string.

(iii) Many candidates correctly stated that the test data showed up a logic error. The stronger responses used the fact that the `NumeralValue` function returns `–1` for invalid characters. They suggested to use an `IF` statement to test for this, give an error message and exit the loop.

(f) (i) Stronger responses were able explain that syntax errors were reported by translator diagnostics during compilation of the program or when running an interpreted program.

(ii) Many vague answers were seen here. Candidates need to understand that logic errors are found during testing when actual results do not match expected results.

Question 3

(a) (i) The stronger responses correctly stated that a breakpoint could be set in program code and that execution would pause at this point. A common error was the misunderstanding that a breakpoint is the part of the program that stops working due to an error.

(ii) Many vague answers were seen here. Very few candidates could explain what stepping was in the context of debugging. Candidates need to have practical experience of using such debugging tools. Stepping is used to execute one statement at a time and program execution pauses after each statement.

(iii) The stronger responses were able to explain that the variable check allows a tester to choose which variables to watch. The variable watch window is then used to see how the variable contents change when stepping through a program.

(b) The majority of candidates were able to state that white box testing makes use of debugging tools.

COMPUTING

Paper 9691/23

Written Paper

Key messages

To succeed in this paper, it is essential that candidates have practical experience of programming using a high-level procedural language. It is recommended that candidates choose one of the following: Pascal, Visual Basic (console mode), or Python. Programming and pseudocode questions from past question papers provide an ideal starting point for the practical work.

General comments

There are a significant number of candidates who do not appear to have any programming experience. Some questions require an answer using a high-level programming language. Although minor discrepancies in syntax are ignored, candidates need to understand that giving pseudocode answers will not gain credit. Some questions ask for pseudocode solutions. Candidates giving answers using a real high-level programming language will be given credit for correct solutions.

Comments on specific questions

Question 1

(a) Many candidates could identify the correct data types but gave incorrect field sizes. A common error was to give a field size for an integer of 1. Candidates need to understand that a data type is of a fixed size. Most programming languages require 4 bytes to store an integer, 8 bytes to store a date, 4 bytes to store single and 8 bytes to store double real numbers, 8 bytes to store currency, 16 bytes to store decimal and 1 or 2 bytes to store a Boolean variable. Candidates need to understand that numeric codes starting with a zero have to be stored as strings.

(b) Most candidates were able to declare a record type in their chosen programming language. There appears to be some confusion between VB6 and VB.NET. Candidates should be aware of the programming language they are using and the keywords that are therefore appropriate.

(c) Many candidates stated that initialisation required to loop 50 times. Fewer candidates were then able to explain that a dummy value needed to be assigned to each field of every record. A common error was to assign NULL or the empty string to the whole array element.

(d) (i) Stronger responses were able to explain that the function `EOF()` returns a Boolean value depending on whether the marker at the end of the file was found at the next read attempt.

(ii) Only the stronger responses appeared to understand the concept of file handling. Many candidates were able to open and close the text file, but were not able to give the steps required to read the file and store the data in the array:

```
OPENFILE CourseData.DAT FOR READING
i ← 1
WHILE NOT EOF(CourseData.DAT)
     READ record from FILE
     Course[i] ← record
     i ← i + 1
ENDWHILE
CLOSEFILE CourseData.DAT
```

(e) Very few candidates recognised that the outer loop could be changed to a conditional loop and even fewer realised that the limit of the inner loop could be reduced to improve efficiency. Some candidates changed the `FOR` loop limits from the maximum of 50 to `NumberOfCourses`.

Question 2

(a) (i) Some candidates were able to write a `CASE` statement in their chosen programming language. Many candidates did not convert the pseudocode assignment operator ← to the relevant assignment symbol for their programming language (VB and Python use = and Pascal uses := for this). How a function returns a value is also not well known. Many candidates invented various input and output statements as part of their solution. Candidates need to understand that when converting a flowchart into program code, they need to translate only the steps that are given, not add other steps. Python programmers should use the `IF` … `ELIF` … construct for `CASE` statements.

(ii) Having just attempted to write program code for the flowchart, some candidates realised that `1` and `X` were invalid test data and the expected result should be `-1`. The majority of candidates correctly stated that G was normal input. A common error was to assume that one of those values must be borderline data.

(b) (i) Many candidates completed the trace table correctly. A common error was to output a value each time round the loop.

(ii) Many candidates correctly identified that line 8 needed correcting. Few candidates could give the correct statement:

```
Denary ← Denary * 10 + ThisNumber
```

(iii) The majority of candidates correctly stated that this was a logic error.

(iv) Many candidates correctly named two other types of error, syntax error and run-time error. The stronger responses were able explain that syntax errors were reported by translator diagnostics during compilation of the program, or when running an interpreted program, and run-time errors were found during execution when the program ‘crashes’ or ‘freezes’.

(c) (i) The majority of candidates were able to list features found in the pseudocode such as meaningful identifiers, capitalisation of keywords, indentation and use of library/built-in functions. Just referring to white space is too vague.

(ii) Most candidates correctly identified comments as a feature that would have been useful to help the understanding of the pseudocode.

(iii) The requirement of this question was to implement the pseudocode given in **part (b)**. Candidates need to understand that they have to write program code for each line of pseudocode. Many candidates invented additional input and output statements or validation. To answer such questions correctly, candidates need to be able to convert the assignment symbol to the relevant symbol used in their programming language. Candidates also need to be able to convert pseudocode statements such as the `FOR` loop and the `OUTPUT` statement into the keywords used by their programming language. The syllabus requires that candidates know, in their chosen programming language, built-in functions including `LENGTH` and `MID`.

(iv) The stronger responses used the fact that the `DenaryDigit` function returns `–1` for invalid characters. They suggested to use an `IF` statement to test for this, give an error message instead of the Denary assignment and exit the loop.

Question 3

(a) (i) The stronger repsonses correctly stated that a breakpoint could be set in program code and that execution would pause at this point. A common error was the misunderstanding that a breakpoint is the part of the program that stops working due to an error.

(ii) Many vague answers were seen here. Very few candidates could explain what stepping was in the context of debugging. Candidates need to have practical experience of using such debugging tools. Stepping is used to execute one statement at a time and program execution pauses after each statement.

(iii) The stronger responses were able to explain that the variable check allows a tester to choose which variables to watch. The variable watch window is then used to see how the variable contents change when stepping through a program.

(b) The majority of candidates were able to state that white box testing makes use of debugging tools.

COMPUTING

Paper 9691/31
Written Paper

General

Candidates' basic understanding of relational database theory is sound but this does not extend to the detail of writing a DML script which requires anything other than a query. **Question 1 part (e)(i)** required a script to add a record to a table and answers seen were weak. Candidates need to be exposed to this in a practical way to foster their understanding.

There were some issues around basic examination technique relating to instructions given on the paper. **Question 1 part (b)(i)** gave the instruction *"Underline the primary key for each table".* The candidate was expected to annotate the three table definitions on the question paper, but this was not done by many candidates. There was a similar issue for **question 5 part (a)(ii)**. There were two instructions *"label the root"* which presented no problem for the candidate. However, the second instruction *"draw a line around the left subtree"* gave a wide range of responses, many of which did not make it clear which nodes formed the left sub-tree.

For **question 4(b)** there was a clear statement *"Five of the six statements below are to be used to complete …"*. Some candidates did not use the bulleted list given and simply made up their own entries to complete the description.

Question 1

Part **(a)** answers were varied. Some correctly stated that the issue was that the design has a repeated group of attributes or gave the detail; that the order date and product ID were repeated for each customer. Some candidates showed their misunderstanding by stating that it was a repeating data value that was the issue (for example product ID 883) and so did not get credit.

Part **(b)(i)** – see the comment made in the general section. Candidates gained one mark for `CustomerID` and `ProductID` for the first two tables. The second mark was for a composite key of `CustomerID` + `OrderDate` for the `ORDER` table, arrived at from the question rubric which stated that *"A customer never places more than one order on any one day"*. A common incorrect answer was a composite primary key of `CustomerID` + `ProductID`.

For part **(ii)**, candidates showed a 1-to-many relationship between `CUSTOMER` and `ORDER` and/or `PRODUCT` and `ORDER`. Alternatively, a many-to-many relationship between `CUSTOMER` and `PRODUCT` gained credit.

Part **(iii)** answers often secured the full two marks, using in the correct manner the terms primary key and a foreign key.

Answers for part **(c)** rarely gained the full three marks. Common errors were the omission of the equality sign for the minimum and maximum boundaries and the inclusion of the $ sign in the condition.

Part **(d)** was well answered by most candidates. Candidates needed to appreciate that within the statements there were two pairs which were each alternatives.

Answers for both parts of **(e)** were weak. See the comment made in the General section for part **(i)**.
This was the first paper on which the term 'referential integrity' had been examined, and weak answers reflected this. Credit was given for candidates who stated that either, if was an attempt to add a record into the `ORDER` table or, to delete a record or change a record in the `PRODUCT` table this could cause problems.

Question 2

The mark scheme allowed for several ways in which the candidates could score the two available marks for part **(a)**. Answers scoring only one mark were common. Many candidates used the term ‘imitates’ instead of 'models', which was not considered worthy of any credit.

Candidates scored better for part **(a)(ii)** with answers suggesting the obvious benefits of using a computer; that it will reduce the time-frame for producing results, it can process large volumes of data, and others.

For part **(b),** most candidates scored the two available marks. The most popular answers were the use of an air pressure and humidity sensor.

Part **(c)** proved challenging to all but the most able candidates. The answer that gained marks was to state that a car design application would require the building of a physical piece of equipment, for example, a model of the car or a wind tunnel. The weather forecasting simulation is an abstraction of the future weather, produced entirely by computer software.

Question 3

Part **(a)(i)** was well answered, but part **(ii)** not as well. Many candidates were often unable to make the connection between hexadecimal and binary and establish that as each hexadecimal digit is encoded with 4 bits. Each of the memory locations shown would then store 16 bits.

For part **(b)(i)**, some candidates lost all four of the available marks by not naming the registers in use. Use of acronyms only was considered insufficient. A common error for the final mark was to misname the Current Instruction Register, the Current Index Register. However, the third step which had been highlighted as a weakness on a previous examination report gave answers which were noticeably better. Candidates were able to express themselves and distinguish between ‘an address’ and the ‘contents of this specified address’.

For part **(b)(ii)**, candidates often secured four of the available five marks for correctly tracing the changes to the four registers. Answers securing the final mark for the use of the address bus and data bus were rare. The address bus is used for the first step (only) and the data bus for the third step (only).

Question 4

For part **(a)**, most candidates were able to apply the four move instructions to the given program.

For part **(b)**, most candidates were able to score three or all of the available marks. Weak answers stated that compiler software was part of the assembly process. Candidates who had experience of some practical programming should have found no problem with this question. In the General section, comments are made about examination technique and the importance of reading the instructions given on the paper before answering the question.

Question 5

Most candidates correctly positioned the four new nodes. Candidates who had made errors in the drawing of the tree for part **(i)** were not then penalised for their answers for parts **(ii)** and **(iii)** as the marking followed through from their tree for part **(i)**. See the comment made in the General section regarding the annotation used to indicate the left sub-tree.

Answers for **(b)(i)** were varied. Common errors were to state that the `RootPtr` variable was an array and that the bounds of the `City` array were 1 to 7 (as there were seven data given for the **(a)(i)** and part **(b)** datasets).

For part **(b)(ii)**, answers that gained the full four marks were rare. Often, the answer showed the correct positioning of the nodes in the `City` array and the `RootPtr` value of 1 but incorrect pointer values.
The algorithm is part **(c)** was well answered. Many candidates were able to secure five or the full six available marks.

Question 6

Many candidates did not perform well on this question. For part **(i)**, most secured the mark for the function name. Errors made for the second mark were the inclusion of the data types for each of the three parameters and the omission of the Flag parameter.
For parts **(iii)**, **(iv)** and **(vi)**, the candidate had to recognise that the function call was not properly formed.
For part **(iii)**, the first two parameters are string values and not integers.
For part **(iv)**, the omission of the third flag parameter is allowed, but the first two parameters again are not integers.
For part **(vi)**, the candidates needed to apply the question rubric "`ThisInteger2` is less than `ThisInteger1`".

Question 7

For part **(a)** many answers secured the mark by stating that the term describes the ability to have more than one program currently loaded in main memory. A large number of answers incorrectly stated that there was more than one program being processed concurrently.

Answers for part **(b)** were weak and few candidates were able to suggest two items of system software. The most popular correct answers were device drivers, spooler software and anti-virus software.

Part **(c)** was well answered and the majority of candidates were able to describe at least two features of batch processing.

For part **(d)(i)**, two key points were looked for. First, that each of the processes is allocated a fixed amount of time (100 milliseconds). Second, that each of the processes in the ready state queue is given its time-slice allocation in sequence. Candidates often secured only one of the available marks.

For part **(ii)**, the candidates were asked to apply this knowledge to a practical scenario for the three processes. The more able candidates were able to secure the full five marks. There were three key points of understanding which had to be applied:

- There is only ever one process in the running state.
- A process may not complete its full time-slice, and …
- When this occurs, the next available process in the ready state starts processing immediately.

The two popular answers for part **(iii)** were that either the process needed to do some input or output, or there was a hardware or software generated interrupt.

Question 8

Candidates had to be resourceful for part **(a)** and add items of hardware to the LAN which would be needed for either the basic operation of the LAN, or to establish a connection with the data loggers. A large number of hardware items which could have scored but most answers did not gain the full four marks. To connect to the Internet, a router, firewall and modem all gained credit. A common error was to show one of these devices, but without any indication of a communications link to the Internet. For the LAN itself, a printer server, file server both scored.

Part **(b)** was well answered with most candidates scoring the full two marks. The most popular answer was optic fibre cabling.

COMPUTING

Paper 9691/32
Written Paper

General Comments

Candidates' basic understanding of relational database theory is sound but this does not extend to the detail of writing a DML script which requires anything other than a query. **Question 1 part (e)(i)** required a script to add a record to a table and answers seen were weak. Candidates need to be exposed to this in a practical way to foster their understanding.

There were some issues around basic examination technique relating to instructions given on the paper. **Question 1** part **(d)(i)** gave the instruction "Underline the primary key of each table above". The candidate was expected to annotate the two table definitions on the question paper, but this was not done by many candidates. Similarly for **question 4** part **(a)(ii)**. There were two instructions "label the root" which presented no problem for the candidate. The second instruction "draw a line around the right subtree" gave a wide range of responses, many of which did not make it clear which nodes formed the right sub-tree.

Question 1

For part **(a)**, most candidates were sufficiently resourceful to name two other attributes which would be in the `MEMBER` table. The member's age did not gain credit.

Part **(b**) answers were varied. One mark was awarded for showing the `MemberID` and `CourseID` (only), and the second mark for indicating the two attributes would form a composite primary key.

For part **(c)**, the answers that stated a table contained a group of repeating attributes was acceptable. Better candidates realised that there was in fact only a single attribute, `Instrument`, which was repeated. A common misunderstanding was that it was the repeating of the value 'saxophone' in the Instrument column which caused the table to not be in 1NF.

For part **(d)(i)**, see the comment made in the General section. For part **(ii)**, 'many-to-one' was the only correct answer, although some candidates wrote a sentence making it clear that the `TUTOR` table was on the 'one side' of the relationship and so gained credit. Most candidates scored for part **(iii)**. Common errors were the misspelling of 'saxophone' or omission of the quotation marks, both of which were penalised.

This was the first paper on which the term 'referential integrity' had been examined, and weak answers reflected this. Credit was given for candidates who stated that either, if was an attempt to add a record into the `TUTOR-INSTRUMENT` table or, to delete a record or change a record in the `TUTOR` table which could cause problems.

Question 2

All three parts of **(a)** were well answered.

For part **(b)**, some candidates lost all four of the available marks when they did not name the registers in use. Use of acronyms only was considered insufficient. A common error for the final mark was to incorrectly name the Current Instruction Register, the Current Index Register.

For part **(c)**, many candidates were able to name a specific control signal. The most common answers seen were an interrupt signal or a signal to indicate the completion of a read/write operation. For the first mark, answers seen were often vague and did not state that either the signal is always from the processor, or to state that individual lines are each dedicated to a particular signal. For part **(c)(ii)**, most candidates were able to name the address bus and data bus. Explanations for the address bus were often vague and did not gain credit. Stating the address bus is 'unidirectional' was considered insufficient. The answer needed to include that the address data it always 'from the processor'.

Question 3

Part **(a)** was well answered by most candidates. For part **(b)**, most candidates were able to score most of the marks.

Question 4

Most candidates correctly positioned the four new nodes for part **(a)(i)**. Candidates who had made errors here were not additionally penalised for their answers for parts **(ii)** and **(iii)** as the marking followed through from their tree shown for part **(i)**. See the comment made in the General section regarding the annotation used to indicate the right sub-tree.

For part **(b)**, answers gaining the full four marks were rare. Often, the answer showed the correct positioning of the nodes in the `FoodName` array and nothing else. Candidates either showed the leaf node pointers as zero or simply left the two pointer columns blank. Because the root data value was stored at array index zero, the showing of null pointers with zero would have been strictly incorrect, although candidates who did this were not penalised.

The algorithm is part **(c)(i)** was well answered. The most common omission was the missing condition from the `UNTIL` line of the pseudocode.

Question 5

The mark scheme allowed for several ways in which the candidates could score the two available marks for part **(a)(i)**. Answers scoring only one mark were common. Many candidates used the term ‘imitates’ instead of 'models' which was not considered creditworthy.

Candidates did better on part **(a)(ii)** with answers suggesting the obvious benefits of using a computer such as it will reduce the time-frame for producing results or that it can process large volumes of data etc..

For parts **(b)** and **(c),** it was clear that students were not aware of the significance of the terms ‘input variable’ and ‘output variable’. Candidates need to appreciate that before any simulation can be attempted, some data will need to be collected (as stated in the rubric of the question). Acceptable answers would include: the number of houses on the new housing development, the percentage of these which own a car, the current traffic flows between Town A and Town B prior to the new road. For part **(c)**, the key aim of the question was to appreciate that certain variables can be changed in the simulation (part **(i)**), which then produces a resulting effect which can be observed (part **(ii)**).

Question 6

Most candidates scored well on this question. Candidates should appreciate that it is the convention on this syllabus to enclose a string datum with double quotation marks and a character value with single quotes.

Contrary to the earlier comment made in the General section, candidates found no problem with following the instruction “label your function header above”. All combinations of answers were seen. Answers showing the three parameters labelled A and the final data type `INTEGER` labelled B were rare. Correct answers for **part (ii)** were rare. The common omission was a correctly formed statement which returned the function value to variable `RejectCount`.

Answers for part **(c)** were better. This question is one where the students will have benefited from the writing and use of functions with practical programming exercises prior to the examination.

Question 7

Most answers for part **(a)** scored either one or both of the available marks. Most candidates stated that the network will be made up of a number of connected computers sited in a small geographical area.

Answers for part **(b)** generally scored well. For part **(i)**, most candidates stated that the computer must have a Network Interface Card. The most popular answer for part **(ii)** was the use of user accounts with a password for authentication. Part **(c)** proved to be more demanding. Candidates had to formulate the specific tasks required of a network operating system. Few candidates were able to suggest two or three tasks. The most popular answers which were creditworthy were:

- The management of a central store of user data files or software.
- The management of all user accounts and log-on to the network.
- The monitoring of the use of the network.

For part **(d)**, the answer expected was a ‘file server’ although just ‘server’ gained credit.

COMPUTING

Paper 9691/33
Written Paper

General

Candidates' basic understanding of relational database theory is sound but this does not extend to the detail of writing a DML script which requires anything other than a query. **Question 1 part (e)(i)** required a script to add a record to a table and answers seen were weak. Candidates need to be exposed to this in a practical way to foster their understanding.

There were some issues around basic examination technique relating to instructions given on the paper. **Question 1 part (b)(i)** gave the instruction *"Underline the primary key for each table".* The candidate was expected to annotate the three table definitions on the question paper, but this was not done by many candidates. There was a similar issue for **question 5 part (a)(ii)**. There were two instructions *"label the root"* which presented no problem for the candidate. However, the second instruction *"draw a line around the left subtree"* gave a wide range of responses, many of which did not make it clear which nodes formed the left sub-tree.

For **question 4(b)** there was a clear statement *"Five of the six statements below are to be used to complete …"*. Some candidates did not use the bulleted list given and simply made up their own entries to complete the description.

Question 1

Part **(a)** answers were varied. Some correctly stated that the issue was that the design has a repeated group of attributes or gave the detail; that the order date and product ID were repeated for each customer. Some candidates showed their misunderstanding by stating that it was a repeating data value that was the issue (for example product ID 883) and so did not get credit.

Part **(b)(i)** – see the comment made in the general section. Candidates gained one mark for `CustomerID` and `ProductID` for the first two tables. The second mark was for a composite key of `CustomerID` + `OrderDate` for the `ORDER` table, arrived at from the question rubric which stated that *"A customer never places more than one order on any one day"*. A common incorrect answer was a composite primary key of `CustomerID` + `ProductID`.

For part **(ii)**, candidates showed a 1-to-many relationship between `CUSTOMER` and `ORDER` and/or `PRODUCT` and `ORDER`. Alternatively, a many-to-many relationship between `CUSTOMER` and `PRODUCT` gained credit.

Part **(iii)** answers often secured the full two marks, using in the correct manner the terms primary key and a foreign key.

Answers for part **(c)** rarely gained the full three marks. Common errors were the omission of the equality sign for the minimum and maximum boundaries and the inclusion of the $ sign in the condition.

Part **(d)** was well answered by most candidates. Candidates needed to appreciate that within the statements there were two pairs which were each alternatives.

Answers for both parts of **(e)** were weak. See the comment made in the General section for part **(i)**.
This was the first paper on which the term 'referential integrity' had been examined, and weak answers reflected this. Credit was given for candidates who stated that either, if was an attempt to add a record into the `ORDER` table or, to delete a record or change a record in the `PRODUCT` table this could cause problems.

Question 2

The mark scheme allowed for several ways in which the candidates could score the two available marks for part **(a)**. Answers scoring only one mark were common. Many candidates used the term ‘imitates’ instead of 'models', which was not considered worthy of any credit.

Candidates scored better for part **(a)(ii)** with answers suggesting the obvious benefits of using a computer; that it will reduce the time-frame for producing results, it can process large volumes of data, and others.

For part **(b),** most candidates scored the two available marks. The most popular answers were the use of an air pressure and humidity sensor.

Part **(c)** proved challenging to all but the most able candidates. The answer that gained marks was to state that a car design application would require the building of a physical piece of equipment, for example, a model of the car or a wind tunnel. The weather forecasting simulation is an abstraction of the future weather, produced entirely by computer software.

Question 3

Part **(a)(i)** was well answered, but part **(ii)** not as well. Many candidates were often unable to make the connection between hexadecimal and binary and establish that as each hexadecimal digit is encoded with 4 bits. Each of the memory locations shown would then store 16 bits.

For part **(b)(i)**, some candidates lost all four of the available marks by not naming the registers in use. Use of acronyms only was considered insufficient. A common error for the final mark was to misname the Current Instruction Register, the Current Index Register. However, the third step which had been highlighted as a weakness on a previous examination report gave answers which were noticeably better. Candidates were able to express themselves and distinguish between ‘an address’ and the ‘contents of this specified address’.

For part **(b)(ii)**, candidates often secured four of the available five marks for correctly tracing the changes to the four registers. Answers securing the final mark for the use of the address bus and data bus were rare. The address bus is used for the first step (only) and the data bus for the third step (only).

Question 4

For part **(a)**, most candidates were able to apply the four move instructions to the given program.

For part **(b)**, most candidates were able to score three or all of the available marks. Weak answers stated that compiler software was part of the assembly process. Candidates who had experience of some practical programming should have found no problem with this question. In the General section, comments are made about examination technique and the importance of reading the instructions given on the paper before answering the question.

Question 5

Most candidates correctly positioned the four new nodes. Candidates who had made errors in the drawing of the tree for part **(i)** were not then penalised for their answers for parts **(ii)** and **(iii)** as the marking followed through from their tree for part **(i)**. See the comment made in the General section regarding the annotation used to indicate the left sub-tree.

Answers for **(b)(i)** were varied. Common errors were to state that the `RootPtr` variable was an array and that the bounds of the `City` array were 1 to 7 (as there were seven data given for the **(a)(i)** and part **(b)** datasets).

For part **(b)(ii)**, answers that gained the full four marks were rare. Often, the answer showed the correct positioning of the nodes in the `City` array and the `RootPtr` value of 1 but incorrect pointer values.
The algorithm is part **(c)** was well answered. Many candidates were able to secure five or the full six available marks.

Question 6

Many candidates did not perform well on this question. For part **(i)**, most secured the mark for the function name. Errors made for the second mark were the inclusion of the data types for each of the three parameters and the omission of the Flag parameter.
For parts **(iii)**, **(iv)** and **(vi)**, the candidate had to recognise that the function call was not properly formed.
For part **(iii)**, the first two parameters are string values and not integers.
For part **(iv)**, the omission of the third flag parameter is allowed, but the first two parameters again are not integers.
For part **(vi)**, the candidates needed to apply the question rubric "`ThisInteger2` is less than `ThisInteger1`".

Question 7

For part **(a)** many answers secured the mark by stating that the term describes the ability to have more than one program currently loaded in main memory. A large number of answers incorrectly stated that there was more than one program being processed concurrently.

Answers for part **(b)** were weak and few candidates were able to suggest two items of system software. The most popular correct answers were device drivers, spooler software and anti-virus software.

Part **(c)** was well answered and the majority of candidates were able to describe at least two features of batch processing.

For part **(d)(i)**, two key points were looked for. First, that each of the processes is allocated a fixed amount of time (100 milliseconds). Second, that each of the processes in the ready state queue is given its time-slice allocation in sequence. Candidates often secured only one of the available marks.

For part **(ii)**, the candidates were asked to apply this knowledge to a practical scenario for the three processes. The more able candidates were able to secure the full five marks. There were three key points of understanding which had to be applied:

- There is only ever one process in the running state.
- A process may not complete its full time-slice, and …
- When this occurs, the next available process in the ready state starts processing immediately.

The two popular answers for part **(iii)** were that either the process needed to do some input or output, or there was a hardware or software generated interrupt.

Question 8

Candidates had to be resourceful for part **(a)** and add items of hardware to the LAN which would be needed for either the basic operation of the LAN, or to establish a connection with the data loggers. A large number of hardware items which could have scored but most answers did not gain the full four marks. To connect to the Internet, a router, firewall and modem all gained credit. A common error was to show one of these devices, but without any indication of a communications link to the Internet. For the LAN itself, a printer server, file server both scored.

Part **(b)** was well answered with most candidates scoring the full two marks. The most popular answer was optic fibre cabling.

COMPUTING

Paper 9691/04

Project 2

General comments

This report provides general feedback on the overall quality of project work for GCE Advanced Level Computing candidates. In addition, all Centres receive specific feedback from their Moderator in the form of a short report that is returned after moderation. This system of reporting provides an ongoing dialogue with Centres, giving valuable pointers to the perceived strengths and weaknesses of the projects moderated.

The projects submitted covered a wide variety of topics, with stronger responses showing evidence of candidates researching a problem beyond their school or college life.

In order to have the full range of marks available to the candidate, the computing project must involve a third party client whose requirements are considered and clearly documented at all stages of the system development. Centres are reminded that the project work is designed to test the candidates' understanding of the systems life cycle. The requirements are clearly set out in syllabus **Section 4: Computing project**. There is a useful checklist for teachers and candidates that sets out the expected contents of each section in **Appendix 8.2 Guidance on marking the Computing Project**.

Centres are also reminded that candidates should use this guidance for the expected contents of their reports, rather than some of the A Level textbooks available on project work, which do not cover the full requirements of the CIE syllabus. Candidates who prepare their work only using these textbooks for guidance and not the syllabus, often miss out vital sections of their reports; or complete unnecessary work, for example, feasibility studies and cost benefit analysis.

Some Centres are still allowing their candidates to produce reports that match the requirements of the previous A Level Computing syllabus. This can mean that the work produced does not meet the requirements of some sub-sections, for example, 'Nature of solution' and 'Systems maintenance documentation'. For other sub-sections, such as 'Installation' and 'Evaluation of the client's and user's response to the system', extra work is included that is not required by the current syllabus.

Project reports and presentation

As usual, the presentation of most of the reports was to a very high standard, with reports word-processed and properly bound. Candidates should ensure that only material essential to the report is included so that they only submit one volume of work. Candidates are reminded that only authentic letters from clients and/or users must be used to provide evidence for the 'Evaluation', 'Installation', 'Investigation and Analysis' sections. These letters could be scanned in to the project report but must not be retyped/typed out by the candidates.

It is strongly recommended that the structure of the candidate's report follows that of the mark scheme set out in the current syllabus. Essential evidence should not be relegated to appendices. This allows teachers at the Centres, and Moderators, to easily check that work for all sections has been included. It is also essential that the pages of the report are clearly numbered by the candidate.

Project assessment and marking

Most Centres used the marking grid on pages 45–52 of the current syllabus to provide a breakdown of marks, showing the marks given for each sub-section of the report. In order to aid the process of moderation, the completed grid should include references to the appropriate pages in the candidate's report where evidence for each section can be found. Teachers should comment as to why they awarded the marks for each section. Moderators have noticed that where there is a good commentary provided by a teacher the marking is usually very close to the agreed standard.

Section 3

Comments on individual sections

The comments set out below identify areas where candidates' work is to be praised, or areas for concern. They are not a guide to the required contents of each section.

(a) Quality of report

Most candidates set out their reports in the appropriate sections and made good use of illustrations, including diagrams and screenshots. Weaker reports sometimes did not include page numbers. This meant that teachers could not clearly identify to the Moderator where evidence was to be found, and those candidates were unable to cross reference items within their report.

(b) Definition, investigation and analysis

(i) Definition – nature of the problem

This is a brief introduction for anyone who is unfamiliar with the organisation and the area under investigation. Most candidates described the organisation and many identified the methods used; better reports described the methods used, the origin of the data and indicated the form of this data.

(ii) Investigation and analysis

In order to gain good marks, candidates must clearly document client and user involvement in their investigation. Candidates need to consider carefully the evidence obtained from interviews, observation of the existing system and study of documents currently in use; then ask follow up questions to fill in any gaps in the knowledge obtained about the current system or requirements needed for the new system. Alternative approaches need to be discussed in depth as they would be applied to the candidate's proposed system.

The detailed requirements specification produced must be based on the information collected and include what the client needs the system to produce. Feasibility studies and cost benefit analysis are not required.

(c) Design

(i) Nature of the solution

The requirements specification set out in the analysis needs to be discussed with the client and a set of measurable objectives agreed. These objectives will then form the basis for the project evaluation.

Most candidates provided designs that included proposed data structures, layouts for input screens and reports required; better reports used pseudocode and/or flowcharts to provide a detailed description of the processes to be implemented.

In order to obtain marks in the top two bands for this sub-section, candidates need to obtain evidence that their client has seen and commented on the design work, and then show what has changed as a result of these comments. Evidence from the solution is not required here. Gantt charts are not required.

(ii) Intended benefits

In order to obtain good marks for this sub-section, candidates should describe the benefits of their intended system, not just provide a list of general statements that could apply to any system.

(iii) Limits of the scope of solution

Candidates should describe the limitations of their intended system including an estimate of the size of any files required, not just provide a list of general statements that could apply to any system. File sizing estimates should be based on information provided by the client.

Full marks for the design section cannot be awarded without candidates clearly supplying evidence for **(i)**, **(ii)** and **(iii)**.

(d) Software development, programming, testing and installation

(i) Development

Evidence of development should include program listings of code written by the candidate, data structures used and evidence of tailoring software packages. For top marks, the solution should have no logical flaws, match the design specification in **(c)(i)** and be annotated by the candidate. If evidence from implementation is included as part of the design then the development cannot be checked against the design specification.

(ii) Programming

It is important that the programming code in this sub-section is written by the candidate and not produced as a result of tailoring a software package. Marks should only be awarded to code that has been written by the candidate. Candidates need to show that they can apply the programming skills developed at AS level in Paper 2 to a real situation. This includes technical programming competence and ensuring that their program could be maintained by writing self-documented code.

(iii) Testing

Evidence of testing needs to be supported by a well-designed test plan that includes the identification of appropriate test data, including valid, invalid and extreme cases, together with expected results for all tests. For top marks to be awarded, the test plan should clearly identify that all parts of the system have been tested. Many candidates only tested the validation and navigation aspects of their system and omitted to test that their system did what it was supposed to do, for example production of reports. This omission meant candidates were unable to gain marks in the highest band for this sub-section.

(iv) Installation

Most candidates provided an implementation plan containing details of user testing, user training and system changeover. For good marks to be awarded, written evidence from the client and/or user(s) must be included in order to show that the system has been seen, used and tested, and the candidate’s plans have been agreed.

Centres are reminded that appropriateness of structure and exploitation of available facilities are not required for this sub-section of the report.

(e) Documentation

(i) Systems maintenance documentation

This sub-section of the report is a Systems Maintenance document. Many candidates incorrectly included Technical Documentation. Please see the current syllabus for details of what should be included in this sub-section. For top marks to be awarded, the candidate must explain how adaptive maintenance could be undertaken for their system.

(ii) User Guide

This section was completed to a good standard by most candidates. Centres are reminded that for full marks the candidate must include an index and a glossary only for the terms used in their User Guide. Glossaries that include items not mentioned in the User Guide would be confusing for anyone reading the guide. For top marks, the User Guide needs to be complete including details of how to install the new system, backup routines and a guide to common errors. Also, good on-screen help should exist where this is a sensible option.

(f) Evaluation

Centres are reminded that in order to gain high marks, candidates need to provide a detailed evaluation that includes the content set out in the guidance for marking projects section of the syllabus.

(i) Discussion of the degree of success in meeting the original objectives

Candidates need to consider each objective set out in **(c)(i)** and explain how their project work met the objective or explain why the objective was not met. Candidates should also indicate where the evidence, probably from testing or feedback from the users of the system, can be found in their report to support these conclusions.

(ii) Evaluation of the client's and users' response to the system

A response must be provided directly from the client and user(s) showing that they have used the system, not just reported by the candidate. The candidate should then evaluate their client's and users' responses.

For evidence in this section to be creditworthy, the candidate must include original letters, preferably on headed notepaper, signed by the client and not typed and/or composed by the candidate.

Centres are reminded that possible extensions and the good and bad points of their final system are not required for this sub-section of the report.