COMPUTING

Paper 9691/11
Written Paper

General comments

The standard of candidates' work was an improvement on last year in many areas.

The format of the examination was similar to November 2013. The candidates seemed better prepared for this new style of paper than they were twelve months ago. The new format leads candidates and Centres gradually into the revised syllabus which has its first sitting in June 2015.

Candidates need to understand that it is no longer the case that simply learning sections from text books will score many marks on future papers. There is a need to understand how to apply their knowledge in new scenarios (both in examinations and in their learning of the subject). This will hopefully give a better grounding of the subject but also enable students of all abilities to perform to their full potential in the examination.

Comments on Specific Questions

Question 1

In part **(i)**, many candidates answered this question in terms of the fetch execute cycle and so gained the mark. However, there were imprecise answers about the control unit controlling things which was not creditworthy.

Again, in part **(ii)**, candidates who mentioned either ROM or RAM gained the mark, but there were also many imprecise answers about just storing data.

In the final part, many candidates did well with some good descriptions of the function of the ALU given.

Question 2

(a) In part **(i)**, most candidates realised that the system would be out of date and so got one mark; but

as with the first part of the question, most candidates realised that the system would need to be updated and so gained one mark. Some candidates needed to expand on this for the second mark.

(b) This question was essentially book work. Whilst many candidates had clearly learned the definitions of adaptive, corrective and perfective maintenance, a number of candidates unfortunately confused this with implementation and described questionnaires, interviews and observation.

Question 3

(a) This question was answered better than similar questions in earlier papers. The idea of analogue and digital now appears to be well known, but there needs to be further understanding of what is meant by the terms "unidirectional" and "bidirectional". Also, a small number of candidates described serial and parallel transmission rather than baseband and broadband.

(b) The links between the terms and the definitions were correct in all but a very small number of cases. The links between the definitions and the examples needed better understanding. The right hand side of the diagram required more application than just pure knowledge.

Question 4

(a) Candidates scored well. Most candidates were able to describe what was meant by a computer virus in enough detail to gain the single mark.

(b) Candidates need to improve on their examination technique, especially reading the instruction carefully in the question. Many candidates imprecisely listed items such as anti-virus, firewall etc. and did not describe or give further detail. Where it is necessary to give a description, simple one word answers are not creditworthy. It is important that candidates read the stem of questions to make sure their answer is at the required depth.

Question 5

(a) Most candidates gave the correct value.

(b) In part **(i)**, this question required the answer *2 dimensional array.* Many candidates gained a mark for "array" but did not indicate the array dimensions.

In part **(ii)**, the full range of marks (0 to 4) were seen.

Question 6

(a) This question was very well answered with many candidates gaining the maximum mark. Candidates need to improve on their drawings of logic gates. It was helpful where a number of candidates had written the gate name inside the shape which made it easier to understand. Without the words, in many cases, it would have been almost impossible to interpret which gate had been drawn. Candidates should practice drawing gates so that it is clear to an Examiner what answer they are giving.

(b) Again many candidates gained the full marks here.

Question 7

(a) For measuring the noise level a sound sensor was a popular and correct answer. Some candidates however needed to improve their knowledge for this question as they gave the incorrect answer of decibel meter.

Candidates needed to develop their knowledge for this question. For measuring air pollution levels there were a large number of different incorrect answers. Those candidates who had carefully read the question stated the correct answers such as carbon dioxide sensor or nitrogen oxide sensor.

(b) Candidates needed to be better prepared for this question and may have needed to read through it carefully to ensure that they understood it. Some candidates appeared to misunderstand the scenario and seemed to expect that the monitoring stations would be connected back to base by all sorts of cabling etc. Candidates needed to develop their understanding that the monitoring needed to be continuous; some wrote that the solid state memory could be removed and taken back to base to be read, and admittedly this could be done, but only if the memory is replaced with a second one, so disrupting readings only for a very short time. Candidates should have mentioned the use of a portable device.

(c) Generally most candidates appreciated that using a spreadsheet was the best software to use. In some cases, expansions on how a spreadsheet could be used could have been clearer and in more detail to obtain all the marks available. Some candidates did mention graphs and trends, but in order for these responses to be creditworthy, the candidates needed to develop their answer rather than just taking the wording from the question stem.

Question 8

Parts **(i)**, **(ii)**, **(iii)** and **(v)** were very well answered; part **(iv)** however was the exception. There was much confusion between a WAN and a WLAN, so the most common answer was that LANs can also operate over WiFi and are then called WANs. Some candidates wrote about LANs covering a small area and WANs covering a large area which did not answer the question.

Question 9

(a) A large number of candidates correctly gave a touch screen as the suitable input device. Incorrect answers, such as mouse and keyboard, were also given which would not be regarded as suitable devices in this scenario. When selecting input and output devices, candidates should read the scenario very carefully to make sure their choice is appropriate.

(b) Many candidates did well in this question.

(c) Candidates needed to be better prepared for this question. Candidates doing project work should be aware of the type of testing necessary to ensure software, such as expert systems, operates correctly. Many candidates gave imprecise descriptions of normal and abnormal data or gave statements such as black box or white box testing without fully clarifying their response.

Question 10

(a) A good range of marks were seen here. The main errors were in the second statement and fifth statement. Many candidates needed to improve their knowledge of custom-written software and off-the-shelf software.

(b) This was fairly well answered with many candidates gaining full marks.

(c) Some very general answers such as user guide or how to use the software were stated here. Some candidates needed to develop their understanding of user documentation and technical documentation.

COMPUTING

Paper 9691/12
Written Paper

General comments

The standard of candidates' work was an improvement on last year in many areas.

The format of the examination was similar to November 2013. The candidates seemed better prepared for this new style of paper than they were twelve months ago. The new format leads candidates and Centres gradually into the revised syllabus which has its first sitting in June 2015.

Candidates need to understand that it is no longer the case that simply learning sections from text books will score many marks on future papers. There is a need to understand how to apply their knowledge in new scenarios (both in examinations and in their learning of the subject). This will hopefully give a better grounding of the subject but also enable candidates of all abilities to perform to their full potential in the examination.

Comments on specific questions

Question 1

(a) Candidates need to improve on their examination technique, especially reading the instruction carefully in the question. The question asked for advantages and disadvantages *when compared to other methods*, and whereas most candidates seemed to know what was involved in this method, unfortunately the direction in the rubric was missed by many candidates who wrote generic, often descriptive, statements about the method without any implied or direct comparison. Those who did attempt a comparison needed to further improve on their answer at this level. To obtain more of the available marks, candidates at A level need to expand on statements such as 'it is faster' or 'it is cheaper'.

(b) Some candidates penalised themselves by writing "install hardware and software" all on one line and then including a third incorrect answer. Some candidates wrote solely in terms of installing a new application on a PC, so missed many of the possible marking points. While it is accepted that the experience of most of the candidates will solely be with personal computers, at this level candidates are expected to have an appreciation of other types of computing installations.

(c) The candidates read the words 'requirements specification' and immediately linked them to hardware and software requirements. This meant that most answers went along the lines of 'the requirements specification is the specification of the hardware and software requirements'. A few did try to describe it correctly, but wrote about 'objectives'. Another examination technique that needs to be further developed is to avoid repeating the question in the answer. This is because some candidates repeated the stem of the question giving answers such as 'the requirements specification is the specification of the requirements'. Unfortunately, such answers are not creditworthy.

Candidates needed to improve their knowledge about design specifications.

Question 2

(a) This was fairly well answered. The most common errors were to choose *Boolean* for "Category A, B, C" data type and *date/time* for the "Running time" field.

(b) Many candidates seemed able to identify features that would be required by a visually impaired user. Voice synthesis and voice recognition were the most popular correct answers. There were quite a number of answers about not using red and/or green colours, this is really only an issue with colour-blind users. Other imprecise answers included braille monitors and the use of speakers to help deaf people. These answers were not creditworthy.

Question 3

Some very high marks were seen on this question. However, there was some confusion between the stack and the queue, and broadband was connected to different definitions. Candidates need to improve on their examination technique, especially reading the instruction carefully in the question. Some candidates did not join up all eight terms to the descriptions and therefore did not gain all the marks available.

Question 4

(a) This part was generally well answered. Some candidates need to improve on their knowledge of the difference between magnetic and optical media. Some candidates did not access the full marks available as they wrote incomplete answers such as 'tape' or 'disk' without further explanation. At this level, it is essential to be more precise with answers and give *magnetic tape* or *hard disk*, for example. One of the most common errors for solid state was to state the answer "USB", this is the name of the port on the computer/tablet and not the name of a device.

(b) Candidates needed to further develop their understanding of this question. The question asked for examples of the *use* of the different types of *storage media*. Some candidates gave answers such as 'a hard disk stores data'. This answer needs more detail since any storage medium stores data. Other answers which were incorrect included using memory sticks to store videos/movies and using CD/DVD-ROM to back-up data. Candidates needed to improve on their knowledge of ROM.

(c) Candidates needed to further develop their knowledge for this question. Candidates needed to refer to greater durability since optical media are not affected by magnetic fields.

Question 5

(a) This part of the question was a conversion of binary to denary. When answering questions like this, candidates should have realised that answers > 24 for location A or answers > 60 for location B were clearly incorrect and needed to be checked.

(b) This was the reverse of part **(a)** and required conversion from denary to binary. A quick check to see if their answers converted back to the original denary answers would have been worthwhile and would have allowed candidates access to both the available marks.

(c) Candidates needed to provide more precise answers for this question to gain both the available marks. Some candidates did understand that comparisons between the locations were required and these usually achieved both marks. Some candidates gained one mark for mentioning the microprocessor sending a signal to sound the clock alarm.

(d) Candidates need to further develop their understanding that the sensors simply record the value and that this data is sent continuously to the microprocessor. Many candidates still seem to believe that it is the sensors that make the decision to carry out some action. Many also believe that sensors only take readings when something changes, for example, in this question it was common to read "When it gets dark, the sensors send signals …". Many candidates did not realise that the back lighting would be lowered when it got dark and suggested it occurred the other way around (this clue was given in the stem of the question).

(e) Candidates need to improve on their examination technique, especially reading the instruction carefully in the question. A large number of candidates appeared to have missed that this question was still part of question 5 and so the microprocessor referred to was that of the alarm clock. This unfortunately resulted in some very correct generic answers for the contents of ROM and RAM, but which gained no marks in this context.

Question 6

(a) Candidates needed to be better prepared for this question. Candidates who realised that using pseudocode was a good way to describe the data structure performed well. This comment was true of both parts.

(b) Many candidates who had done badly in part **(a)** managed to gain two marks here for the FOR counter = 1 TO 8 construct. The better candidates managed to give the correct syntax for looking up the two-dimensional array and printing the result in the one-dimensional array.

Question 7

(a) Candidates needed to be better prepared for this question. However, there were still a significant number who gained full marks.

(b) Usually if part **(a)** was right, then part **(i)** here was also correct.

In part **(ii)**, a small number of candidates correctly identified the binary patterns for the coloured lights, but seemed not to understand that the leftmost five bits represented the time in tenths of a second. So, some had left these bits blank and others had generated imprecise five-bit patterns.

Question 8

(a) Apart from the candidates who got the two methods completely the wrong way round, it was generally answered well and most candidates gained two of the four marks. There is a point to note here about examination technique also, in that candidates cannot expect to be credited twice for opposite answers. So stating that the packets in circuit switching all travel by a single route and that in packet switching each packet travels by a different route is only worth one mark (not two).

In part **(ii)**, this seemed to be a 50:50 chance of getting it right. There seemed little evidence that answers given in part **(i)** or part **(iii)** led the candidate to choose the correct method of packet switching.

The responses to this question needed further development. Some candidates gained some marks for mentioning the issue of lagging and dropped packets. In general, it was the expected answers about VoIP being free because there is no need to pay the telephone company. An occasional mention of video calls, but usually the answers were based around cost and the availability of hardware and an internet connection.

(b) Candidates needed to develop their knowledge for this question. Candidates were able to transfer their normal drawings of a star network to the arrangement given in the question. This unfortunately indicated too many candidates simply remembering a diagram from a book. They needed to understand how a star network is actually arranged in real life.

The second part was better answered since this was pure book work.

Question 9

(a) Candidates needed to be better prepared for this question. The following answers were seen: “the knowledge base contains all the knowledge”, “the rules base contains all the rules” and “the inference engine makes decisions”. All these type of answers were too imprecise at this level to be creditworthy.

(b) Candidates need to improve on their examination technique, especially reading the instruction carefully in the question. Many candidates read the two words ‘features’ and ‘interface’ and decided to give features such as: descriptions of colour contrast, layout of text, font size and so on (unfortunately these were answers from previous mark schemes that asked a completely different question about interfaces). The occasional mark was awarded for use of a touch screen and/or an output screen to display results.

(c) This question was answered well. Many candidates gained the full marks available here

COMPUTING

Paper 9691/13
Written Paper

General comments

The standard of candidates' work was an improvement on last year in many areas.

The format of the examination was similar to November 2013. The candidates seemed better prepared for this new style of paper than they were twelve months ago. The new format leads candidates and Centres gradually into the revised syllabus which has its first sitting in June 2015.

Candidates need to understand that it is no longer the case that simply learning sections from text books will score many marks on future papers. There is a need to understand how to apply their knowledge in new scenarios (both in examinations and in their learning of the subject). This will hopefully give a better grounding of the subject but also enable students of all abilities to perform to their full potential in the examination.

Comments on Specific Questions

Question 1

In part **(i)**, many candidates answered this question in terms of the fetch execute cycle and so gained the mark. However, there were imprecise answers about the control unit controlling things which was not creditworthy.

Again, in part **(ii)**, candidates who mentioned either ROM or RAM gained the mark, but there were also many imprecise answers about just storing data.

In the final part, many candidates did well with some good descriptions of the function of the ALU given.

Question 2

(a) In part **(i)**, most candidates realised that the system would be out of date and so got one mark; but

as with the first part of the question, most candidates realised that the system would need to be updated and so gained one mark. Some candidates needed to expand on this for the second mark.

(b) This question was essentially book work. Whilst many candidates had clearly learned the definitions of adaptive, corrective and perfective maintenance, a number of candidates unfortunately confused this with implementation and described questionnaires, interviews and observation.

Question 3

(a) This question was answered better than similar questions in earlier papers. The idea of analogue and digital now appears to be well known, but there needs to be further understanding of what is meant by the terms "unidirectional" and "bidirectional". Also, a small number of candidates described serial and parallel transmission rather than baseband and broadband.

(b) The links between the terms and the definitions were correct in all but a very small number of cases. The links between the definitions and the examples needed better understanding. The right hand side of the diagram required more application than just pure knowledge.

Question 4

(a) Candidates scored well. Most candidates were able to describe what was meant by a computer virus in enough detail to gain the single mark.

(b) Candidates need to improve on their examination technique, especially reading the instruction carefully in the question. Many candidates imprecisely listed items such as anti-virus, firewall etc. and did not describe or give further detail. Where it is necessary to give a description, simple one word answers are not creditworthy. It is important that candidates read the stem of questions to make sure their answer is at the required depth.

Question 5

(a) Most candidates gave the correct value.

(b) In part **(i)**, this question required the answer *2 dimensional array.* Many candidates gained a mark for "array" but did not indicate the array dimensions.

In part **(ii)**, the full range of marks (0 to 4) were seen.

Question 6

(a) This question was very well answered with many candidates gaining the maximum mark. Candidates need to improve on their drawings of logic gates. It was helpful where a number of candidates had written the gate name inside the shape which made it easier to understand. Without the words, in many cases, it would have been almost impossible to interpret which gate had been drawn. Candidates should practice drawing gates so that it is clear to an Examiner what answer they are giving.

(b) Again many candidates gained the full marks here.

Question 7

(a) For measuring the noise level a sound sensor was a popular and correct answer. Some candidates however needed to improve their knowledge for this question as they gave the incorrect answer of decibel meter.

Candidates needed to develop their knowledge for this question. For measuring air pollution levels there were a large number of different incorrect answers. Those candidates who had carefully read the question stated the correct answers such as carbon dioxide sensor or nitrogen oxide sensor.

(b) Candidates needed to be better prepared for this question and may have needed to read through it carefully to ensure that they understood it. Some candidates appeared to misunderstand the scenario and seemed to expect that the monitoring stations would be connected back to base by all sorts of cabling etc. Candidates needed to develop their understanding that the monitoring needed to be continuous; some wrote that the solid state memory could be removed and taken back to base to be read, and admittedly this could be done, but only if the memory is replaced with a second one, so disrupting readings only for a very short time. Candidates should have mentioned the use of a portable device.

(c) Generally most candidates appreciated that using a spreadsheet was the best software to use. In some cases, expansions on how a spreadsheet could be used could have been clearer and in more detail to obtain all the marks available. Some candidates did mention graphs and trends, but in order for these responses to be creditworthy, the candidates needed to develop their answer rather than just taking the wording from the question stem.

Question 8

Parts **(i)**, **(ii)**, **(iii)** and **(v)** were very well answered; part **(iv)** however was the exception. There was much confusion between a WAN and a WLAN, so the most common answer was that LANs can also operate over WiFi and are then called WANs. Some candidates wrote about LANs covering a small area and WANs covering a large area which did not answer the question.

Question 9

(a) A large number of candidates correctly gave a touch screen as the suitable input device. Incorrect answers, such as mouse and keyboard, were also given which would not be regarded as suitable devices in this scenario. When selecting input and output devices, candidates should read the scenario very carefully to make sure their choice is appropriate.

(b) Many candidates did well in this question.

(c) Candidates needed to be better prepared for this question. Candidates doing project work should be aware of the type of testing necessary to ensure software, such as expert systems, operates correctly. Many candidates gave imprecise descriptions of normal and abnormal data or gave statements such as black box or white box testing without fully clarifying their response.

Question 10

(a) A good range of marks were seen here. The main errors were in the second statement and fifth statement. Many candidates needed to improve their knowledge of custom-written software and off-the-shelf software.

(b) This was fairly well answered with many candidates gaining full marks.

(c) Some very general answers such as user guide or how to use the software were stated here. Some candidates needed to develop their understanding of user documentation and technical documentation.

COMPUTING

Paper 9691/21
Written Paper

Key Messages

To succeed in this paper it is essential that candidates have practical experience of programming using a high-level procedural language. It is recommended that candidates choose one of the following: Pascal, Visual Basic (console mode), Python.

Programming and pseudocode questions from past examination papers provide an ideal starting point for practical work.

General Comments

There are a significant number of candidates who do not appear to have any programming experience.

Some questions require an answer using a high-level programming language. Although minor discrepancies in syntax are ignored, candidates need to understand that giving pseudocode answers will not gain credit.

Some questions ask for pseudocode solutions. Candidates giving answers using a real high-level programming language will be given credit for correct solutions.

Comments on Specific Questions

Question 1

(a) Candidates were required to declare a 1D array to store counts. Candidates need to improve their understanding of declaring integer arrays. Some candidates gave correct answers using a programming language. This was given credit. Common errors were not giving the dimension and/or the data type. A string data type was not appropriate here. The following is a suitable pseudocode declaration:

```
DECLARE Tally(4): INTEGER
```

Those candidates who chose to use the array element with index 0 and therefore wrote something like

```
DECLARE Tally(3): INTEGER
```

were given full credit, although it made later questions slightly more complex.

(b) (i) Candidates needed to develop their understanding of this question as few candidates seemed to understand the suggested modularisation of the initial pseudocode.

```
Cricket ← 0                                          }
Football ← 0                                         }
Tennis ← 0                                           }   InitialiseArrayTotals
Swimming ← 0                                         }
REPEAT                                               }
  INPUT Choice                                       }
  CASE Choice OF                                     }
    1: Cricket ← Cricket + 1                         }
    2: Football ← Football + 1                       }   InputStudentChoices
    3: Tennis ← Tennis + 1                           }
    4: Swimming ← Swimming + 1                       }
  ENDCASE                                            }
UNTIL Choice = 0                                     }
OUTPUT "Cricket", Cricket                            }
OUTPUT "Football", Football                          }
OUTPUT "Tennis", Tennis                              }   OutputTallyChart
OUTPUT "Swimming", Swimming                          }
```

Better candidates were able to write the more concise statements for the InitialiseArrayCounts procedure:

```
DECLARE Index: INTEGER
FOR Index ← 1 TO 4
  Tally(Index) ← 0
```

Few candidates appeared to appreciate that the CASE statement could be exchanged for a single statement:

```
Tally(Choice) ← Tally(Choice) + 1
```

Candidates who used indexes from 0 to 3 needed to make an adjustment:

```
Tally(Choice-1) ← Tally(Choice-1) + 1
```

(ii) Most candidates noted that the variable Football was replaced by Tally(2). For those candidates who used indexes from 0 to 3, it was the variable Tennis.

(c) Some excellent answers were seen for this part question. However, some candidates restricted their answers to work only to produce output as in the given example. The pseudocode should work for any values held in Tally. Candidates needed to understand the need for a new line to be output at the end of the OutputTally procedure.

(d) This part question required candidates to choose relevant test data. Candidates needed to give more precise answers to this question with reference to what was being tested. The OutputTally procedure takes a parameter. This parameter value was generated from incrementing the count in the Tally array. Therefore erroneous data is not appropriate. Normal data would be a value greater than 0, as this would check that the correct number of bars were output. Borderline data would be the value 0, as this would check that the procedure outputs an empty line without any bars. Extreme data would be a very large integer, where the number of bars does not fit on one line and tests are required that output is still readable.

(e) (i) This was mostly answered very well.

(ii) The more able candidates noticed that local variables were not used. Many candidates gave a feature which was clearly appropriate as a response to **part (i)**.

Question 2

(a) Many candidates completed this flowchart correctly. Common errors were to swap statement numbers 1 and 8, 8 and 9, or 7 and 9. Candidates should take some guidance from the shapes. Input/Output shapes are parallelograms, process shapes are rectangles and decision shapes are diamonds.

(b) Many candidates made a good attempt at completing the pseudocode for the FindPassword function. Some did not give a data type in the function header. This is to show what type of value the function is to return. The majority of candidates got one of the conditions for the WHILE loop, but very few got both: WHILE NOT EndOfFile() AND NOT Found

Many candidates compared ThisUserID with the whole record that was read in. It should only be compared with the UserID field of that record. Many candidates output string values rather than returned them. It is not appropriate to output a message to the user from this function.

The statement should read:

```
IF Found = TRUE
  THEN
    RETURN EncryptedPassword
  ELSE
    RETURN ErrorCode
```

Candidates need to understand the difference between the = symbol (equivalence) and the ← symbol (assignment) in pseudocode. Although some programming languages use the = symbol for both of these concepts, the distinction between these concepts is fundamental.

Question 3

(a) Many screen designs were well thought out. Essential to any screen design is the inclusion of a heading, so any user can see what they are working on. Good design should minimise the amount of typing by the user, as this is likely to introduce errors. Clear guidance should be given to the user as to what is expected. For example, it is essential to state the units (cm) next to the fields where the user is expected to enter measurements. Appropriate space for results and command buttons to calculate were also required. Candidates needed to explain clearly how the data would be entered. Creditworthy responses needed more than just re-iterating the words in the stem of the question.

(b) Many candidates gave the correct logic expression. Some candidates need to improve on their understanding of the symbols used in computing as a common error was that candidates used the mathematical symbol ≥ rather than the computing operator >=. Another common error was that candidates did not use the given variable RoomWidth.

(c) Most candidates knew that DIV gives the integer part of division. Few candidates knew how this is implemented in their chosen programming language. Candidates should have experience of writing programs using integer division. Pascal uses DIV, VB uses \ and Python uses //.

Although the question said that the number of tiles needs to be rounded up if the room measurement is not an exact multiple of the tile measurement, few candidates were able to program this. A hint was given in pseudocode:

```
IF RoomWidth MOD 30 > 0 THEN TilesForWidth ← TilesForWidth + 1
```

Question 4

(a) The majority of candidates chose 0 and 1 to represent white and coloured tiles. Many other candidates chose characters such as “W” and “C”. Candidates need to understand that W and C (without quotes) were not acceptable as these are not character values. Few candidates successfully wrote code to initialise every element of the 2D array. In this case, candidates needed to be familiar with a nested loop construct.

(b) Candidates needed to be better prepared for this question. Those unfamiliar with nested loops wrote weak solutions. Some candidates also needed to improve on their flowchart answers as they did not access the full marks available. Some candidates forgot to initialise the running totals for counting the number of white and coloured tiles.

Question 5

(a) The majority of candidates made a good attempt at dry-running the algorithm. Many candidates missed writing down the results of the last iteration. A common mistake was to write 0 or 1 instead of TRUE or FALSE in the column headed a>= x.

(b) Very few candidates noticed that the algorithm was the standard method of converting a denary integer into binary even though many candidates got the correct output from the procedure.

COMPUTING

Paper 9691/22

Written Paper

Key Messages

To succeed in this paper it is essential that candidates have practical experience of programming using a high-level procedural language. It is recommended that candidates choose one of the following: Pascal, Visual Basic (console mode), Python.

Programming and pseudocode questions from past examination papers provide an ideal starting point for practical work.

General Comments

There are a significant number of candidates who do not appear to have any programming experience.

Some questions require an answer using a high-level programming language. Although minor discrepancies in syntax are ignored, candidates need to understand that giving pseudocode answers will not gain credit.

Some questions ask for pseudocode solutions. Candidates giving answers using a real high-level programming language will be given credit for correct solutions.

Comments on Specific Questions

Question 1

(a) Many screen designs were well thought out. Good design should minimise the amount of typing by the user, as this is likely to introduce errors. The question clearly lists the information required to be gathered from the competitor. Suitable screen designs needed to provide a text box to type the name. The age to be entered was between 8 and 18. A drop-down box was the most appropriate way to enter this data. Some candidates instead asked for date of birth or year of birth. Neither of these was in the specification. Sports club membership was either yes or no, so two radio buttons or one checkbox were appropriate here. Many candidates expected the user to type the letter code into a text box. This was not suitable. A list box, which also gave the event descriptions or a set of radio buttons, would be suitable methods here. Candidates needed to have also included a label or similar to display the fee and a button for the user to confirm that their details were correct.

(ii) Many candidates gave a description of their screen design rather than the required justification. The reason for using radio buttons is that it limits the user to just one option (appropriate for membership and choice of event). The reason for using a drop-down list is to limit the choices and minimise typing, which would reduce errors. Candidates need to understand that general comments about user friendliness and ease of use are not sufficient.

(b) This was generally done well. Candidates need to understand that ClubMember should be of Boolean data type, which uses 1 byte. EventEntered is just a single character, or a string of length 1. Candidates needed to have read the information in the question correctly as some did not appreciate that a single value should be given for each field size. Single value field sizes help with calculating the size of the file if 100 event entries are stored. Candidates need to understand that a clear layout of their working helps to gain credit for their method. A common error was to calculate the file size in KBytes when the question asked for the size in bytes.

Question 2

(a) The majority of candidates made a good attempt at dry-running the algorithm. A common mistake was to write 0 or 1 instead of TRUE or FALSE in the columns headed List(m) = s and List(m) > s. Candidates need to understand the importance of recording every change of variables or expressions at every step of the code "execution". Some candidates did not follow the instruction of starting a new row in the trace table every time round the loop. This made it difficult to see which values were held in the variables at any given point of program execution.

(b) Very few candidates noticed that the algorithm was the standard method of a binary search which output the position in the list of the search item.

Question 3

(a) (i) The majority of candidates chose 1 and 2 to represent white and black tokens. This was a suitable solution as long as the empty cells were also represented by an integer (usually 0). Many other candidates chose characters such as "W" and "B" for the tokens and "" for the empty cells. Candidates need to understand that W and B (without quotes) were not acceptable as these are not character values. A common error was to give the black and white tokens the same value.

(ii) Candidates needed to be better prepared for this question. Few candidates successfully wrote code to declare the 2D array and initialise every element. In this case, candidates needed to be familiar with a nested loop construct.

(iii) Only the better candidates seemed to be able to assign a value to a specified single cell of the grid.

(b) Candidates needed to be better prepared for this question. Those unfamiliar with nested loops wrote weak solutions. A very small minority of candidates were able to write a FOR loop counting down from a higher starting value to a smaller end value. Candidates appear to need a lot more practice at writing program code from a flowchart. The most popular programming language quoted for the answer to this question was VB. Using this language, a suitable solution would be:

```
For Row = 6 To 1 Step -1
  For Column = 1 To 7
    Console.Write(Grid(Row, Column))
  Next Column
  Console.WriteLine()
Next Row
```

(c) (i) Completing an algorithm that was presented as incomplete pseudocode statements was done fairly well. A common error was to use AND rather than OR in the Boolean expression to test if the column number was out of range. Candidates should check their answers carefully. It is impossible for x to be less than 1 and also greater than 7. Candidates also needed to have written the data types for the parameter and the return value in the function header.

(ii) Many candidates were able to choose the type of test data correctly and give suitable example values. A common error was to give a pair of co-ordinates rather than just a value for the column. The justifications given were usually descriptions of expected results rather than acceptable reasons for choice. More than just generic textbook answers were required here. Good responses included the reason for choosing a value as normal data because the column existed and still had an empty row. A reason for choosing 1 or 7 as a boundary value was because these were the first and last columns of the grid. Candidates need to understand that they need to make reference to the given scenario and give specific answers relevant to the code being tested.

(d) This part question required candidates to understand the problem set and the solution outlined in the question. Very few candidates could give the correct parameter required in line 03. Some candidates remembered the value they had chosen to represent an empty cell and completed line 05 correctly. Line 06 required the row number to be incremented. The better candidates realised that in line 08 they needed to test whether it was Player A or Player B's turn. A common error was to write:

IF NextPlayer = A rather than IF NextPlayer = "A".

The quotes are important. They show that these are character values. Lines 10 and 12 should assign the values to represent a white and a black token respectively. If candidates opted for character values they needed to be enclosed in quotes.

(e) (i) The better candidates stated that the parameter Player was passed by value and the parameter Number was passed by reference. Candidates need to understand that the random number generated within the procedure is passed to the calling program via the parameter Number and that is why this parameter must be passed by reference.

(ii) Candidates needed to be better prepared for this question. Few candidates could give the correct procedure call, even fewer were able to give the correct parameters required: GetColumn(NextPlayer, ChosenColumnNumber). A common mistake was to quote the procedure declaration header given in the question.

(f) This part was done well by most candidates.

COMPUTING

Paper 9691/23
Written Paper

Key Messages

To succeed in this paper it is essential that candidates have practical experience of programming using a high-level procedural language. It is recommended that candidates choose one of the following: Pascal, Visual Basic (console mode), Python.

Programming and pseudocode questions from past examination papers provide an ideal starting point for practical work.

General Comments

There are a significant number of candidates who do not appear to have any programming experience.

Some questions require an answer using a high-level programming language. Although minor discrepancies in syntax are ignored, candidates need to understand that giving pseudocode answers will not gain credit.

Some questions ask for pseudocode solutions. Candidates giving answers using a real high-level programming language will be given credit for correct solutions.

Comments on Specific Questions

Question 1

(a) Candidates were required to declare a 1D array to store counts. Candidates need to improve their understanding of declaring integer arrays. Some candidates gave correct answers using a programming language. This was given credit. Common errors were not giving the dimension and/or the data type. A string data type was not appropriate here. The following is a suitable pseudocode declaration:

```
DECLARE Tally(4): INTEGER
```

Those candidates who chose to use the array element with index 0 and therefore wrote something like

```
DECLARE Tally(3): INTEGER
```

were given full credit, although it made later questions slightly more complex.

(b) (i) Candidates needed to develop their understanding of this question as few candidates seemed to understand the suggested modularisation of the initial pseudocode.

```
Cricket ← 0                                          }
Football ← 0                                         }
Tennis ← 0                                           }   InitialiseArrayTotals
Swimming ← 0                                         }
REPEAT                                               }
  INPUT Choice                                       }
  CASE Choice OF                                     }
    1: Cricket ← Cricket + 1                         }
    2: Football ← Football + 1                       }   InputStudentChoices
    3: Tennis ← Tennis + 1                           }
    4: Swimming ← Swimming + 1                       }
  ENDCASE                                            }
UNTIL Choice = 0                                     }
OUTPUT "Cricket", Cricket                            }
OUTPUT "Football", Football                          }
OUTPUT "Tennis", Tennis                              }   OutputTallyChart
OUTPUT "Swimming", Swimming                          }
```

Better candidates were able to write the more concise statements for the InitialiseArrayCounts procedure:

```
DECLARE Index: INTEGER
FOR Index ← 1 TO 4
  Tally(Index) ← 0
```

Few candidates appeared to appreciate that the CASE statement could be exchanged for a single statement:

```
Tally(Choice) ← Tally(Choice) + 1
```

Candidates who used indexes from 0 to 3 needed to make an adjustment:

```
Tally(Choice-1) ← Tally(Choice-1) + 1
```

(ii) Most candidates noted that the variable Football was replaced by Tally(2). For those candidates who used indexes from 0 to 3, it was the variable Tennis.

(c) Some excellent answers were seen for this part question. However, some candidates restricted their answers to work only to produce output as in the given example. The pseudocode should work for any values held in Tally. Candidates needed to understand the need for a new line to be output at the end of the OutputTally procedure.

(d) This part question required candidates to choose relevant test data. Candidates needed to give more precise answers to this question with reference to what was being tested. The OutputTally procedure takes a parameter. This parameter value was generated from incrementing the count in the Tally array. Therefore erroneous data is not appropriate. Normal data would be a value greater than 0, as this would check that the correct number of bars were output. Borderline data would be the value 0, as this would check that the procedure outputs an empty line without any bars. Extreme data would be a very large integer, where the number of bars does not fit on one line and tests are required that output is still readable.

(e) (i) This was mostly answered very well.

(ii) The more able candidates noticed that local variables were not used. Many candidates gave a feature which was clearly appropriate as a response to **part (i)**.

Question 2

(a) Many candidates completed this flowchart correctly. Common errors were to swap statement numbers 1 and 8, 8 and 9, or 7 and 9. Candidates should take some guidance from the shapes. Input/Output shapes are parallelograms, process shapes are rectangles and decision shapes are diamonds.

(b) Many candidates made a good attempt at completing the pseudocode for the FindPassword function. Some did not give a data type in the function header. This is to show what type of value the function is to return. The majority of candidates got one of the conditions for the WHILE loop, but very few got both: WHILE NOT EndOfFile() AND NOT Found

Many candidates compared ThisUserID with the whole record that was read in. It should only be compared with the UserID field of that record. Many candidates output string values rather than returned them. It is not appropriate to output a message to the user from this function.

The statement should read:

```
IF Found = TRUE
  THEN
    RETURN EncryptedPassword
  ELSE
    RETURN ErrorCode
```

Candidates need to understand the difference between the = symbol (equivalence) and the ← symbol (assignment) in pseudocode. Although some programming languages use the = symbol for both of these concepts, the distinction between these concepts is fundamental.

Question 3

(a) Many screen designs were well thought out. Essential to any screen design is the inclusion of a heading, so any user can see what they are working on. Good design should minimise the amount of typing by the user, as this is likely to introduce errors. Clear guidance should be given to the user as to what is expected. For example, it is essential to state the units (cm) next to the fields where the user is expected to enter measurements. Appropriate space for results and command buttons to calculate were also required. Candidates needed to explain clearly how the data would be entered. Creditworthy responses needed more than just re-iterating the words in the stem of the question.

(b) Many candidates gave the correct logic expression. Some candidates need to improve on their understanding of the symbols used in computing as a common error was that candidates used the mathematical symbol ≥ rather than the computing operator >=. Another common error was that candidates did not use the given variable RoomWidth.

(c) Most candidates knew that DIV gives the integer part of division. Few candidates knew how this is implemented in their chosen programming language. Candidates should have experience of writing programs using integer division. Pascal uses DIV, VB uses \ and Python uses //.

Although the question said that the number of tiles needs to be rounded up if the room measurement is not an exact multiple of the tile measurement, few candidates were able to program this. A hint was given in pseudocode:

```
IF RoomWidth MOD 30 > 0 THEN TilesForWidth ← TilesForWidth + 1
```

Question 4

(a) The majority of candidates chose 0 and 1 to represent white and coloured tiles. Many other candidates chose characters such as “W” and “C”. Candidates need to understand that W and C (without quotes) were not acceptable as these are not character values. Few candidates successfully wrote code to initialise every element of the 2D array. In this case, candidates needed to be familiar with a nested loop construct.

(b) Candidates needed to be better prepared for this question. Those unfamiliar with nested loops wrote weak solutions. Some candidates also needed to improve on their flowchart answers as they did not access the full marks available. Some candidates forgot to initialise the running totals for counting the number of white and coloured tiles.

Question 5

(a) The majority of candidates made a good attempt at dry-running the algorithm. Many candidates missed writing down the results of the last iteration. A common mistake was to write 0 or 1 instead of TRUE or FALSE in the column headed a>= x.

(b) Very few candidates noticed that the algorithm was the standard method of converting a denary integer into binary even though many candidates got the correct output from the procedure.

COMPUTING

Paper 9691/31
Written Paper

General

There was evidence of an improvement in the standard of answers seen from some areas of the syllabus which had been highlighted in previous examiner reports as a weakness. This included the answers seen for **Question 3 (b)**. This was a question where only one of the marks for parts **(i)** and **(ii)** was awarded for the basic definition of Second and Third Normal Form. The majority of the marks were given for the candidate's ability to apply this understanding to the scenario given. For part **(b)** many candidates correctly identified the tables and were then able to suggest the solution to the poorly designed tables Sales and Painting.

Candidates need to improve on their theory knowledge; an example being **Question 2 (a)** for paged memory management.

There were examples in the scripts where examination technique needed to be further improved. In **Question 3**, to answer part **(b)**, some candidates referred to the tables with numbers which did not appear in the question rubric; for example, table Painting was referred to as '2'. For **Question 5(b)**, the key word in the stem of the question was 'explain'. To achieve more of the available four marks, candidates needed to expand on the description of the special purpose registers.

Another examination technique that needs to be further developed is to avoid repeating the question in the answer. For example, in **Question 7 (f)**, some candidates used wording such as *"A user-defined function is a function defined by the user"*.

Question 1

Many candidates gained the mark for part **(a)(i)**. Some candidates needed to improve on their knowledge to part **(ii)**. The expression in part **(b)(ii)** included a 'power of' operator and candidates needed to demonstrate further understanding of this. Credit was given for either an incomplete expression which was correctly formed with the power displayed correctly, or evidence that the candidate had understood that the power symbol was to be treated as an operator.

Many correct answers were seen for part **(c)(ii)** for the use of a stack to evaluate a reverse Polish expression. Candidates were expected to evaluate the intermediate contents of the stack, but many answers did not include this.

Question 2

See comments in the General section about part **(a)**. In order to access more of the available marks, candidates needed to further improve on their understanding of this question. Three of the following ideas would have secured the marks:

- that the memory is divided into 'page frames' and the program code divided into 'pages' of the same size
- not all of the pages of the program are concurrently loaded into main memory
- the page frame table keeps track of which pages are loaded into which page frames
- there will be occasions when a new page or pages must be loaded into memory, and then will require other loaded pages to be swapped out

For part **(b)(i)** the most common answers suggested some form of priority allocation to all processes. Answers which attempted to give a particular priority order were, however, expressed with an imprecise statement such as 'shortest job first', which did not gain credit.

Candidates needed to improve on their answers to part **(c)**.

Question 3

See earlier comments in the General section which would suggest a clear improvement in the understanding of certain aspects of this syllabus content, namely 'normalisation'. For part **(a)(ii)** candidates needed to further develop their knowledge of entity-relationship diagrams. Weak answers included labelling entity boxes with the name of an attribute from each of the three entities (and not the entity name). Candidates scored with either a 1-to-many relationship line between Customer and Sales, Painting and Sales, or the many-to-many relationship between Customer and Painting.

In part **(b)(ii)**, candidates showed good understanding of Third Normal Form. Most candidates gained credit in part **(ii)** with the introduction of a new table Artist with primary key artist name, and then the retention of the artist name in the Painting table to act as a foreign key.

Candidates needed to develop their understanding for part **(c)**, as the first part of the command was to 'UPDATE Customer'.

Question 4

For part **(a)** the better responses included drawing lines to show the transfer of data values for (say) a memory location to the Accumulator and/or circled the key memory location contents and/or addresses which were to be used. This was what was expected from the question stem directive to "Write on the diagram to explain". Some answers which gave a written explanation were given credit if correct.

Candidates needed to develop their understanding for part **(b)(i)** as very few correct answers of 256 were seen. Some answers gave 255, perhaps not understanding that zero could be an instruction code. For part **(iii)**, candidates needed to improve their knowledge of why hexadecimal is used in computing.

Parts **(b)(v)** and **(vi)** were well answered with candidates referring to the given table and using the binary machine code for the two required instructions. These questions proved to be a good discriminator with the full range of marks seen.

Most candidates were able to score some of the available five marks for part **(c)**.

Question 5

Part **(a)** was answered well, with candidates including many different points to secure the two marks. For part **(b)**, see the earlier comments made in the General section.

Candidates needed to improve their knowledge for part **(c)**. The answer expected was the control bus followed by one of the signals it carried. The most common correct answer seen was an 'interrupt' signal.

Answers seen for part **(d)** were varied. For part **(i)**, candidates needed to understand that this would be a 'Case 1' followed by an explanation that the number to be used was already held in the Accumulator (and so no further access to memory was required).

Question 6

Candidates needed to further develop their understanding of keyword and symbol tables.

Candidates who understood the translation process had no problem finding from the given program code three language keywords (to answer part **(ii)** and three identifier names to correctly answer part **(iv)**).

For part **(b)**, many candidates appreciated that optimisation meant the code would execute faster. Candidates who provided an imprecise explanation, for example, “code is quicker”, did not gain full marks. Candidates usually stated correctly for the second mark that it would take up less main memory space.

Question 7

Parts **(a)** through to **(f)** were generally well answered.

See earlier comment in the General section for part **(f)**. Answers contrasting user-defined and built-in functions needed further development. Some answers followed the suggestion in the question stem to illustrate their understanding of a built-in function by referring to one they had used in their practical programming. Candidates need to learn about the coding of functions in their practical high-level language programming.

COMPUTING

Paper 9691/32
Written Paper

General

There was a marked improvement in the standard of answers seen from some areas of the syllabus which had been highlighted in previous examiner reports as a weakness. This was especially true of the answers seen for question **3 (b)**. This was a question where only one of the marks for parts **(i)** and **(ii)** was awarded for the basic definition of Second and Third Normal Form. The majority of the marks were given for the candidate's ability to apply this understanding to the scenario given. Many candidates correctly identified the tables and were able to suggest the solution to the poorly designed tables RaceRunner and Race.

Candidates need to develop their understanding of some topics; an example being the scenario given in question **2 (a)(i)** for paged memory management.

There were examples in the scripts where examination technique needed to be further improved. In **Question 3**, to answer part **(b)**, some candidates referred to the tables with numbers which did not appear in the question rubric; for example, table RaceRunner was referred to as '3'. For **Question 6(a)**, the key word in the stem of the question was 'explain'. To achieve more of the available four marks, candidates needed to expand on the description of the special purpose registers.

Another examination technique that needs to be further developed is to avoid repeating the question in the answer. For example, in **Question 7 (f)**, some candidates used wording such as *"A user-defined function is a function defined by the user"*.

Question 1

Many candidates gained the mark for part **(a)(i)** but could not repeat this for the expression given in part **(ii)**. Candidates needed to be better prepared for the expression given in part **(b)(ii)** which included two 'power of' operators. Credit was given for either an incomplete expression which was correctly formed with the power displayed correctly, or evidence that the candidate had understood the power symbol was to be treated as an operator.

For part **(c)** correct descriptions for the operation of a stack and a queue were usually forthcoming for the two marks. Candidates needed to be able to further expand on this and give an application where they are used. Candidates sometimes did give an example of stack like behaviour, for example the use of the 'undo' feature in software. In order to obtain the available marks, candidates needed to understand the key words in the question stem which required an example of where these data structures are used in the operation of a 'computer system'. Some candidates provided correct answers such as, (for a stack) the storage and retrieval of return address when procedure calling in a high level language, (for a queue) the spooling of print jobs or the queuing of jobs by the high-level scheduler in a multiprogramming operating system. Candidates need to be aware where these basic data structures have a practical application in a computer system or high-level language programming.

For part **(d)** the majority of candidates scored the mark by either stating "the first item to join the stack will be the last to leave" – or more commonly, "the last item to join will be the first to leave".

Part **(e)** was well answered. Some candidates were unable to secure the final mark, they needed to be able to describe that the sequence of operations has reversed the order of the data items stored on Stack-A.

Question 2

See comments in the General section about part **(a)**. The following three ideas would have secured the marks; that a page will be 'swapped out' from memory – the page containing the required program code must be 'swapped in' – and the management of pages of the program is stored in a page management table.

For part **(b)(i)** the most common answer was to give each process a 'time-slice' and then operate a round-robin allocation. Answers which attempted to give a particular priority order needed to be well expressed. A statement such as 'shortest job first' did not gain credit as it was not clear what the word shortest referred to.

Part **(c)** was presented to the candidate using a different question framework to previous papers with the intention of making the understanding clearer to the candidate. Answers securing full marks were achieved by the better candidates.

Question 3

See earlier comments in the General section which would suggest a clear improvement in the understanding of certain aspects of this syllabus content. For part **(a)(ii)** a not uncommon error was for the candidate to label entity boxes with the name of an attribute from each of the three entities (not the entity name). Candidates scored with either 1-to-many relationship lines between Race and RaceRunner, Runner and RaceRunner, or the many-to-many relationship between Race and Runner.

For part **(b)** better answers were seen than on previous papers where this topic has been examined. Weaker candidates usually scored one or two marks by solving the 3NF issue, a separate Club table would be part of the design. A common error was to include a ClubID when the question clearly stated that the club names were unique.

For part **(c)(i)** the query was well answered with many candidates gaining the full three marks. Answers for part **(ii)** suggested that candidates need to develop their understanding of how a DDL is used to update, amend or delete a record.

Question 4

Candidates needed to be better prepared for this question as few correct answers of 256 were seen. Some answers gave 255; candidates needed to improve their understanding and appreciate that zero could be a memory address. Candidates needed to explain for part **(iii)** why hexadecimal is used extensively in computing. Part **(a)(vi)** was well understood by the more able candidates who gave the IN and OUTCH instructions as examples to support the true statement.

Part **(b)** proved to be a good discriminator with the full range of marks seen.

Question 5

There were several statements the candidates could make to gain the two marks. The most popular answers seen were that both translators product executable code or an object file and both will identify errors in the source code.

For part **(b)**, candidates could gain the full five marks for stating sequence F, H, B, C and A. Thereafter a single mark credit was given for; realising that D was the statement which was not required, C would be followed by A and that H would be followed by B.

Question 6

See earlier comments in the General section for **part (a).**
For **part (b)** candidates were able to name the control bus and the most common example given for a control signal was an interrupt.

Answers seen for **part (c)** were varied. For **part (i)** candidates correctly stated 'Case 2'. Candidates needed to follow this with a statement which made reference to the 78 value being an address. An answer which only used the wording in the question stem *"No further use of the address bus"* did not gain credit.

Better answers were seen for **part (ii)**. Candidates appreciated that incrementing the Accumulator contents did not require the retrieval of more data from main memory and hence no use of the address bus.

Question 7

Candidates needed to improve their understanding for question parts **(a)** to **(e)**. The question framework had been used on several past examination papers. The only new feature of this function definition was that it contained an optional parameter. Many candidates scored for parts **(a)** and **(d)** but other answers were incorrect. Few candidates recognised in part **(c)** that the 'Dr' character sequence not only was the title at the start of the string, but would also be removed from the start of the surname.

See earlier comment in the General section for part **(f)**. Answers contrasting user-defined and built-in functions needed to expand on the key differences. Few answers followed the suggestion in the question stem to illustrate their understanding of a built-in function by referring to ones they had used in their practical programming. Candidates need to use built-in functions in their practical programming; in particular, functions available for string handling, number formatting and a random number generator.

COMPUTING

Paper 9691/33
Written Paper

General

There was evidence of an improvement in the standard of answers seen from some areas of the syllabus which had been highlighted in previous examiner reports as a weakness. This included the answers seen for **Question 3 (b)**. This was a question where only one of the marks for parts **(i)** and **(ii)** was awarded for the basic definition of Second and Third Normal Form. The majority of the marks were given for the candidate's ability to apply this understanding to the scenario given. For part **(b)** many candidates correctly identified the tables and were then able to suggest the solution to the poorly designed tables Sales and Painting.

Candidates need to improve on their theory knowledge; an example being **Question 2 (a)** for paged memory management.

There were examples in the scripts where examination technique needed to be further improved. In **Question 3**, to answer part **(b)**, some candidates referred to the tables with numbers which did not appear in the question rubric; for example, table Painting was referred to as '2'. For **Question 5(b)**, the key word in the stem of the question was 'explain'. To achieve more of the available four marks, candidates needed to expand on the description of the special purpose registers.

Another examination technique that needs to be further developed is to avoid repeating the question in the answer. For example, in **Question 7 (f)**, some candidates used wording such as *"A user-defined function is a function defined by the user"*.

Question 1

Many candidates gained the mark for part **(a)(i)**. Some candidates needed to improve on their knowledge to part **(ii)**. The expression in part **(b)(ii)** included a 'power of' operator and candidates needed to demonstrate further understanding of this. Credit was given for either an incomplete expression which was correctly formed with the power displayed correctly, or evidence that the candidate had understood that the power symbol was to be treated as an operator.

Many correct answers were seen for part **(c)(ii)** for the use of a stack to evaluate a reverse Polish expression. Candidates were expected to evaluate the intermediate contents of the stack, but many answers did not include this.

Question 2

See comments in the General section about part **(a)**. In order to access more of the available marks, candidates needed to further improve on their understanding of this question. Three of the following ideas would have secured the marks:

- that the memory is divided into 'page frames' and the program code divided into 'pages' of the same size
- not all of the pages of the program are concurrently loaded into main memory
- the page frame table keeps track of which pages are loaded into which page frames
- there will be occasions when a new page or pages must be loaded into memory, and then will require other loaded pages to be swapped out

For part **(b)(i)** the most common answers suggested some form of priority allocation to all processes. Answers which attempted to give a particular priority order were, however, expressed with an imprecise statement such as 'shortest job first', which did not gain credit.

Candidates needed to improve on their answers to part **(c)**.

Question 3

See earlier comments in the General section which would suggest a clear improvement in the understanding of certain aspects of this syllabus content, namely 'normalisation'. For part **(a)(ii)** candidates needed to further develop their knowledge of entity-relationship diagrams. Weak answers included labelling entity boxes with the name of an attribute from each of the three entities (and not the entity name). Candidates scored with either a 1-to-many relationship line between Customer and Sales, Painting and Sales, or the many-to-many relationship between Customer and Painting.

In part **(b)(ii)**, candidates showed good understanding of Third Normal Form. Most candidates gained credit in part **(ii)** with the introduction of a new table Artist with primary key artist name, and then the retention of the artist name in the Painting table to act as a foreign key.

Candidates needed to develop their understanding for part **(c)**, as the first part of the command was to 'UPDATE Customer'.

Question 4

For part **(a)** the better responses included drawing lines to show the transfer of data values for (say) a memory location to the Accumulator and/or circled the key memory location contents and/or addresses which were to be used. This was what was expected from the question stem directive to "Write on the diagram to explain". Some answers which gave a written explanation were given credit if correct.

Candidates needed to develop their understanding for part **(b)(i)** as very few correct answers of 256 were seen. Some answers gave 255, perhaps not understanding that zero could be an instruction code. For part **(iii)**, candidates needed to improve their knowledge of why hexadecimal is used in computing.

Parts **(b)(v)** and **(vi)** were well answered with candidates referring to the given table and using the binary machine code for the two required instructions. These questions proved to be a good discriminator with the full range of marks seen.

Most candidates were able to score some of the available five marks for part **(c)**.

Question 5

Part **(a)** was answered well, with candidates including many different points to secure the two marks. For part **(b)**, see the earlier comments made in the General section.

Candidates needed to improve their knowledge for part **(c)**. The answer expected was the control bus followed by one of the signals it carried. The most common correct answer seen was an 'interrupt' signal.

Answers seen for part **(d)** were varied. For part **(i)**, candidates needed to understand that this would be a 'Case 1' followed by an explanation that the number to be used was already held in the Accumulator (and so no further access to memory was required).

Question 6

Candidates needed to further develop their understanding of keyword and symbol tables.

Candidates who understood the translation process had no problem finding from the given program code three language keywords (to answer part **(ii)** and three identifier names to correctly answer part **(iv)**).

For part **(b)**, many candidates appreciated that optimisation meant the code would execute faster. Candidates who provided an imprecise explanation, for example, “code is quicker”, did not gain full marks. Candidates usually stated correctly for the second mark that it would take up less main memory space.

Question 7

Parts **(a)** through to **(f)** were generally well answered.

See earlier comment in the General section for part **(f)**. Answers contrasting user-defined and built-in functions needed further development. Some answers followed the suggestion in the question stem to illustrate their understanding of a built-in function by referring to one they had used in their practical programming. Candidates need to learn about the coding of functions in their practical high-level language programming.

COMPUTING

Paper 9691/04
Project

General comments

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

The projects submitted covered a wide variety of topics with better candidates showing evidence of researching a problem beyond their School or college life.

In order to have the full range of marks available to the candidate, the computing project must involve a third party client whose requirements are considered and clearly documented at all stages of the system development. Centres are reminded that the project work is designed to test the candidates' understanding of the systems life cycle. The requirements are clearly set out in syllabus ***section 4***, 'The Guidance on Marking the Computing Project' ***section 7.2*** acts as a useful checklist, for teachers and candidates, setting out the expected contents of each section.

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

Project Reports and Presentation

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

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

Project assessment and marking

Nearly all Centres used the marking grid on pages 45-48 of the current syllabus to provide a breakdown of marks showing the marks given for each sub-section of the report. In order to aid the process of moderation, the completed grid should include references to the appropriate pages in the candidates' reports where evidence for each section can be found. Teachers should comment as to why they awarded the marks for each section. Moderators have noticed that where there is a good commentary provided by a teacher the marking is usually very close to the agreed standard.

Section 3

Comments on Individual Sections

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

(a) Quality of report.

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

(b) Definition Investigation and Analysis

(i) Definition - nature of the problem

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

(ii) Investigation and Analysis

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

The detailed requirements specification produced must be based on the information collected and include what the client needs the system to produce.

(c) Design

(i) Nature of the solution

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

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

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

(ii) Intended benefits

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

(iii) Limits of the scope of solution

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

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

(d) Software Development, Programming Testing and Installation

(i) Development

Evidence of development should include program listings of code written by the candidate, data structures used and evidence of tailoring of software packages. For top marks, the solution should have no logical flaws, match the design specification in **(c)(i)** and be annotated by the candidate.

(ii) Programming

It is important that the programming code in this sub-section is written by the candidate and not produced as a result of tailoring a software package. Marks should only be awarded to code that has been written by the candidate.

Candidates need to show that they can apply the programming skills developed at AS level in Paper 2 to a real situation. This includes technical programming competence and ensuring that their program could be maintained by writing self-documented code.

(iii) Testing

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

(iv) Installation

Most candidates provided an implementation plan containing details of user testing, user training and system changeover.

For good marks to be awarded, written evidence from the client and/or user(s) must be included in order to show that the system has been seen, used and tested, and the candidate's plans have been agreed.

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

(e) Documentation

(i) Systems Maintenance Documentation

This sub-section of the report is a Systems Maintenance document. Many candidates incorrectly included Technical Documentation. Please see the current syllabus for details of what should be included in this sub-section.

For top marks to be awarded, the candidate must explain how adaptive maintenance could be undertaken for their system.

(ii) User Guide

This section was completed to a good standard by most candidates. Centres are reminded that for full marks the candidate must include an index and a glossary for the terms used in their User Guide. This needs to be complete including details of how to install the new system, backup routines and a guide to common errors. Also good on-screen help should exist where this is a sensible option.

(f) Evaluation

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

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

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

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

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

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

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