COMPUTER SCIENCE

Paper 9608/11
Written Paper

Key messages

Candidates should use the appropriate technical terminology for precision in answering questions. At this level of study vague, generalised answers will not be given credit.

If there is insufficient space on the question paper for a complete answer, candidates may use additional sheets or blank spaces within the question paper, but the continuation should be clearly indicated. It is also essential, particularly in questions that require a calculation, that candidates indicate in some way the answer which is to be marked if any draft work has been produced.

Some candidates continue to answer in pencil and then overwrite their answers in ink which makes some responses very difficult to read. Candidates' work can only be given credit if it can be read by examiners.

General comments

It is very important that candidates read the question stem carefully and highlight the key words. Some of these key words will indicate the type of answer required, either a single statement or more extended prose, and others will indicate the context in which the question is set. Identifying and understanding these key words will help candidates to give more appropriate answers to questions. For example, **Question 7** required answers in a particular context and generalised responses or responses in a different context are not accepted.

Comments on specific questions

Question 1

(a) The majority of candidates were able to correctly match the terms '*entity*', '*table*' and '*foreign key*', although there was considerable confusion between the definitions for primary, secondary and candidate keys and candidates need to improve their understanding of these terms. Candidates must also ensure they read the question carefully. Quite a few candidates drew more than one line from a description box although the question clearly stated '*Draw **a** line to link...*'

(b) Candidates found this question very challenging. The majority of candidates need to improve their understanding of what is meant by referential integrity. Responses to this question were often either completely inadequate or far too generalised and vague to gain any credit at this level of study. Candidates must be aware of the need to familiarise themselves with the terminology associated with relational databases so that they can explain it appropriately.

Question 2

(a) There were some very good answers to this question, with quite a few candidates able to state two differences between SRAM and DRAM. The majority of candidates seemed to appreciate that SRAM has more complex circuitry, but there was some confusion about what this meant in practice. Incorrect answers such as '*SRAM needs 6 to 8 transistors, while DRAM needs 2 to 4 transistors*' were common. Candidates need to improve their understanding of the construction of these two memory types. It was also not enough to say, for example, '*SRAM is faster than DRAM*'. This was far too vague and imprecise for an answer at this level. There needed to be an indication of what it is that SRAM is faster at doing than DRAM; that is, accessing data.

(b) (i) While **Question 2(b)(ii)** was about the management tasks of an operating system, **Question 2(b)(i)** asked why an operating system is needed. There were a small number of very good answers, but the vast majority of candidates simply repeated the same answers in both parts of the question. Candidates must be aware that there will not be two questions on a paper which require exactly the same answers.

(ii) Most candidates were able to name two management tasks of the Operating System. Some candidates were confused between management tasks of the operating system and utility software and so gave incorrect answers such as disk defragmentation. A significant number of candidates repeated answers given in the previous question.

(c) Many candidates need to improve their understanding of library routines in general and of Dynamic Link Libraries (DLLs) in particular. There were a small number of very good answers, but the vast majority of candidates gave answers that referred to library files in general and not specifically to DLL files. A small number of candidates confused Dynamic Link Library (DLL) with Data Definition Language (DDL).

Question 3

(a) (i) This question was very well answered. The most common cause of error was the omission of the leading zeros when the question asked for conversion to 8-bit binary.

(ii) This question was also very well answered. However, a common error by candidates using the method of 'flipping the bits' was to forget to then add one, so a common incorrect answer was 1101 0001.

(iii) This was also very well answered. Most of the candidates correctly converted the denary number to hexadecimal.

(b) (i) Most of the candidates were able to explain how to convert a denary number of more than one digit to Binary Coded Decimal (BCD) and there were some excellent examples given. A small number of candidates confused BCD with hexadecimal and explained how to convert denary numbers to hexadecimal.

(ii) This question was less well answered. Most candidates knew the principles involved, but when illustrating with an example many candidates used 4-bit binary numbers greater than denary 9 in their 8-bit BCD representation and so gave incorrect conversions. For example, 8-bit BCD representation 0010 1110 was typically used as an illustration and given as 214 in denary.

Question 4

This was another question that candidates found very challenging. The question asked for technical descriptions of the internal operation of the devices but many candidates described in very general terms what the devices were used for, which did not answer the question.

(i) There were a few excellent answers to this question and some candidates clearly understood how a keyboard operates, but many candidates need to improve their understanding in this area. Candidates seemed to have little appreciation of the role of the key matrix and there was a general misconception that ASCII codes were used to transfer characters.

(ii) There were a small number of excellent responses, but many candidates need to improve their understanding of the operation of optical drives. It was not enough at this level of study to say simply, '*an optical disc uses light*'. This was far too imprecise for credit.

(iii) As in **Question 4(ii)**, answers such as '*an optical mouse uses light*' were far too generalised for credit. Candidates should show understanding of the operation of optical mice. Several answers included references to the mouse ball, and incorrect statements such as *'...the laser detects...*' were common, rather than the sensor detecting the reflection (of the laser) from the surface.

(iv) As in **Question 4(ii)** above, answers such as, '*a scanner uses (laser) light*', were far too generalised for credit. There needed to be some explanation of why the light is used or what it is used for. Many candidates need to improve their understanding of the internal operation of a scanner. A common misconception was that a scanned image comprised a series of photographs stitched together.

Question 5

(i) There were many excellent answers to this part question, with the majority of candidates able to draw a correct circuit. Candidates should be aware that it is important to use the correct symbols for each logic gate. It was sometimes difficult to differentiate between the symbols for AND and OR gates and frequently the circle was missing on the symbol for a NOT gate.

(ii) The majority of candidates were able to give an appropriate logic statement corresponding to the circuit in **part (i)**. Many candidates need to improve their understanding of what is meant by a logic statement.

(iii) This part question was also well answered, with many candidates able to give a completely correct truth table.

Question 6

Many candidates found this question challenging. These candidates need to improve their understanding in this area. On-demand bit streaming was frequently confused with real-time bit streaming. Frequently, candidates who realised that a buffer would be required used the terms '*buffer*' or '*buffering*' without any clear explanation of where the buffer was located or why it was needed. There were a few very good responses, with a small number of candidates able to explain the process very clearly.

Question 7

(a) The majority of candidates were able to name three utility programs, although there was some confusion between utilities and application software. There were also some good descriptions of the purpose of the various programs but sometimes the descriptions were too vague and imprecise to be given credit at this level. It was not enough, for example, to say '*compression software compresses files*'. An explanation of what compression does to the file was required. Candidates especially need to improve their understanding of how defragmentation software works. Care must be taken too to write the name of the software correctly. Quite a few candidates wrote '*disk fragmentation software*' rather than '*disc defragmentation software*'.

(b) This question was generally answered well; the majority of candidates could give four differences between bitmap and vector graphics. Candidates should be aware that it is not enough to give both aspects of the difference as two separate points; for example, the two statements '*bitmap files are usually bigger than vector graphics files*' and '*vector graphics files are usually smaller than bitmap files*' were, in effect, the same answer and so were only given credit once. Answers also needed to be precise, for example, '*bitmaps are larger in size*' was not enough; there needed to be some clarification of how they are larger in size and whether the answer refers to file size or dimensions.

(c) (i) Most candidates seemed have a general understanding that encryption would not prevent access to the company's computers, but many candidates need to improve their understanding of what encryption actually does. Encryption does not prevent access to the data, nor does it make the data unreadable. A hacker could still read the data, what encryption does is to ensure that if a hacker does read the data it will be incomprehensible.

(ii) This question was very well answered with the vast majority of candidates understanding that verification was being described and that validation ensures that data meets certain criteria. There was some confusion between validation and data integrity, and candidates needed to ensure that they use the precise terminology. '*Validation ensures that data is valid*' was not exact enough for credit at this level of study.

(iii) Many candidates understood that a password was only useful if it was kept secure, and that there were many ways in which passwords could be misappropriated. Again candidates need to be careful with terms used in answers. Just '*passwords can be hacked*' was not a sufficient answer at this level. There needed to be some explanation of **how** a password can be hacked.

Question 8

(a) (i) Almost all candidates were able to correctly write the contents of the accumulator after the execution of the given instruction.

(ii) There were some very good answers to this question, with many candidates entering the correct value into the accumulator and explaining clearly how they had carried out the operation. There is still, however, considerable confusion amongst some candidates with regard to indexed addressing. Nearly every incorrect answer to this question had the value 0011 0001 in the accumulator. This value was obtained by incorrectly adding the contents of the index register to the contents of the given address rather than by correctly adding the contents of the index register to the actual address given in the instruction. There were many good explanations of the method used, although candidates need to be aware of the need to carefully state each step in the process.

(b) (i) Many candidates successfully traced the first four lines of the code, although some candidates incorrectly changed the value in the accumulator to zero when storing the value in address 802. However, when executing line 104 quite a number of candidates used the original value of zero from address 802 instead of the updated value of 90. This meant that the result of the comparison was false and execution was incorrectly transferred to line 110 instead of line 107. Candidates must take care on questions of this type and check carefully on the column headings in the trace table. In this case there was not a column headed 'Address'. The first column was headed ACC. A small number of candidates incorrectly put the address values in the ACC column and then had nowhere to write the values that should have been in the accumulator.

(ii) The majority of the candidates who correctly used the updated value of the contents of address 802 in **part (b)(i)** realised that the statement at line 107 was the one that was not needed. Candidates must take care to read the question carefully. Some gave the instruction as the answer when the question clearly says '*state the address of this instruction*'.

(c) (i) There were some excellent answers to this question, with many candidates able to fully describe the limitations of ASCII code. A few candidates confused the word '*disadvantage*' with the word '*advantage*' and gave reasons why ASCII would be preferable. Some candidates needed to understand that when the numbers 0 to 127 or 0 to 255 are used to represent characters, there are actually 128 or 256 different representations.

(ii) There were many very good answers and the majority of candidates clearly understood the ways in which Unicode overcomes the limitations of ASCII.

COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

Candidates should use the appropriate technical terminology for precision in answering questions. At this level of study vague, generalised answers will not be given credit.

If there is insufficient space on the question paper for a complete answer, candidates may use additional sheets or blank spaces within the question paper, but the continuation should be clearly indicated. It is also essential, particularly in questions that require a calculation, that candidates indicate in some way the answer which is to be marked if any draft work has been produced.

In one question candidates were asked to write SQL statements. The best way to prepare candidates for questions on this topic is by exposing them to some practical work using simple databases which they can develop by writing straightforward SQL scripts. If suitable software is not available there are a number of excellent online resources that could be used.

Some candidates continue to answer in pencil and then overwrite their answers in ink which makes some responses very difficult to read. Candidates' work can only be given credit if it can be read by examiners.

General comments

It is very important that candidates read the question stem carefully and highlight the key words. Some of these key words will indicate the type of answer required, either a single statement or more extended prose, and others will indicate the context in which the question is set. Identifying and understanding these key words will help candidates to give more appropriate answers to questions. For example, **Question 2(a)**, and **Question 9(a)** required answers or examples in a particular context. Generalised responses or responses in a different context are not accepted.

Comments on specific questions

Question 1

(a) There were many excellent answers to this part question, with the majority of candidates able to draw a correct circuit. Some candidates recognised that (NOT B) AND (NOT S) could be replaced with a single gate, but most of the candidates who made the replacement incorrectly, used a NAND gate instead of a NOR gate. Candidates should be aware that it is important to use the correct symbols for each logic gate. It was sometimes difficult to differentiate between the symbols for AND and OR gates. On many occasions, the circle was missing on the symbol for a NOT gate.

(b) This part question was also well answered, with many candidates able to give a completely correct truth table.

Question 2

Many candidates found this question challenging. Candidates need to improve their understanding of the processes associated with video recording and playing, so that methods can be correctly described. Responses often did not answer the question or were far too generalised and vague to gain any credit at this level of study. Candidates must be aware of the need to familiarise themselves with the technical terminology of this subject so that they can use it appropriately.

(a) Candidates needed to read the question carefully. Whilst a significant number of candidates understood what was meant by '*frame rate*' many did not read the question properly and gave answers referring to *displaying* a video, when the question clearly stated '*When recording a video*'.

(b) A small number of candidates gave very full correct descriptions of Interlaced and Progressive Encoding, but there was considerable confusion between two encoding methods and on-demand and real-time bit streaming, and many candidates described those techniques instead. Even when candidates had some understanding of the methods, answers were often far too imprecise to be credited at this level of study. It is not enough to say, for example, '*Interlaced encoding is old technology*'. Candidates needed to describe what the technology was (and still is) used for. Similarly, it was not enough to say that '*a frame is divided into an odd field and an even field*', there needed to be a description of what constitutes an odd field or an even field. A significant number of candidates did understand the different bandwidth requirements of the two methods.

(c) (i) Only a minority of candidates were able to identify the correct names for the terms described. The most common incorrect answer was putting the terms 'Interlaced Encoding' and 'Progressive Encoding' in the spaces provided. Some candidates, who knew the names of the two terms, put the term next to the wrong definition. A small number of candidates correctly put the first word but omitted the word 'redundancy'.

(ii) A significant number of candidates correctly identified the technique as file compression, but others missed the words '*file technique*' in the question, and so the most common incorrect answer was '*run-length encoding*'.

Question 3

There were many good answers to this question, but only a small number of completely correct sequences. The vast majority of candidates correctly identified step 2 as the statement labelled G in the table. The most common errors were the reversal of steps 4 and 5 with some candidates writing DF instead of FD. In other words, the head was moving to the correct track (D) before the track had been identified (F) or the reversal of steps 5 and 6, waiting for the correct sector to arrive (H) before the head was over the correct track (D). Quite a few candidates also reversed the last two steps so that the contents of the buffer were incorrectly transferred to memory (E) before the interrupt had been generated (A).

Question 4

There were many excellent answers to this question, and the majority of candidates were able to successfully complete the required conversions in order to join up the correct answer to each left hand box. Some candidates need to improve their understanding of Binary Coded Decimal (BCD) representation. One of the most commonly occurring incorrect answers was to join 0100 1001 in BCD to 73, the straightforward binary to decimal conversion. Other candidates need to improve their understanding of the two's complement representation of negative numbers. There was some confusion with sign and magnitude representation, so another common incorrect answer was to join the two's complement binary integer 11000 001 to – 65, rather than – 63.

Question 5

(a) (i) This question was very well answered, and almost all candidates were able to correctly write the contents of the accumulator after the execution of the given instruction.

(ii) There were also some very good answers to this question, with many candidates entering the correct value into the accumulator and explaining clearly how they had carried out the operation. There is still considerable confusion amongst some candidates about indexed addressing. Nearly every incorrect answer to this question had the value 1001 0111 in the accumulator. This value was obtained by incorrectly adding the contents of the index register to the contents of the given address rather than correctly adding the contents of the index register to the actual address given in the instruction.

(iii) The majority of candidates understood what was meant by indirect addressing and gave the correct contents of the accumulator. There were also many good explanations of the method used, but candidates need to be aware of the need to carefully state each step in the process.

(b) Many candidates successfully traced the first three lines of the code, but some candidates incorrectly changed the value in the accumulator to zero when storing the values in addresses 812 and 813. The instruction at line 803 was also generally understood and completed correctly. When executing line 804, a significant number of candidates used the original value of zero from address 812 instead of the updated value of 29. This meant that the final value in the accumulator and the value stored in address 813 were then both incorrect. Candidates must take care on questions of this type and check carefully on the column headings in the trace table. In this case, there was not a column headed 'Address'. The first column was headed ACC. A small number of candidates incorrectly put the address values in the ACC column and then had nowhere to write the values that should have been in the accumulator.

Question 6

(a) There were many excellent answers to this question, although candidates must be aware that when the question says '*describe two differences between RAM and ROM*' both RAM and ROM must be mentioned in each answer in order to gain credit. It was not enough to say, for example, '*ROM is non-volatile*' for one answer and then '*RAM is volatile*' for the second difference. There was some confusion about which was the volatile memory, with a small number of candidates giving the answer the wrong way round. Candidates also need to take care when describing how data can be accessed from each type of memory. Some candidates answered '*ROM can only read data*' which suggested that it is the ROM that is doing the reading. '*Data can only be read from ROM*' is a much clearer statement.

(b) There were some good answers to this question, with quite a few candidates able to state three differences between SRAM and DRAM. The majority of candidates seemed to appreciate that SRAM has more complex circuitry, but there was some confusion about what this meant in practice. Incorrect answers such as '*SRAM needs 6 to 8 transistors, while DRAM needs 2 to 4 transistors*' were common. Candidates need to improve their understanding of the construction of these two memory types. It was also not enough to say, for example, '*SRAM is faster than DRAM*'. This was far too vague and imprecise for an answer at this level. There needed to be an indication of what it is that SRAM is faster at doing than DRAM; that is, accessing data.

Question 7

This question was generally answered well, with most candidates able to name two management tasks of the Operating System, but fewer were able to then describe the operations included in that task. This is another example where the question needs to be carefully read. A significant number of candidates included the provision of a user interface as one of the tasks when it was given in the question and the question clearly states '*Describe two more of these tasks*'. There was some confusion between management tasks of the operating system and utility software and incorrect answers such as disk defragmentation were not uncommon.

Question 8

Many candidates need to improve their understanding of library routines in general and of Dynamic Link Libraries (DLLs) in particular.

(a) A small number of candidates understood the benefits to a programmer of using a library routine in a program. There was considerable confusion between using routines from a library and the writing of sub-routines. Many candidates described the benefits of using sub-routines, rather than the benefits of library routines.

(b) (i) While **Question 8(a**) was about library routines in general, **Question 8(b)** was specifically about DLL files. There were a small number of very good answers, but the vast majority of candidates simply repeated the answers given in **part (a)**. Candidates must be aware that there will not be two questions on a question paper which require the same answers. A small number of candidates confused Dynamic Link Library (DLL) with Data Definition Language (DDL).

(ii) There were a small number of correct answers to this question but many candidates did not attempt an answer.

Question 9

Candidates found some parts of this question challenging. Many candidates need to improve their understanding of relational databases and associated topics.

(a) Candidates should be aware that this question was set in the context of a Health Club, and while some of the reasons why a programmer would use a relational database were clearly generic, examples used to illustrate the reasons should have been given in context. Many responses were far too imprecise for credit at this level. It is not enough, for example, to say '*databases are secure*'. There needed to be an explanation of why databases can be more secure than a file-based system. There was also a general misconception that the file system would be paper based, so answers about saving space or losing and tearing records were common.

(b) Candidates did not find this part of the question as difficult and there were more correct answers. However, candidates should be aware of the need to use a recognised convention for showing the degree of a relationship on an E-R diagram, for example, 'crows' feet' or 1 and ∞. A single arrow on a line is not an acceptable convention for degree of relationship. In questions of this type candidates must be particularly careful to completely erase or cross out any lines which have been corrected or changed. Credit cannot be given if the answer is not clear.

(c) A small number of candidates provided completely correct answers. Candidates need to improve their understanding of what is meant by an SQL script. There were answers written in a variety of programming languages. There was considerable confusion about the syntax and use of the CREATE TABLE statement, with quite a few candidates attempting to create an empty table and then use the ALTER TABLE statement to insert the fields. Candidates also need to improve their understanding of data types applicable to SQL, rather than trying to use data types from programming languages which are not appropriate.

COMPUTER SCIENCE

Paper 9608/13
Written Paper

Key messages

Candidates should use the appropriate technical terminology for precision in answering questions. At this level of study vague, generalised answers will not be given credit.

If there is insufficient space on the question paper for a complete answer, candidates may use additional sheets or blank spaces within the question paper, but the continuation should be clearly indicated. It is also essential, particularly in questions that require a calculation, that candidates indicate in some way the answer which is to be marked if any draft work has been produced.

Some candidates continue to answer in pencil and then overwrite their answers in ink which makes some responses very difficult to read. Candidates' work can only be given credit if it can be read by examiners.

General comments

It is very important that candidates read the question stem carefully and highlight the key words. Some of these key words will indicate the type of answer required, either a single statement or more extended prose, and others will indicate the context in which the question is set. Identifying and understanding these key words will help candidates to give more appropriate answers to questions. For example, **Question 7** required answers in a particular context and generalised responses or responses in a different context are not accepted.

Comments on specific questions

Question 1

(a) The majority of candidates were able to correctly match the terms '*entity*', '*table*' and '*foreign key*', although there was considerable confusion between the definitions for primary, secondary and candidate keys and candidates need to improve their understanding of these terms. Candidates must also ensure they read the question carefully. Quite a few candidates drew more than one line from a description box although the question clearly stated '*Draw **a** line to link...*'

(b) Candidates found this question very challenging. The majority of candidates need to improve their understanding of what is meant by referential integrity. Responses to this question were often either completely inadequate or far too generalised and vague to gain any credit at this level of study. Candidates must be aware of the need to familiarise themselves with the terminology associated with relational databases so that they can explain it appropriately.

Question 2

(a) There were some very good answers to this question, with quite a few candidates able to state two differences between SRAM and DRAM. The majority of candidates seemed to appreciate that SRAM has more complex circuitry, but there was some confusion about what this meant in practice. Incorrect answers such as '*SRAM needs 6 to 8 transistors, while DRAM needs 2 to 4 transistors*' were common. Candidates need to improve their understanding of the construction of these two memory types. It was also not enough to say, for example, '*SRAM is faster than DRAM*'. This was far too vague and imprecise for an answer at this level. There needed to be an indication of what it is that SRAM is faster at doing than DRAM; that is, accessing data.

(b) (i) While **Question 2(b)(ii)** was about the management tasks of an operating system, **Question 2(b)(i)** asked why an operating system is needed. There were a small number of very good answers, but the vast majority of candidates simply repeated the same answers in both parts of the question. Candidates must be aware that there will not be two questions on a paper which require exactly the same answers.

(ii) Most candidates were able to name two management tasks of the Operating System. Some candidates were confused between management tasks of the operating system and utility software and so gave incorrect answers such as disk defragmentation. A significant number of candidates repeated answers given in the previous question.

(c) Many candidates need to improve their understanding of library routines in general and of Dynamic Link Libraries (DLLs) in particular. There were a small number of very good answers, but the vast majority of candidates gave answers that referred to library files in general and not specifically to DLL files. A small number of candidates confused Dynamic Link Library (DLL) with Data Definition Language (DDL).

Question 3

(a) (i) This question was very well answered. The most common cause of error was the omission of the leading zeros when the question asked for conversion to 8-bit binary.

(ii) This question was also very well answered. However, a common error by candidates using the method of 'flipping the bits' was to forget to then add one, so a common incorrect answer was 1101 0001.

(iii) This was also very well answered. Most of the candidates correctly converted the denary number to hexadecimal.

(b) (i) Most of the candidates were able to explain how to convert a denary number of more than one digit to Binary Coded Decimal (BCD) and there were some excellent examples given. A small number of candidates confused BCD with hexadecimal and explained how to convert denary numbers to hexadecimal.

(ii) This question was less well answered. Most candidates knew the principles involved, but when illustrating with an example many candidates used 4-bit binary numbers greater than denary 9 in their 8-bit BCD representation and so gave incorrect conversions. For example, 8-bit BCD representation 0010 1110 was typically used as an illustration and given as 214 in denary.

Question 4

This was another question that candidates found very challenging. The question asked for technical descriptions of the internal operation of the devices but many candidates described in very general terms what the devices were used for, which did not answer the question.

(i) There were a few excellent answers to this question and some candidates clearly understood how a keyboard operates, but many candidates need to improve their understanding in this area. Candidates seemed to have little appreciation of the role of the key matrix and there was a general misconception that ASCII codes were used to transfer characters.

(ii) There were a small number of excellent responses, but many candidates need to improve their understanding of the operation of optical drives. It was not enough at this level of study to say simply, '*an optical disc uses light*'. This was far too imprecise for credit.

(iii) As in **Question 4(ii)**, answers such as '*an optical mouse uses light*' were far too generalised for credit. Candidates should show understanding of the operation of optical mice. Several answers included references to the mouse ball, and incorrect statements such as *'...the laser detects...*' were common, rather than the sensor detecting the reflection (of the laser) from the surface.

(iv) As in **Question 4(ii)** above, answers such as, '*a scanner uses (laser) light*', were far too generalised for credit. There needed to be some explanation of why the light is used or what it is used for. Many candidates need to improve their understanding of the internal operation of a scanner. A common misconception was that a scanned image comprised a series of photographs stitched together.

Question 5

(i) There were many excellent answers to this part question, with the majority of candidates able to draw a correct circuit. Candidates should be aware that it is important to use the correct symbols for each logic gate. It was sometimes difficult to differentiate between the symbols for AND and OR gates and frequently the circle was missing on the symbol for a NOT gate.

(ii) The majority of candidates were able to give an appropriate logic statement corresponding to the circuit in **part (i)**. Many candidates need to improve their understanding of what is meant by a logic statement.

(iii) This part question was also well answered, with many candidates able to give a completely correct truth table.

Question 6

Many candidates found this question challenging. These candidates need to improve their understanding in this area. On-demand bit streaming was frequently confused with real-time bit streaming. Frequently, candidates who realised that a buffer would be required used the terms '*buffer*' or '*buffering*' without any clear explanation of where the buffer was located or why it was needed. There were a few very good responses, with a small number of candidates able to explain the process very clearly.

Question 7

(a) The majority of candidates were able to name three utility programs, although there was some confusion between utilities and application software. There were also some good descriptions of the purpose of the various programs but sometimes the descriptions were too vague and imprecise to be given credit at this level. It was not enough, for example, to say '*compression software compresses files*'. An explanation of what compression does to the file was required. Candidates especially need to improve their understanding of how defragmentation software works. Care must be taken too to write the name of the software correctly. Quite a few candidates wrote '*disk fragmentation software*' rather than '*disc defragmentation software*'.

(b) This question was generally answered well; the majority of candidates could give four differences between bitmap and vector graphics. Candidates should be aware that it is not enough to give both aspects of the difference as two separate points; for example, the two statements '*bitmap files are usually bigger than vector graphics files*' and '*vector graphics files are usually smaller than bitmap files*' were, in effect, the same answer and so were only given credit once. Answers also needed to be precise, for example, '*bitmaps are larger in size*' was not enough; there needed to be some clarification of how they are larger in size and whether the answer refers to file size or dimensions.

(c) (i) Most candidates seemed have a general understanding that encryption would not prevent access to the company's computers, but many candidates need to improve their understanding of what encryption actually does. Encryption does not prevent access to the data, nor does it make the data unreadable. A hacker could still read the data, what encryption does is to ensure that if a hacker does read the data it will be incomprehensible.

(ii) This question was very well answered with the vast majority of candidates understanding that verification was being described and that validation ensures that data meets certain criteria. There was some confusion between validation and data integrity, and candidates needed to ensure that they use the precise terminology. '*Validation ensures that data is valid*' was not exact enough for credit at this level of study.

(iii) Many candidates understood that a password was only useful if it was kept secure, and that there were many ways in which passwords could be misappropriated. Again candidates need to be careful with terms used in answers. Just '*passwords can be hacked*' was not a sufficient answer at this level. There needed to be some explanation of **how** a password can be hacked.

Question 8

(a) (i) Almost all candidates were able to correctly write the contents of the accumulator after the execution of the given instruction.

(ii) There were some very good answers to this question, with many candidates entering the correct value into the accumulator and explaining clearly how they had carried out the operation. There is still, however, considerable confusion amongst some candidates with regard to indexed addressing. Nearly every incorrect answer to this question had the value 0011 0001 in the accumulator. This value was obtained by incorrectly adding the contents of the index register to the contents of the given address rather than by correctly adding the contents of the index register to the actual address given in the instruction. There were many good explanations of the method used, although candidates need to be aware of the need to carefully state each step in the process.

(b) (i) Many candidates successfully traced the first four lines of the code, although some candidates incorrectly changed the value in the accumulator to zero when storing the value in address 802. However, when executing line 104 quite a number of candidates used the original value of zero from address 802 instead of the updated value of 90. This meant that the result of the comparison was false and execution was incorrectly transferred to line 110 instead of line 107. Candidates must take care on questions of this type and check carefully on the column headings in the trace table. In this case there was not a column headed 'Address'. The first column was headed ACC. A small number of candidates incorrectly put the address values in the ACC column and then had nowhere to write the values that should have been in the accumulator.

(ii) The majority of the candidates who correctly used the updated value of the contents of address 802 in **part (b)(i)** realised that the statement at line 107 was the one that was not needed. Candidates must take care to read the question carefully. Some gave the instruction as the answer when the question clearly says '*state the address of this instruction*'.

(c) (i) There were some excellent answers to this question, with many candidates able to fully describe the limitations of ASCII code. A few candidates confused the word '*disadvantage*' with the word '*advantage*' and gave reasons why ASCII would be preferable. Some candidates needed to understand that when the numbers 0 to 127 or 0 to 255 are used to represent characters, there are actually 128 or 256 different representations.

(ii) There were many very good answers and the majority of candidates clearly understood the ways in which Unicode overcomes the limitations of ASCII.

COMPUTER SCIENCE

Paper 9608/21
Written Paper

Key messages

In preparation for this examination, candidates were expected to have previously studied the pre-release material sent to Centres. This material included a range of tasks designed to help candidates practise their problem-solving and programming skills. This information, combined with past papers, gives a clear indication of the types of question that candidates can expect.

There were some excellent answers to the programming questions, but a significant number of candidates displayed low programming ability. Candidates need extensive practical programming experience before they sit this examination.

Many candidates appeared to be confused over the requirements of certain questions.
Candidates need to appreciate the importance of good examination technique and particularly the need to fully read and understand each question before they attempt to answer it.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings, and it is important that these are used correctly. It is also important for candidates to use the correct syntax when writing or explaining algorithms using pseudocode. Candidates particularly need to appreciate when it is appropriate to use the assignment operator '←' as instead of the '=' symbol.
It is also important for candidates to understand the use quotation marks to differentiate between an identifier name and the value of a character or string.

General comments

If a candidate writes the answer to a question on an additional page, they must indicate very clearly where their revised answer is to be found.

If answers have been crossed out, the new answers must be written very clearly so that the text can be read easily and candidates can be rewarded with the correct mark.

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

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

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

Candidates need to be clear if they are being asked to write pseudocode or program when reading a question. Some candidates incorrectly stated '*pseudocode*' as their programming language.

There is an increasing tendency for candidates to use incorrect sections of code in their answers to the programming questions. A particular example of this is the use of file-handling statements in a programming question where no file access is required.

Several questions ask the candidate to '*State*' or *Explain*' something. Centres need to emphasise that for these questions, it is not enough simply to repeat the words or phrases from the question itself. An example of this is the question that asks to differentiate between user-defined and built-in functions; in this case, it is not enough to simply state that 'a user-defined function is defined by the user' as this does not demonstrate any understanding.

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

Comments on specific questions

Question 1

(a) This was generally well answered. The majority of candidates scored at least four marks. A common mistake was to omit the 'YES' and 'NO' labels from the outputs of the selection symbols.

Weaker candidates tended to gain two marks by labelling the two `INPUT` / `OUTPUT` symbols.

(b) A wide range of marks was awarded, but the question was only answered well by a minority of candidates. Marks were often lost due to inadequate explanations. A common mistake was to give a positive numeric value for `BaggageWeight` and to then explain that this was '*erroneous*' or '*normal*'. These terms were not appropriate to the scenario. Explanations needed to clearly refer to the baggage allowance.

Mistakes in the 'Expected output' included:

- not outputting '0' (zero) in the case of no excess charge
- incorrect calculations
- giving an expression (such as '3 * 3.50') rather than evaluating the excess charge.

(c) This question was not well answered. Only around 40% of candidates gave the correct pre-condition loop construct, and only a small number of candidates gained all three marks.

The question asked for a pre-condition loop, so a `WHILE ... ENDWHILE` construct was required. Many candidates gave a solution based on `REPEAT ... UNTIL`. Few candidates who used the correct loop construct realised that an additional `INPUT` statement was required before the loop.

The condition was often incorrect. Common mistakes included incorrect logic and missing quotation marks around the characters 'E' and 'S'.

Many candidates simply copied the text from the flowchart, giving:

```
WHILE TicketType <> E OR S
```

This was incorrect both logically and syntactically.

Question 2

(a) This question attracted a range of responses. A small number of students gave completely correct answers, but there were also a significant number of candidates who gained no marks. A number of candidates made no attempt at this question. Trace tables are important tools for debugging and as such are likely to feature in some form or other in many papers.

With reference to the published mark scheme, the marks most often awarded were for those at the start of the table; with perhaps the most commonly awarded mark being given for '`5 Error – investigate'`.

A number of candidates made a good start, but then made mistakes after the first set of six iterations.

(b) Many candidates correctly identified the different programming constructs.
Incorrect answers were normally limited to the 'selection' and 'iteration' parts of the question.

Question 3

(a)(i) and (ii)
These two parts were very well answered with the majority providing a correct answer.

(a)(iii) and (iv)
A minority of candidates answered these two parts correctly. Many candidates omitted any form of quotation marks.

(b) (i) There was a mixed response for this question.

Common mistakes included:

- enclosing the variable `Surname` in quotes (the problem of the difference between a string and an identifier)
- giving '`i`' as both the second **and** third parameter of the `SUBSTR()` function
- using a non-pseudocode function.

(ii) This was a straightforward exercise in translating an algorithm from pseudocode into a high level language. Many candidates gained full marks.

Common mistakes included:

- declaring `NextChar` as a `STRING` or `INTEGER`
- using a pseudocode function rather than one from the chosen high level language
- terminating the loop incorrectly.

As mentioned in the general comments section, candidates who give answers in Python must learn that indentation is an intrinsic part of the syntax and must be clearly shown.

(c) (i) This part was answered well by a minority of candidates. Candidates appeared to have most students have had little experience of writing modular code.

A significant number of candidates offered the name of a programming language, but nothing else.

In addition to general errors of syntax, common mistakes included:

- In VB, use of `Dim` in place of `Function`.
- incorrect parameter types

(ii) This part was not answered well. Many candidates offered no answer at all. A fairly common mistake was to attempt to define a modified function header. Several responses involved returning a value other than the required one.

(iii) This part was not answered well. There were many variations on the required answer, with a number having the variable `ThisID` appearing on the right-hand side of the assignment. Of those answers that were generally of the correct format, the most common mistake was to not have the surname "Wilkes" enclosed by quotation marks. As mentioned in the key messages section, many candidates did not seem clear about the difference between a literal string and an identifier name.

Some candidates seemed confused between the use of a procedure and a function.

(d) (i) Candidates generally gained one mark for this question. In many cases, the use of vague terminology caused them to fall short of the second mark. For example, the phrase 'already there' is not acceptable in place of 'part of the programming language'.

(ii) Many candidates correctly referred to the functions both needing to be called and both returning a value, but several candidates mistakenly used the term `OUTPUT` rather than `RETURN`.

A number of candidates referred to both types of function having a 'similar format', which was too vague.

Question 4

(a) A significant number of candidates gave completely incorrect answers, indicating a lack of understanding of the terms in the question. Many responses had the functions completely reversed, for example, suggesting that the compiler turned the executable code into source code.

A number of candidates did give an acceptable alternative to the word 'translate'.

(b) (i) The majority of candidates correctly suggested that 'errors are highlighted' in some form or another.

Several candidates offered answers that were little more than the question re-worded.

A significant number of candidates gave no answer, and there were many one word answers that were insufficient at best, for example, 'compiler'.

(ii) There were only a small number of correct answers.

As for the previous question, many candidates gave no answer, and there were many one word answers that were insufficient. For example 'Debugging'

(c) (i) This question had four main mark points; with a fifth being available for candidates who realised that type conversion would be required before a string variable could be stored in the `PRetailPrice` array of type `REAL`. This mark was given in a small number of cases, and it was encouraging to see that some candidates did recognise this requirement.

Many candidates gained no marks for this question. Common mistakes included:

- Not including the file name in quotation marks in the `OPEN` statement
- Not specifying the read mode (or specifying `WRITE` mode)
- Incorrect `EOF()` check
- Incorrect array indexing
- Use of '=' instead of '←' in the increment of the index variable

(ii) There were very many vague attempts at stating both benefits and drawbacks. 'Easier to read' is an example of an answer that may have been the start of a valid point but on its own is insufficient.

Of the correct answers, the most popular was for some reference to the data being 'more difficult to interpret' or 'more difficult to separate'. Again 'difficult to read' is not meaningful enough.

Many candidates gained a mark for a sensible reference to the storage space taken up by the new design.

(d) Most candidates gained marks for the first two items of data but made errors with the last two groups by including `ThisIndex`.

A number of candidates labelled the diagram with data types or simply just 'input' and 'output'.

(e) A minority of candidates answered this part well. Here, there was significant use of code fragments that did not fit the given scenario. Two examples are the use of code relating to file handling and input/output; neither of these was required by the question.

Common problems included:

- not declaring a return parameter
- not having a loop
- incorrect comparison
- not correctly returning a value corresponding to found and not found
- outputting a message rather than returning a value.

Question 5

(i) to (v)

This question was mostly very well answered. No particular pattern of incorrect answers seen.

COMPUTER SCIENCE

Paper 9608/22
Written Paper

Key messages

In preparation for this examination, candidates were expected to have previously studied the pre-release material sent to Centres. This material included a range of tasks designed to help candidates practise their problem-solving and programming skills. This information, combined with past papers, gives a clear indication of the types of question that candidates can expect.

There were some excellent answers to the programming questions, but a significant number of candidates displayed low programming ability. Candidates need extensive practical programming experience before they sit this examination.

Many candidates appeared to be confused over the requirements of certain questions.
Candidates need to appreciate the importance of good examination technique and particularly the need to fully read and understand each question before they attempt to answer it.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings, and it is important that these are used correctly. It is also important for candidates to use the correct syntax when writing or explaining algorithms using pseudocode. Candidates particularly need to appreciate when it is appropriate to use the assignment operator '←' as instead of the '=' symbol.
It is also important for candidates to understand the use quotation marks to differentiate between an identifier name and the value of a character or string.

General comments

If a candidate writes the answer to a question on an additional page, they must indicate very clearly where their revised answer is to be found.

If answers have been crossed out, the new answers must be written very clearly so that the text can be read easily and candidates can be rewarded with the correct mark.

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

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

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

Candidates need to be clear if they are being asked to write pseudocode or program when reading a question. Some candidates incorrectly stated '*pseudocode*' as their programming language.

There is an increasing tendency for candidates to use incorrect sections of code in their answers to the programming questions. A particular example of this is the use of file-handling statements in a programming question where no file access is required.

Several questions ask the candidate to '*State*' or *Explain*' something. Centres need to emphasise that for these questions, it is not enough simply to repeat the words or phrases from the question itself. An example of this is the question that asks to differentiate between user-defined and built-in functions; in this case, it is not enough to simply state that 'a user-defined function is defined by the user' as this does not demonstrate any understanding.

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

Comments on specific questions

Question 1

(a) This was generally well answered. The majority of candidates scored at least six marks. A common mistake was to not recognise the CASE construct and to attempt to implement the selection using an `IF` construct.

Weaker responses gained marks simply by labelling the `INPUT` / `OUTPUT` symbols and for the final selection, `IF PointsTotal >= 12`.

(b) A wide range of marks was awarded, but the question was generally not answered well. Many candidates appeared to be unfamiliar with the use of test data.

Some candidates filled the table with variable names rather than data values, and a large proportion of candidates were unable to correctly predict what data went in each column.

A common mistake was not to include the updated points total when the player was '*Eliminated*'.

(c) This part was not answered well. Only around 40% of candidates gave the correct pre-condition loop construct and only a small number of candidates gained all three marks.

The question asked for a pre-condition loop, so a `WHILE ... ENDWHILE` construct was required. Many candidates gave a solution based on `REPEAT ... UNTIL`. Few candidates who used the correct loop construct understood that an additional `INPUT` statement was required before the loop.

The condition was often incorrect. Common mistakes included incorrect logic and missing quotation marks around the characters 'A', 'B', 'C' and 'D'.

Many candidates simply copied the text from the flowchart, giving:

```
WHILE PlayerGameGrade <> A OR B OR C OR D
```

This was incorrect both logically and syntactically.

A small number of more able candidates made use of a Boolean value as part of the loop termination. This was unnecessary but in many cases provided a completely workable solution.

Question 2

(a)(i) and (ii)

Many candidates correctly identified the required character and string, but only few candidates included the quotation marks that are necessary to differentiate between characters / strings and identifier names.

(iii) Most candidates had the right idea about concatenation. Around half of these candidates gave the answer as 132 instead of 213.

(b)(i) to (iv)

These parts were mostly answered correctly. Candidates seemed to have the most difficulty with part **(ii)**.

(c) Most candidates would have responded correctly to this if they had not included the `WHILE` in their answer. The question specifically asked for the *condition* that controlled the loop.

(d) (i) The difference between those candidates that had practised trace tables and those that had not was clearly evident, with candidates gaining either very high or very low marks for this question.

(ii) There was a wide spread of answers; some very clearly described the algorithm, but a large number demonstrated very little understanding. Some candidates described functions that simply were not present, for example, a 'bubble sort'.

Many candidates correctly stated that the algorithm 'separates the numbers'.

This was a question where clarity of technical description was important. For example, the pseudocode stores an `INTEGER` value in the array and not just a 'number'.

Question 3

(a) (i) This part was mostly correctly answered. Simply referring to 'a declaration' was not sufficient as adequate technical description was required.

(ii) Many correct responses were provided for this part, but a significant number of candidates omitted the leading '$' from the variable name.

(iii) Most candidates provided a correct response.

(iv) A significant number of candidates correctly identified the curly brackets / braces as the loop delimiters, but then many candidates simply wrote individual lines of code from the question as their answer.

(b) (i) A 'text book' question, which some candidates answered very well. Many candidates gave an answer that was little more than the question re-worded.

Common misconceptions included references to:

- the transfer of data between modules
- training: the transfer of knowledge from one programmer to another
- the skill that was 'built into the program'.

(ii) Candidates found this more difficult than the previous question and many did not recognise that they had themselves already made use of a transferrable skill by answering the previous parts of the question.

A small number of candidates correctly identified the ability to recognise specific features in an 'unknown' language.

Question 4

(a) From many of the answers given, it was apparent that some candidates had not fully read the contents of the appendix and so did not know how to use the function given in the question. Those that had read the appendix scored full marks.

(b) Most candidates who answered this question were able to score 1 or 2 marks, and good percentages were able to score 3 marks or more. It was clear which candidates had real programming experience as they provided succinct solutions for full marks.

Most candidates used or attempted to use a `FOR ... NEXT` loop, but many of these did not employ the correct syntax for the chosen language.
Many Python solutions lacked the necessary indentation to support the loop structure.

Solutions that did *not* make use of a loop were relatively few, but in many cases they were functionally correct.

Some candidates chose to declare additional variables that were not required.

Common mistakes included:

- the use of pseudocode
- declaring variables with invalid data types
- the use of `INPUT` statements to obtain the random values
- the inclusion of unrequired file handling.

(c) A minority of candidates answered this part well. Most candidates appeared to have little experience of writing modular code.

In addition to general errors of syntax, common mistakes included:

- in VB, use of `Dim` in place of `Function`
- incorrect parameter types or keywords.

(d) (i) Many candidates did not appear to understand the differences between adaptive, corrective and perfective maintenance. A significant number of responses were more perfective than adaptive.

Candidates commonly gained the first mark for referring to the code ‘being changed’. They often did not say that this change followed a change in the requirements.

(ii) Many candidates did not put their answers into sentences. The question offered a large hint by asking what type of data structure may be needed. The more able candidates were able to pick this up and correctly describe the use of an array, list or file. A common mistake was to attempt to use a single variable (either `INTEGER` or `BOOLEAN`) to store the previous winning numbers.

Many candidates achieved only the last mark point by stating that the number should be re-generated if it already exists but the rest of their explanations were often vague.

Question 5

(a) The majority of candidates correctly stated that the data structure was a 2 D array, but very few gained the second mark for stating either the data type or name. A common mistake was to omit the array dimension.

(b) (i) Another 'textbook' question, which was generally not answered well. Most candidates seemed to offer a computing term not related to the question.

A common mistake was to suggest 'Adaptive Maintenance'.

(ii) Those candidates that had some programming experience were able to write a good, succinct piece of code for this question, which represented a very straightforward task.

The majority of candidates would clearly benefit from more practical programming experience if they are to prepare adequately for this exam.

Candidates often gained the marks for opening and closing the file but many struggled with the rest of the code.

(iii) For some candidates there appeared to be confusion over whether they were answering this question or the previous one. Their solutions included file-handling functions when the question specifically states that the data is contained in an array.

Common errors included:

- the use of `FOR` loop rather than a conditional loop to exit when found
- a loop which would be infinite if the name was not found
- unnecessary `INPUT` / `OUTPUT` statements.

(c) (i) The majority of candidates did not use the correct technical terminology. Most candidates were unable to describe the structure as a count-controlled nested loop. Candidates must recognise and describe the fundamental programming structures.

Many candidates gave vague descriptions of the purpose of the instructions *within* the loop structure.

(ii) Candidates provided every permutation of `'TRUE'` and `'FALSE'`, and many left the question unanswered. The correct answer was given by less than half the candidates.

(iii) This part was answered well, with most of the correct answers giving the correction to line 11.

A small number of candidates identified the error in either line 5 or 9, but over-complicated their correction by adding additional unnecessary tests of the element from the `PlayerScore[]` array.

Question 6

(i) to (v) Most candidates correctly answered this last questions.

Incorrect answers were most likely to occur in parts **(iii)** and **(v)**, which perhaps suggests a lack of mathematical skill required to interpret operator precedence rather than a difficulty in following the logic of an expression.

COMPUTER SCIENCE

Paper 9608/23
Written Paper

Key messages

In preparation for this examination, candidates were expected to have previously studied the pre-release material sent to Centres. This material included a range of tasks designed to help candidates practise their problem-solving and programming skills. This information, combined with past papers, gives a clear indication of the types of question that candidates can expect.

There were some excellent answers to the programming questions, but a significant number of candidates displayed low programming ability. Candidates need extensive practical programming experience before they sit this examination.

Many candidates appeared to be confused over the requirements of certain questions.
Candidates need to appreciate the importance of good examination technique and particularly the need to fully read and understand each question before they attempt to answer it.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings, and it is important that these are used correctly. It is also important for candidates to use the correct syntax when writing or explaining algorithms using pseudocode. Candidates particularly need to appreciate when it is appropriate to use the assignment operator '←' as instead of the '=' symbol.
It is also important for candidates to understand the use quotation marks to differentiate between an identifier name and the value of a character or string.

General comments

If a candidate writes the answer to a question on an additional page, they must indicate very clearly where their revised answer is to be found.

If answers have been crossed out, the new answers must be written very clearly so that the text can be read easily and candidates can be rewarded with the correct mark.

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

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

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

Candidates need to be clear if they are being asked to write pseudocode or program when reading a question. Some candidates incorrectly stated '*pseudocode*' as their programming language.

There is an increasing tendency for candidates to use incorrect sections of code in their answers to the programming questions. A particular example of this is the use of file-handling statements in a programming question where no file access is required.

Several questions ask the candidate to '*State*' or *Explain*' something. Centres need to emphasise that for these questions, it is not enough simply to repeat the words or phrases from the question itself. An example of this is the question that asks to differentiate between user-defined and built-in functions; in this case, it is not enough to simply state that 'a user-defined function is defined by the user' as this does not demonstrate any understanding.

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

Comments on specific questions

Question 1

(a) This was generally well answered. The majority of candidates scored at least four marks. A common mistake was to omit the 'YES' and 'NO' labels from the outputs of the selection symbols.

Weaker candidates tended to gain two marks by labelling the two `INPUT` / `OUTPUT` symbols.

(b) A wide range of marks was awarded, but the question was only answered well by a minority of candidates. Marks were often lost due to inadequate explanations. A common mistake was to give a positive numeric value for `BaggageWeight` and to then explain that this was '*erroneous*' or '*normal*'. These terms were not appropriate to the scenario. Explanations needed to clearly refer to the baggage allowance.

Mistakes in the 'Expected output' included:

- not outputting '0' (zero) in the case of no excess charge
- incorrect calculations
- giving an expression (such as '3 * 3.50') rather than evaluating the excess charge.

(c) This question was not well answered. Only around 40% of candidates gave the correct pre-condition loop construct, and only a small number of candidates gained all three marks.

The question asked for a pre-condition loop, so a `WHILE ... ENDWHILE` construct was required. Many candidates gave a solution based on `REPEAT ... UNTIL`. Few candidates who used the correct loop construct realised that an additional `INPUT` statement was required before the loop.

The condition was often incorrect. Common mistakes included incorrect logic and missing quotation marks around the characters 'E' and 'S'.

Many candidates simply copied the text from the flowchart, giving:

```
WHILE TicketType <> E OR S
```

This was incorrect both logically and syntactically.

Question 2

(a) This question attracted a range of responses. A small number of students gave completely correct answers, but there were also a significant number of candidates who gained no marks. A number of candidates made no attempt at this question. Trace tables are important tools for debugging and as such are likely to feature in some form or other in many papers.

With reference to the published mark scheme, the marks most often awarded were for those at the start of the table; with perhaps the most commonly awarded mark being given for '`5 Error – investigate'`.

A number of candidates made a good start, but then made mistakes after the first set of six iterations.

(b) Many candidates correctly identified the different programming constructs.
Incorrect answers were normally limited to the 'selection' and 'iteration' parts of the question.

Question 3

(a)(i) and (ii)
These two parts were very well answered with the majority providing a correct answer.

(a)(iii) and (iv)
A minority of candidates answered these two parts correctly. Many candidates omitted any form of quotation marks.

(b) (i) There was a mixed response for this question.

Common mistakes included:

- enclosing the variable `Surname` in quotes (the problem of the difference between a string and an identifier)
- giving '`i`' as both the second **and** third parameter of the `SUBSTR()` function
- using a non-pseudocode function.

(ii) This was a straightforward exercise in translating an algorithm from pseudocode into a high level language. Many candidates gained full marks.

Common mistakes included:

- declaring `NextChar` as a `STRING` or `INTEGER`
- using a pseudocode function rather than one from the chosen high level language
- terminating the loop incorrectly.

As mentioned in the general comments section, candidates who give answers in Python must learn that indentation is an intrinsic part of the syntax and must be clearly shown.

(c) (i) This part was answered well by a minority of candidates. Candidates appeared to have most students have had little experience of writing modular code.

A significant number of candidates offered the name of a programming language, but nothing else.

In addition to general errors of syntax, common mistakes included:

- In VB, use of `Dim` in place of `Function`.
- incorrect parameter types

(ii) This part was not answered well. Many candidates offered no answer at all. A fairly common mistake was to attempt to define a modified function header. Several responses involved returning a value other than the required one.

(iii) This part was not answered well. There were many variations on the required answer, with a number having the variable `ThisID` appearing on the right-hand side of the assignment. Of those answers that were generally of the correct format, the most common mistake was to not have the surname "Wilkes" enclosed by quotation marks. As mentioned in the key messages section, many candidates did not seem clear about the difference between a literal string and an identifier name.

Some candidates seemed confused between the use of a procedure and a function.

(d) (i) Candidates generally gained one mark for this question. In many cases, the use of vague terminology caused them to fall short of the second mark. For example, the phrase 'already there' is not acceptable in place of 'part of the programming language'.

(ii) Many candidates correctly referred to the functions both needing to be called and both returning a value, but several candidates mistakenly used the term `OUTPUT` rather than `RETURN`.

A number of candidates referred to both types of function having a 'similar format', which was too vague.

Question 4

(a) A significant number of candidates gave completely incorrect answers, indicating a lack of understanding of the terms in the question. Many responses had the functions completely reversed, for example, suggesting that the compiler turned the executable code into source code.

A number of candidates did give an acceptable alternative to the word 'translate'.

(b) (i) The majority of candidates correctly suggested that 'errors are highlighted' in some form or another.

Several candidates offered answers that were little more than the question re-worded.

A significant number of candidates gave no answer, and there were many one word answers that were insufficient at best, for example, 'compiler'.

(ii) There were only a small number of correct answers.

As for the previous question, many candidates gave no answer, and there were many one word answers that were insufficient. For example 'Debugging'

(c) (i) This question had four main mark points; with a fifth being available for candidates who realised that type conversion would be required before a string variable could be stored in the `PRetailPrice` array of type `REAL`. This mark was given in a small number of cases, and it was encouraging to see that some candidates did recognise this requirement.

Many candidates gained no marks for this question. Common mistakes included:

- Not including the file name in quotation marks in the `OPEN` statement
- Not specifying the read mode (or specifying `WRITE` mode)
- Incorrect `EOF()` check
- Incorrect array indexing
- Use of '=' instead of '←' in the increment of the index variable

(ii) There were very many vague attempts at stating both benefits and drawbacks. 'Easier to read' is an example of an answer that may have been the start of a valid point but on its own is insufficient.

Of the correct answers, the most popular was for some reference to the data being 'more difficult to interpret' or 'more difficult to separate'. Again 'difficult to read' is not meaningful enough.

Many candidates gained a mark for a sensible reference to the storage space taken up by the new design.

(d) Most candidates gained marks for the first two items of data but made errors with the last two groups by including `ThisIndex`.

A number of candidates labelled the diagram with data types or simply just 'input' and 'output'.

(e) A minority of candidates answered this part well. Here, there was significant use of code fragments that did not fit the given scenario. Two examples are the use of code relating to file handling and input/output; neither of these was required by the question.

Common problems included:

- not declaring a return parameter
- not having a loop
- incorrect comparison
- not correctly returning a value corresponding to found and not found
- outputting a message rather than returning a value.

Question 5

(i) to (v)

This question was mostly very well answered. No particular pattern of incorrect answers seen.

COMPUTER SCIENCE

Paper 9608/31
Advanced Theory

General Comments

Stronger candidates were able to demonstrate a good knowledge of the topics examined on the paper. There were many candidates however who showed a more limited knowledge.

For the question on paging, only a very small number of candidates displayed any understanding of what is involved in this process. Questions on straight forward concepts - floating point representation, RPN - which have been examined previously caused problems for many candidates in this session.

Candidates performed better on those questions involving the topics of logic circuits, Boolean algebra, malware and asymmetric key cryptography.

Comments on Specific Questions

Question 1

Many candidates did not demonstrate adequate knowledge of binary floating-point representation and the application of that knowledge. Despite the inclusion of a diagram that showed the position of the binary point within the mantissa, some candidates used a fixed point representation where the binary point was assumed to be between the mantissa and the exponent. Indication for whether a given number was normalised or not was often answered well. The effect of changing the number of bits in either the exponent or the mantissa was also often answered well.

(a) Stronger candidates were successful in giving a correct floating point representation. The question did not state that the solution should be normalised and some candidates did provide a correct, un-normalised, representation. Many incorrect solutions did not demonstrate any evidence of an appropriate technique for converting a denary real number to a binary floating point representation

(b) Those candidates who had given a correct answer for **Part (a)** usually gave a correct answer for this question. Even if candidates had an incorrect mantissa in **Part (a)** credit was given for evidence of an appropriate technique being applied.

(c) Candidates found conversion from binary to denary less challenging and there were more correct solutions to this question. There were a number of answers that ignored the position of the given binary point and treated the number as a fixed point representation.

(d) (i) The majority of candidates correctly identified that the number given was not normalised.

(ii) Most candidates who had identified that the number was not normalised gave an appropriate justification for their choice. Some candidates did not explain clearly that the bits of interest were the leftmost two in the mantissa and consequently did not gain credit.

(e) Many candidates described correctly the effects of changing the number of bits in both the mantissa and the exponent.

Question 2

This question showed that candidates had some knowledge of the various processes involved in compilation. In **Part (b)** and **Part (c)** few candidates showed complete mastery of Reverse Polish Notation (RPN) and the evaluation of an RPN expression.

(a) The majority of candidates answered this question well. Candidates found it more challenging to correctly identify statements about the lexical analysis stage.

(b) (i) A minority of candidates gained full credit on this question.

(ii) Only the strongest candidates gained full credit on this question. Many answers did not cope with the unary minus operator. The brackets in the final part of the expression obviously aided candidates in determining the order of some of the operations.

(c) (i) A number of answers placed operators on the stack. In the evaluation of an RPN expression whenever an operator is encountered, the operation is immediately performed on the items on top of the stack and the result placed back on the stack. Credit was not given where the candidate failed to evaluate an operation and left the unfinished calculation (e.g. '4 + 1') on the stack.

(ii) Candidates often gave 'x*' correctly but did not include necessary brackets for the remainder of the expression. Credit was not given for answers where the operators were in an incorrect order.

(iii) This question proved challenging. Correct answers were likely to mention that brackets are not required. However few candidates gave an answer that explained that brackets were not required because operations were performed as the operator was encountered. Many answers gave reasons which were not closely enough connected to the question set such as: saves storage space, evaluation is faster, easier for computers to understand.

Question 3

This question proved challenging and the concept and operation of paging was not understood by a large majority of candidates. Despite the information given in the question there seem to be widespread confusion as to the difference between a page and a page frame.

(a) The vast majority of candidates did not interpret the question correctly. Stating that '245 is an address' was insufficient as that fact was already evident from the entry in the page table. Candidates were expected to recognise that each page frame is a fixed size block of memory and that consequently it occupies more than a single address. Therefore 245 would represent the address of the start of the 245th page frame, possibly from some base address.

(b) The required usage necessitated a hardware device that would have fast read/write speeds. So 'hard disc', 'hard drive' or 'solid state drive' were all acceptable answers but 'USB flash drive' and 'external hard disc' were not acceptable.

(c) (i) Correct answers to this question were rare. Many candidates' answers referred to the length of time the page had been in the page frame. Those who gave this answer did not recognise the problem that this data would need to be updated on a very regular basis. However the majority of candidates gave an answer that ignored any reference to time. This was despite the information about 'longest time' in the question.

(ii) Despite their answers to **Part (i)** many candidates gained credit by giving correct values for the second and third entries in the table row.

(iii) More candidates gave a correct response than was the case in **Part (i)**.

(iv) The majority of candidates gained some credit in this part. Only a small number of candidates recognised that the final entry in the table should be '0' for a page just entering memory.

(d) Very few candidates could indicate potential problems with both of the page replacement algorithms. Credit was most often awarded for recognising that a program longest resident may be being processed on a regular basis. As a consequence it would be inefficient to swap it out.

Question 4

The majority of candidates performed reasonably well on this question.

(a) (i) Most candidates completed the truth table accurately.

(ii) The majority of candidates were successful in naming the logic circuit.

(iii) This question proved challenging. Some candidates who identified the circuit as being that for a half adder incorrectly labelled A as the Sum and B as the Carry. The overwhelming majority of answers which attempted to explain why the choice of output labels was appropriate, did not match the values in the truth table to the function of a half adder: the addition and output when two bits are added together.

(b) (i) Many candidates gave a correct expression. Answers which used a multiply symbol (either '*' or 'X') or used AND, OR or NOT were not credited. It was usually the case that candidates producing this type of expression rewrote the expression in **Part (ii)** using a correct terminology.

(ii) A sizable minority of candidates produced a fully correct simplification. Many candidates could be awarded some credit by demonstrating appropriate use of Boolean algebra laws.

Question 5

There was a mixed performance on this question with some strong answers but many which gained no credit.

(a) (i) Most candidates knew that the top layer was the Application layer.

(ii) Correct answers were extremely rare. Clearly candidates did not recognise that the rules associated with a protocol have to be implemented as code. Viewing the protocol suite as a layered stack allows for the code to be split into well-defined individual modules.

(b) (i) Although the question asked for tasks associated with incoming data, a number of candidates wrote about activities associated with outgoing data. The reassembly of the packets into order was a popular correct answer but not many candidates mentioned that this was achieved by using the packet sequence number. Although there were answers regarding integrity of the packet, candidates usually wrote, incorrectly, about parity rather than about checksums.

(ii) A majority of candidates correctly identified the required protocol.

(iii) A majority of candidates correctly identified the required protocol.

Question 6

Most candidates demonstrated good knowledge of malware and asymmetric key cryptography.

(a) Most candidates correctly identified at least two of the terms. More candidates did not identify 'Virus' in comparison to 'Pharming' and 'Phishing'. Only a small minority of answers confused 'Pharming' and 'Phishing'.

(b) (i) Candidates needed to be clear that the plain text referred to the original email. Many candidates did not do this. More candidates gained credit for explaining the term 'cipher text'.

(ii) There were many candidates who gave a very clear description of the necessary processes for the scenario given and gained full marks. Many answers referred to 'a public key' or 'a private key' or 'the public key'. For credit to be given, the owner of any key being used needed to be identified.

COMPUTER SCIENCE

Paper 9608/32
Advanced Theory

General Comments

Stronger candidates were able to demonstrate a good knowledge of the topics examined on the paper. There were many candidates however who showed a more limited knowledge.

For the question on paging, only a very small number of candidates displayed any understanding of what is involved in this process. Questions on straight forward concepts - floating point representation, RPN - which have been examined previously caused problems for many candidates in this session.

Candidates performed better on those questions involving the topics of protocols, logic circuits, Boolean algebra, and networks.

Comments on Specific Questions

Question 1

Many candidates failed to demonstrate adequate knowledge of binary floating-point representation and the application of that knowledge. Despite the inclusion of a diagram that showed the position of the binary point within the mantissa, some candidates used a fixed point representation where the binary point was assumed to be between the mantissa and the exponent. Indication for a normalised number was often answered well.

(a) Stronger candidates were successful in giving a correct floating point representation. The question did not state that the solution should be normalised and some candidates did provide a correct, un-normalised, representation. Many incorrect solutions failed to demonstrate any evidence of an appropriate technique for converting a denary real number to a binary floating point representation.

(b) Those candidates who had given a correct answer for **Part (a)** usually gave a correct answer for this question. Even if candidates had an incorrect mantissa in **Part (a)** credit was given for evidence of an appropriate technique being applied.

(c) Candidates found conversion from binary to denary less challenging and there were more correct solutions to this question. There were a number of answers that ignored the position of the given binary point and treated the number as a fixed point representation.

(d) (i) The majority of candidates correctly identified that the number given was normalised.

(ii) Most candidates who had identified that the number was normalised gave an appropriate justification for their choice. Some candidates failed to explain clearly that the bits of interest were the leftmost two in the mantissa and consequently failed to gain credit.

(e) Only a small minority of candidates gave both the correct mantissa and exponent. A common incorrect answer was to reverse the mantissa and exponent.

Question 2

This question showed that candidates had some knowledge of the various processes involved in compilation. In **Part (b)** and **Part (c)** few candidates showed complete understanding of Reverse Polish Notation (RPN) and the evaluation of an RPN expression.

(a) The majority of candidates answered this question well. The last two statements were less often answered correctly than the first two statements.

(b) A minority of candidates gained full credit on this question.

(c) (i) Fully correct answers were rare. Some candidates made the fundamental error of placing the operators on the stack. Many answers gave stack values which demonstrated a clear lack of understanding of how an RPN expression is evaluated.

(ii) Candidates often gave correctly 'b*a' but did not include necessary brackets for the remainder of the expression. Credit was not given for answers where the operators were not in the correct order.

(iii) Many candidates used the word 'BODMAS' but did not explain what this meant. On its own, the word does not constitute an explanation. Most answers that gained any credit referred to the fact that brackets are not required in an RPN expression. Only a very small number of candidates mentioned that in the evaluation of an RPN expression, operations are carried out as the operators are encountered.

Question 3

This question proved challenging and the concept and operation of paging was not understood by a majority of candidates.

(a) Despite the information given in the question, few candidates could explain what a presence flag value of 1 indicated and also what a page frame address of 542 indicated.

(b) (i) The majority of candidates thought that the situation described in the question was catastrophic and that the program would likely crash or be aborted. There was very little understanding that the next instruction to be executed would be the first instruction in page 6, a page currently not in memory, and that this type of event was nothing unusual in paging systems. All that was required to continue execution of the program was for page 6 had to be swapped into memory.

(ii) This was only answered well by the strongest candidates. Most candidates treated the question as a general question about how interrupts are handled. Few candidates showed understanding that a page fault would occur, an interrupt would be generated and, as a consequence, pages would be swapped.

(c) (i) Candidates often gave 'time in memory' as their answer. This answer was not acceptable as this is a data value that, for every page in memory, would have to be updated on a regular basis. Very few candidates gave the unchanging 'time of entry into memory' as a suitable data item. Despite the information given in the question, many incorrect answers were not time based.

(ii) Despite their answers to **Part (i)** many candidates gained credit by giving correct values for the second and third entries in the table row.

(iii) Again it was evident from the answers that paging and page swapping were not understood by the majority of candidates.

(iv) Only a few candidates recognised that the scenario described would result in thrashing.

Question 4

Some parts of this question were answered well.

(a) (i) This was usually answered well.

(ii) Only a minority of candidates identified an appropriate client application.

(iii) Those candidates who gave a correct answer for **Part (ii)** had few problems in giving a correct server.

(iv) Full credit was rare but many candidates gained partial credit, often giving 'security' as their answer.

(b) The vast majority of answers ignored the explicit need to give a parameter. However credit was given for answers where it was clear what parameter was being described.

(c) Candidates could usually name at least one appropriate application. Shopping and banking applications were popular. Descriptions of processes without an identifiable application were given no credit.

Question 5

The majority of candidates performed reasonably well on this question, gaining credit in a number of the part questions.

(a) (i) Many candidates did not produce a fully correct truth table with a large number producing a table with multiple errors.

(ii) Many candidates recognised the full adder logic circuit including some who had not produced a correct truth table in **Part (i)**.

(iii) This proved challenging for many candidates. Some candidates who identified the circuit as being that for a full adder incorrectly labelled J as the Sum and K as the Carry. The majority of answers, which attempted to explain why the choice of output labels was appropriate, did not match the values in the truth table to the function of a full adder: the addition and output when three bits are added together.

(b) (i) Many candidates gave a correct expression. Answers which used a multiply symbol (either '*' or 'X') or used AND, OR or NOT were not credited. It was usually the case that candidates producing this type of expression rewrote the expression in **Part (ii)** using a correct terminology.

(ii) Relatively few candidates could produce a fully correct simplification, although most demonstrated appropriate use of Boolean algebra laws.

Question 6

Many candidates answered this question well.

(a) Most answers demonstrated a star topology. However a significant number of these were incorrect as the server rather than the switch was the device chosen to have connections to all the other devices.

(b) Most candidates gained partial credit but fully correct answers were rare.

(c) (i) Many candidates gave a correct device with router being the most popular choice. A gateway is not a device so was not accepted.

(ii) Many answers mentioned the use of addresses. Credit was only given where the correct type of address – IP address for a router and MAC address for a bridge - matched the device given in **Part (i)**. Only a few candidates described the use of an address table for making decisions about packet destinations.

COMPUTER SCIENCE

Paper 9608/33
Advanced Theory

General Comments

Stronger candidates were able to demonstrate a good knowledge of the topics examined on the paper. There were many candidates however who showed a more limited knowledge.

For the question on paging, only a very small number of candidates displayed any understanding of what is involved in this process. Questions on straight forward concepts - floating point representation, RPN - which have been examined previously caused problems for many candidates in this session.

Candidates performed better on those questions involving the topics of logic circuits, Boolean algebra, malware and asymmetric key cryptography.

Comments on Specific Questions

Question 1

Many candidates did not demonstrate adequate knowledge of binary floating-point representation and the application of that knowledge. Despite the inclusion of a diagram that showed the position of the binary point within the mantissa, some candidates used a fixed point representation where the binary point was assumed to be between the mantissa and the exponent. Indication for whether a given number was normalised or not was often answered well. The effect of changing the number of bits in either the exponent or the mantissa was also often answered well.

(a) Stronger candidates were successful in giving a correct floating point representation. The question did not state that the solution should be normalised and some candidates did provide a correct, un-normalised, representation. Many incorrect solutions did not demonstrate any evidence of an appropriate technique for converting a denary real number to a binary floating point representation

(b) Those candidates who had given a correct answer for **Part (a)** usually gave a correct answer for this question. Even if candidates had an incorrect mantissa in **Part (a)** credit was given for evidence of an appropriate technique being applied.

(c) Candidates found conversion from binary to denary less challenging and there were more correct solutions to this question. There were a number of answers that ignored the position of the given binary point and treated the number as a fixed point representation.

(d) (i) The majority of candidates correctly identified that the number given was not normalised.

(ii) Most candidates who had identified that the number was not normalised gave an appropriate justification for their choice. Some candidates did not explain clearly that the bits of interest were the leftmost two in the mantissa and consequently did not gain credit.

(e) Many candidates described correctly the effects of changing the number of bits in both the mantissa and the exponent.

Question 2

This question showed that candidates had some knowledge of the various processes involved in compilation. In **Part (b)** and **Part (c)** few candidates showed complete mastery of Reverse Polish Notation (RPN) and the evaluation of an RPN expression.

(a) The majority of candidates answered this question well. Candidates found it more challenging to correctly identify statements about the lexical analysis stage.

(b) (i) A minority of candidates gained full credit on this question.

(ii) Only the strongest candidates gained full credit on this question. Many answers did not cope with the unary minus operator. The brackets in the final part of the expression obviously aided candidates in determining the order of some of the operations.

(c) (i) A number of answers placed operators on the stack. In the evaluation of an RPN expression whenever an operator is encountered, the operation is immediately performed on the items on top of the stack and the result placed back on the stack. Credit was not given where the candidate failed to evaluate an operation and left the unfinished calculation (e.g. '4 + 1') on the stack.

(ii) Candidates often gave 'x*' correctly but did not include necessary brackets for the remainder of the expression. Credit was not given for answers where the operators were in an incorrect order.

(iii) This question proved challenging. Correct answers were likely to mention that brackets are not required. However few candidates gave an answer that explained that brackets were not required because operations were performed as the operator was encountered. Many answers gave reasons which were not closely enough connected to the question set such as: saves storage space, evaluation is faster, easier for computers to understand.

Question 3

This question proved challenging and the concept and operation of paging was not understood by a large majority of candidates. Despite the information given in the question there seem to be widespread confusion as to the difference between a page and a page frame.

(a) The vast majority of candidates did not interpret the question correctly. Stating that '245 is an address' was insufficient as that fact was already evident from the entry in the page table. Candidates were expected to recognise that each page frame is a fixed size block of memory and that consequently it occupies more than a single address. Therefore 245 would represent the address of the start of the 245th page frame, possibly from some base address.

(b) The required usage necessitated a hardware device that would have fast read/write speeds. So 'hard disc', 'hard drive' or 'solid state drive' were all acceptable answers but 'USB flash drive' and 'external hard disc' were not acceptable.

(c) (i) Correct answers to this question were rare. Many candidates' answers referred to the length of time the page had been in the page frame. Those who gave this answer did not recognise the problem that this data would need to be updated on a very regular basis. However the majority of candidates gave an answer that ignored any reference to time. This was despite the information about 'longest time' in the question.

(ii) Despite their answers to **Part (i)** many candidates gained credit by giving correct values for the second and third entries in the table row.

(iii) More candidates gave a correct response than was the case in **Part (i)**.

(iv) The majority of candidates gained some credit in this part. Only a small number of candidates recognised that the final entry in the table should be '0' for a page just entering memory.

(d) Very few candidates could indicate potential problems with both of the page replacement algorithms. Credit was most often awarded for recognising that a program longest resident may be being processed on a regular basis. As a consequence it would be inefficient to swap it out.

Question 4

The majority of candidates performed reasonably well on this question.

(a) (i) Most candidates completed the truth table accurately.

(ii) The majority of candidates were successful in naming the logic circuit.

(iii) This question proved challenging. Some candidates who identified the circuit as being that for a half adder incorrectly labelled A as the Sum and B as the Carry. The overwhelming majority of answers which attempted to explain why the choice of output labels was appropriate, did not match the values in the truth table to the function of a half adder: the addition and output when two bits are added together.

(b) (i) Many candidates gave a correct expression. Answers which used a multiply symbol (either '*' or 'X') or used AND, OR or NOT were not credited. It was usually the case that candidates producing this type of expression rewrote the expression in **Part (ii)** using a correct terminology.

(ii) A sizable minority of candidates produced a fully correct simplification. Many candidates could be awarded some credit by demonstrating appropriate use of Boolean algebra laws.

Question 5

There was a mixed performance on this question with some strong answers but many which gained no credit.

(a) (i) Most candidates knew that the top layer was the Application layer.

(ii) Correct answers were extremely rare. Clearly candidates did not recognise that the rules associated with a protocol have to be implemented as code. Viewing the protocol suite as a layered stack allows for the code to be split into well-defined individual modules.

(b) (i) Although the question asked for tasks associated with incoming data, a number of candidates wrote about activities associated with outgoing data. The reassembly of the packets into order was a popular correct answer but not many candidates mentioned that this was achieved by using the packet sequence number. Although there were answers regarding integrity of the packet, candidates usually wrote, incorrectly, about parity rather than about checksums.

(ii) A majority of candidates correctly identified the required protocol.

(iii) A majority of candidates correctly identified the required protocol.

Question 6

Most candidates demonstrated good knowledge of malware and asymmetric key cryptography.

(a) Most candidates correctly identified at least two of the terms. More candidates did not identify 'Virus' in comparison to 'Pharming' and 'Phishing'. Only a small minority of answers confused 'Pharming' and 'Phishing'.

(b) (i) Candidates needed to be clear that the plain text referred to the original email. Many candidates did not do this. More candidates gained credit for explaining the term 'cipher text'.

(ii) There were many candidates who gave a very clear description of the necessary processes for the scenario given and gained full marks. Many answers referred to 'a public key' or 'a private key' or 'the public key'. For credit to be given, the owner of any key being used needed to be identified.

COMPUTER SCIENCE

Paper 9608/41
Written Paper

Key messages

It is essential that candidates have practice experience of programming (including object-oriented programming) using one of the following languages: Pascal/Delphi (console mode), VB.NET (console mode) or Python. Programming and pseudocode questions from previous syllabus past papers and the tasks in the pre-release material provide ideal topics for practical work.

General comments

The pre-release material can provide scenarios and topics to aid teaching. Some candidates wrote solutions, or partial solutions using pseudocode code, or languages other than that identified. Candidates need to produce program code in the language they declare at the beginning of the question part.

Comments on specific questions

Question 1

(a) Most candidates answered this question well and gained full marks.

(b) (i) Some candidates gave a correct method header. Fewer candidates were able to initialise the code to `""` or `State` to `"Open-NoCode"`.

(ii) Many candidates gave a correct method header, but were unable to set the `Code` to be blank, often incorrectly setting it to `0` instead.

(iii) Many candidates gave a method header that did not take a parameter. This often led to the state being set incorrectly. Some candidates read in a new state to be output incorrectly, instead of outputting the state sent as a parameter.

(iv) Many candidates did not set a parameter in the method header, and were unable to set the code to become this parameter. Some candidates incorrectly attempted to read in a new code and set the code to this value. Some candidates were able to proceed and output the code after it had been changed.

(v) Few candidates were able to gain all the marks in this question. Candidates need to be aware of the differences between functions and procedures and be able to write code for both. Many candidates wrote a function header (for VB.NET or Pascal) that did not identify the data type to be returned, or did this incorrectly. Many candidates were able to check the length of the string, but very few attempted to check that each character was numerical. Many candidates were able to return true or false by either setting them to the function name, or using a return keyword.

(vi) There were a range of answers for this question. Many candidates were able to gain a significant number of the marks. Common errors included not using the `SetState` method to change the state, and outputting the error message in incorrect places.

(vii) Once a class has been created, instances of the class need to be declared and initialised using the constructor method. Candidates need to have experience of declaring classes and then produces and manipulating instances of this class. Only a few candidates were able to gain marks for this question, most commonly for writing an infinite loop. Some candidates were able to create a new instance of `SafetyDepositBox`, but very few were able to apply the `StateChange` method to `This Safe`.

(c) Candidates should have experience of declaring properties and methods as both public and private, and therefore have an understanding of the impact of these on the programs that they write.

(i) Some candidates were able to give clear descriptions of the how declaring the properties as private affects where and how they can be accessed, most commonly that they can only be accessed in the class. Many candidates gave answers that did not clearly identify where they could be accessed. Candidates need to be confident in using the correct terminology in respect to classes, objects, properties and methods. Many candidates incorrectly stated that private meant users could not see them, or could not use them.

(ii) Some candidates were able to identify that the private methods could not be called outside the class, but many candidates were unable to give a second point, or explain fully that the public methods could be called by the main program.

Question 2

(a) (i) Most candidates gave correct IDE tools, but some candidates gave error detection tools which are used to debug the code, not write the code.

(ii) Most candidates correctly identified how the syntax error was shown, but some gave a definition of what a syntax error is, as opposed to when it is detected.

(iii) Some candidates were able to correctly identify the line where the syntax error was, and then correct it. Some candidates did not make the correction, and repeated the same line of code.

(b) (i) Many candidates who used VB.NET as their language often incorrectly stated an interpreter, or stated that VB.NET uses both one after the other.

(ii) Many candidates were able to correctly identify that there was a logic error. Few candidates were able to identify and correct this error.

(iii) Many candidates answered this question well, identifying two tools. Fewer candidates were able to give descriptions. Common errors included identifying what the tool did i.e. stops the program at a specific point, but did not describe how it could be used to find the error.

Question 3

Most candidates gained some marks on this question, with some candidates getting it completely correct. A common error was where candidates were unable to increment COUNT, and they attempted to increment the memory location direct. Other common errors included missing the # in front of the numbers, and using incorrect addressing to load a number of memory locations.

Question 4

(a) Most candidates were able to identify that the end user performs acceptance testing. Fewer could identify when this takes place, but most correctly identified the purpose.

(b) Most candidates could identify that the programmer performs the integration testing. Fewer candidates gave sufficient detail for when, with some candidates stating it was during development. This did not explain where in the software development process it takes place. Many candidates correctly identified that it ensures the modules work together when combined.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

It is essential that candidates have practice experience of programming (including object-oriented programming) using one of the following languages: Pascal/Delphi (console), VB.NET (console) or Python. Programming and pseudocode questions from previous syllabus past papers and the tasks in the pre-release material provide ideal topics for practical work.

General comments

The pre-release material can provide scenarios and topics to aid this teaching. Some candidates wrote solutions, or partial solutions using pseudocode, or languages other than those identified. Candidates need to produce program code in the language they declare at the beginning of the question part.

Comments on specific questions

Question 1

(a) Many candidates gave a correct answer to this question, correctly labelling the activities.

(b) (i) Some good responses were given, with candidates correctly setting amount to `0`. A common error was setting state to an empty string (" "). Candidates need to know how to write a constructor method for a class in the programming language they are taught in.

(ii) Many candidates gave a method header that did not take a parameter. This led to the state being set incorrectly. Some candidates read in a new state that was then output whilst some candidates were able to output the state correctly.

(iii) Many candidates were able to output the amount or set the amount to be `0`. A common error was the order that these took place in, meaning the new value of `0` was output instead of the previous value.

(iv) Candidates need to be aware of the differences between functions and procedures and be able to write code for both. Many candidates wrote a function header (for VB.NET or Pascal) that did not identify the data type to be returned, or that did this syntactically incorrectly. Only a few candidates were able to write a correct if statement, with common errors that included missing speech marks around the string values 10, 20, 50 and 100, or missing the 's =' before each comparison. Many candidates correctly returned true or false, by either setting them to the function name, or using a return key word.

(v) Candidates need to be aware of the differences in data types and how to convert from one data type to another. Few candidates were able to convert the string parameter to an integer. A greater number of candidates correctly added the value to the amount.

(vi) There were a range of answers for this question. Many candidates were able to gain a significant number of the marks. Commons errors included missing speech marks around strings in the `IF` statements, and not setting the state to be "Idle" in the correct places. Some candidates were unable to use the `setState` procedure calls correctly, and outputting the new state instead of setting it.

(vii) Once a class has been created, instances of the class need to be declared and initialised using the constructor method. Candidates need to have experience of declaring classes and then produces and manipulating instances of this class. Only a few candidates were able to gain marks for this question, most commonly for writing an infinite loop. Some candidates correctly created a new instance of `TicketMachine`, very few candidates were able to call the `StateChange` method on `ParkingMeter` or declare the main method that the code should run in.

(c) Candidates should have experience of declaring properties and methods as both public and private, and therefore have an understanding of the impact of these on the programs that they write.

(i) Some candidates were able to give clear descriptions of the how declaring the properties as private affects where and how they can be accessed, most commonly that they can only be accessed in the class. Many candidates gave answers that did not clearly identify where they could be accessed. Candidates need to be confident in using the correct terminology in respect to classes, objects, properties and methods. Many candidates incorrectly stated that private meant users could not see them, or could not use them.

(ii) Some candidates were able to identify that the private methods could not be called outside the class, but many candidates were unable to give a second point, or explain fully that the public methods could be called by the main program.

Question 2

(i) Most candidates were able to identify that alpha testing is carried out by those writing the program or by in-house testers. Fewer could fully explain when it was carried out. Many candidates stated this was whilst it was written. Few candidates could distinguish the exact purpose of alpha testing, and gave a generic description of testing such as to find errors.

(ii) Many candidates correctly identify that end users carry out beta testing. A common error was to state that it was by the general public. This is not an accurate enough answer. Some candidates correctly identified when it takes place. A common error was to state that it was after the program was written which was insufficient to identify exactly when in the software development process this took place. Many candidates correctly identified the purpose.

Question 3

(a) (i) Some candidates gave the correct bitwise operation. Many candidates gave the binary number that was the answer to part **(ii)** as their answer here.

(ii) Some candidates gave the correct bitwise operation. Many candidates gave the incorrect binary and gave the answer to part **(i)** here.

(b) Candidates should be familiar with the assembly language instructions given in the specification, and have experience of reading, tracing and writing assembly language programs. Most candidates gained some marks on this question, with some candidates getting it completely correct. Common errors included attempting to increment COUNT directly, missing the # in front of the numbers, and including the <> given in the instruction table in the code they wrote. Some candidates mixed up the different types of load instruction. Few candidates correctly added the binary code to the masks.

Question 4

(a) (i) Many candidates gave correct IDE tools. Some candidates gave error detection tools which are used to debug the code and not write the code. Other common errors included the writing of

comments and indentation (automatic indentation was an appropriate tool, but indentation can be performed without an IDE).

(ii) Most candidates correctly identified how the syntax error was shown, but some gave a definition of what a syntax error is, as opposed to when it is detected.

(iii) Few candidates were able to correctly identify the syntax error. Many candidates identified logic errors, or gave the correct line number with an incorrect (or many times, unchanged) line of code.

(b) (i) Many candidates Who chose VB.NET as their language often incorrectly stated an interpreter, or stated that VB.NET uses both and a compiler, one after the other.

(ii) Many candidates were able to correctly identify that there was a logic error. Few candidates were able to identify and correct this error.

(iii) Many candidates answered this question well, identifying two tools. Fewer candidates were able to give descriptions. Common errors included identifying what the tool did i.e. stops the program at a specific point, but did not describe how it could be used to find the error. Some candidates incorrectly described different types of testing (e.g. black box and white box) as tools for debugging.

COMPUTER SCIENCE

Paper 9608/43
Written Paper

Key messages

It is essential that candidates have practice experience of programming (including object-oriented programming) using one of the following languages: Pascal/Delphi (console mode), VB.NET (console mode) or Python. Programming and pseudocode questions from previous syllabus past papers and the tasks in the pre-release material provide ideal topics for practical work.

General comments

The pre-release material can provide scenarios and topics to aid teaching. Some candidates wrote solutions, or partial solutions using pseudocode code, or languages other than that identified. Candidates need to produce program code in the language they declare at the beginning of the question part.

Comments on specific questions

Question 1

(a) Most candidates answered this question well and gained full marks.

(b) (i) Some candidates gave a correct method header. Fewer candidates were able to initialise the code to `""` or `State` to `"Open-NoCode"`.

(ii) Many candidates gave a correct method header, but were unable to set the `Code` to be blank, often incorrectly setting it to `0` instead.

(iii) Many candidates gave a method header that did not take a parameter. This often led to the state being set incorrectly. Some candidates read in a new state to be output incorrectly, instead of outputting the state sent as a parameter.

(iv) Many candidates did not set a parameter in the method header, and were unable to set the code to become this parameter. Some candidates incorrectly attempted to read in a new code and set the code to this value. Some candidates were able to proceed and output the code after it had been changed.

(v) Few candidates were able to gain all the marks in this question. Candidates need to be aware of the differences between functions and procedures and be able to write code for both. Many candidates wrote a function header (for VB.NET or Pascal) that did not identify the data type to be returned, or did this incorrectly. Many candidates were able to check the length of the string, but very few attempted to check that each character was numerical. Many candidates were able to return true or false by either setting them to the function name, or using a return keyword.

(vi) There were a range of answers for this question. Many candidates were able to gain a significant number of the marks. Common errors included not using the `SetState` method to change the state, and outputting the error message in incorrect places.

(vii) Once a class has been created, instances of the class need to be declared and initialised using the constructor method. Candidates need to have experience of declaring classes and then produces and manipulating instances of this class. Only a few candidates were able to gain marks for this question, most commonly for writing an infinite loop. Some candidates were able to create a new instance of `SafetyDepositBox`, but very few were able to apply the `StateChange` method to `This Safe`.

(c) Candidates should have experience of declaring properties and methods as both public and private, and therefore have an understanding of the impact of these on the programs that they write.

(i) Some candidates were able to give clear descriptions of the how declaring the properties as private affects where and how they can be accessed, most commonly that they can only be accessed in the class. Many candidates gave answers that did not clearly identify where they could be accessed. Candidates need to be confident in using the correct terminology in respect to classes, objects, properties and methods. Many candidates incorrectly stated that private meant users could not see them, or could not use them.

(ii) Some candidates were able to identify that the private methods could not be called outside the class, but many candidates were unable to give a second point, or explain fully that the public methods could be called by the main program.

Question 2

(a) (i) Most candidates gave correct IDE tools, but some candidates gave error detection tools which are used to debug the code, not write the code.

(ii) Most candidates correctly identified how the syntax error was shown, but some gave a definition of what a syntax error is, as opposed to when it is detected.

(iii) Some candidates were able to correctly identify the line where the syntax error was, and then correct it. Some candidates did not make the correction, and repeated the same line of code.

(b) (i) Many candidates who used VB.NET as their language often incorrectly stated an interpreter, or stated that VB.NET uses both one after the other.

(ii) Many candidates were able to correctly identify that there was a logic error. Few candidates were able to identify and correct this error.

(iii) Many candidates answered this question well, identifying two tools. Fewer candidates were able to give descriptions. Common errors included identifying what the tool did i.e. stops the program at a specific point, but did not describe how it could be used to find the error.

Question 3

Most candidates gained some marks on this question, with some candidates getting it completely correct. A common error was where candidates were unable to increment COUNT, and they attempted to increment the memory location direct. Other common errors included missing the # in front of the numbers, and using incorrect addressing to load a number of memory locations.

Question 4

(a) Most candidates were able to identify that the end user performs acceptance testing. Fewer could identify when this takes place, but most correctly identified the purpose.

(b) Most candidates could identify that the programmer performs the integration testing. Fewer candidates gave sufficient detail for when, with some candidates stating it was during development. This did not explain where in the software development process it takes place. Many candidates correctly identified that it ensures the modules work together when combined.