COMPUTING

Paper 9691/11
Written Paper

Question 1

Most candidates did not answer the question that was asked. Most candidates appeared to ignore “expert systems” and simply talk about the systems life cycle, which was certainly not asked for. Other candidates appeared to concentrate on the idea of a car engine rather than the expert system.

Question 2

(a) Many candidates defined some of the types of memory rather than a use for each of the storage devices named.

(b) The advantages of an SSD device are not well known. Most candidates' responses were based on Flash drives.

Question 3

This question was generally answered very well. The majority of candidates gained full or nearly full credit.

Question 4

Many candidates did not enter appropriate named validation checks. Some candidates gave answers such as parity or checksum or thumbprint. Some candidates did not attempt to answer at all, even though they had been shown correct entries for each of the fields and were asked to simply give an example that would fail a validity check in the second column. They could have used the correct entries given and altered them so that they would NOT match the type of validation check they named.

Question 5

(a) (i) Most candidates gained full credit for this question. Where candidates did lose a mark, it was almost always for the incorrect mode of transmission.

(ii) Candidates did even better on this part, with most gaining full credit.

(iii) This proved far harder for many candidates with many gaining no credit and many others only half of the available marks. It was the mode of transmission that candidates struggled to answer.

(b) Candidates found this question difficult to answer, when they were simple being asked to name a defined computing term.

Question 6

(a) This question was more suited to the candidates’ knowledge and experience with computing ideas. Many candidates are more aware of ideas regarding presentations, but it was also common for repetitive answers to be given, such as giving the same benefit and drawback for each of the three points the question was asking for. This was a valid way of answering for two of the points but NOT the third. As a result, candidates lost marks here.

(b) This is a question that has appeared before, though it was presented in a slightly different way. It asked about the uses of specific application software. Candidates often did not relate their answer to the scenario presented in the question. There were also a number of candidates who answered the question using product brand names which gains no credit.

Question 7

(a) This form of question used to cause problems for candidates, but the skills of the candidates in working out a truth table when presented with a logic diagram has improved immensely.

(b) Producing a logic gate diagram from a Boolean logic statement is still a skill that needs a great deal of practise before many candidates gain more than a minimum amount of credit. There were still too many instances of single input OR gates and single input AND gates.

Question 8

(a) Candidates are clearly told that the array stores integers (as shown in the example in the question) but many candidates continued to put a space symbol to represent an empty square which is unlikely to be accepted into the array as presented.

(b) Writing code snippets is another skill that has not improved significantly with the passage of time. When this form of question is asked, many candidates appear to not know the approach they need to take to write an answer that will gain some credit. Many candidates appear do not know how to write simple nested loops. The most common answer seen was an incorrect attempt to declare and dimension the board array which had at least in part been given in the rubric of the previous question part.

(c) This question asked candidates to state the value of the contents of two indexed cells in the array and also state whether the first subscript indicated a row or column. Many answers showed just one part of the answer, not both. Where the contents were stated, most candidates gave “X” and “O” rather than the 1 and 2 that were stated in the initial scenario.

(d) The majority of candidates gained full credit on this question part.

(e) Very few candidates chose to answer by producing either pseudocode or a program snippet which would be preferred way of answering this kind of question.

Question 9

(a) In answer to this question, many candidates chose to simply reword the question but did not offer any additional information, and so gained no credit for their answer. Many answers were quite vague. Many candidates did not name a suitable hardware device. Many candidates did not recognise that this was a question about the HCI, presented in a slightly different way compared to previous years.

(b) Most candidates do not seem to have made much use of the preamble given at the start of this question. Candidates should have used this as the starting point to say how an appropriate computer system would store the customers’ requests and then match it to cars stored in pod system that had been manufactured. This should have involved the idea of using databases or tables to store both types of data and then involving a search to match one with the other.

COMPUTING

Paper 9691/12
Written Paper

General comments:

Candidates appeared to be well prepared for this examination. There were fewer weaker responses seen in this examination. A number of candidates interpreted some questions in ways that were not expected, and many others at times appeared to be answering questions that had been asked about a similar but not the same topic from a previous paper.

Comments on specific questions.

Question 1

(a) Most candidates gave the expected answer which was optical but a small number gave some unexpected answers such as hard disk, flash drive and CD-ROM. This latter response was actually in the stem of the question.

(b) Many candidates confused the question with an animated presentation. There were many answers referring to animation or slide transition to make the presentation entertaining. The candidates did not pick up on the meaning of the question or misunderstood the word interactive as the attractiveness of the presentation.

(c) Many candidates understand that a CD-ROM cannot be re-written or the alternative that it has enough storage capacity for a large number of lessons.

(d) (i) The fact that a CD-ROM is read only well known, but the correct answer “so cannot store user progress” was rarely seen.

(ii) Many candidates simply stated that "install the software" but should have given a target device, for example, the hard disk onto which the software is installed.

(e) Few candidates explained the differences between validation and verification. The majority of candidates simply gave examples of both processes.

Question 2

(a) A relatively straightforward question that most candidates found easy to answer. Where full marks were not achieved, it was usually because the candidate switched their answers in the top and bottom rows around.

(b) (i) Only partial answers were given by candidates on many occasions. This was especially true for candidates who gave answers about pressure receptors to detect intruders.

(ii) Many candidates gave a sensor to detect ambient light. Many candidates did offer alternatives which were not valid.

(iii) There were quite a variety of given answers and usually a valid one, but then the reason for that choice was less well answered with invalid reasons given for the most part.

Question 3

(a) Candidates provided a partial answer to this question. A single-user operating system can be used by more than one person, **but** only one person at a time.

(b) (i) The answers given to this question were often a simple rephrasing of the question that added nothing extra to the words and so did not gain any credit.

(ii) This question was generally correctly answered, but some candidates gave basic answers that would not require a high speed connection.

(c) When asked for two benefits of a broadband connection, it was unusual to see an acceptable answer to this question. The majority of candidates answered in terms of security, cost or ease of setting it up which was not what was asked for.

(d) (i) - (iii) The following parts d(i) to d(iii) all involved some minor calculations which proved too difficult for many candidates. Most candidates did carry out the calculations correctly but for part d(ii) did not add units to their answer which invalidated the number given.

(iv) This question asked for an opinion as to why a download speed might be slower than one advertised. Some candidates chose to answer by saying there was deliberately misleading advertising. The idea of over capacity or many people all downloading at the same time reducing individual band with was what was expected.

Question 4

(a) This was a straight forward question with the answer presented within the question and candidates just had to order their answer. A number of candidates ordered their answers incorrectly. Some candidates chose not to follow the instructions in the question and wrote a series of made up answers, not using the list provided. Usually what they wrote had no connection to the included list.

(b) (i) The question asks candidates to DESCRIBE ... , so just listing fact finding methods is not enough for an A level standard answer to gain full credit. The description need not be extensive, but each fact finding method should be expanded in some way. Only the answers that did describe rather than name a method gained full credit.

(ii) This was a question that has been asked in previous papers, though this time it had a slightly different approach. It was clear that the majority of candidates did know about corrective, adaptive and perfective maintenance, and gave valid descriptions of each type of maintenance.

(c) (i) This question was about Data Flow Diagrams and asked the candidates for what would be found within a DFD. This needs rather more than "*a flow of data*". This is rewording the question but adding nothing new and so gets no credit. From the answers seen, it would appear that there is a lack of real knowledge of DFDs.

(ii) Most candidates answered this question correctly.

Question 5

(a) There were some good answers for this question about truth tables. Equally, a number of candidates did not attempt to answer this question. Other candidates simply had no little knowledge of this topic, and appeared to put random ones and zeros in the table provided.

(b) Candidates were asked to write out a logic statement from a given logic diagram. Many candidates answered this question perfectly but a significant minority did not put parentheses into their answer which due to the precedence of the AND and OR gates caused them to lose at least one mark.

(c) **C**andidates were shown two logic diagrams and asked to state what basic gates they represented. There were almost as many answers as possible gates that are used. Full marks were fairly common, but the majority of candidates gained half marks because they stated the wrong gate for the first circuit.

Question 6

(a) When asked for a suitable data capture device to obtain vehicle registration data, “a camera” is not an adequate answer at this level.

(b) (i) When asked for a suitable input/output device when betting ready to pay their car park fee, most candidates gave the expected correct answer.

(ii) When asked how the system might calculate a correct fee for the time parked, many responses simply reworded the question or said the fee is calculated. There was no breakdown of the stages needed to obtain a correct fee for the time parked. A pseudocode approach is what was looked for. Very few candidates made any real effort here.

(c) (i) Programming an initialisation of an array is a skill that is not well known. It requires a nested pair of loops for the row and column so that each array element is covered and then set to a symbol that means empty. This was very rare to see in an answer and even fewer attempted to put this into some form of code.

(ii) This question just required a simple YES or NO for the answer after reading the grid/array of vehicles parked in the picture presented. Many candidates gave long sentences with no definitive decision one way or the other.

(iii) This question asked how a parked car’s location could be found from the information given. Few candidates seemed to realise that they also need to cater for the situation where an incorrect value is entered so that it returns an error if the car number plate is **not** found, which is where most able candidates lost a mark. The majority of candidates did not understand that to program this solution required nested loops to search the array for an entered number plate. A few candidates answered by attempting to show how the search would be coded.

(d) Candidates here were asked to extend the idea of a single level car park to a 10 level one and how to change the data structure to accommodate this change. There were more incorrect than correct answers. What was looked for was for a change from a 2 dimension array to a 3 dimension but **not** a multi-dimensional array.

Question 7

There are 5 related parts to this question where candidates are asked to explain **why** a given statement is incorrect. Many candidates did **not** explain why and just stated what was wrong, which is only half the expected answer.

(i) Often, incorrect answers refer to the size of a pen drive rather than potential virus problems on the pen drive. Candidates should realise that backup is **not** necessarily of the entire contents of a hard disk and so a pen drive, especially modern larger ones, can easily accommodate the backup of file changes. Many candidates appear to think that a virus cannot cause the loss of data.

(ii) Stack operation is very well known, so the answers often gained maximum credit.

(iii) The idea of how parity works is another item from the syllabus that is very well known, but many candidates did give imprecise answers.

(iv) Most candidates knew that broadband used analogue transmission rather than digital, **but** other valid points were rarely mentioned.

(v) Many candidates defined what a buffer is rather than answering the question asked i.e. **why** the statement is incorrect.

COMPUTING

Paper 9691/13
Written Paper

Comments on specific questions

Question 1

The differences between RAM and ROM are a subject that is well known by the majority of the candidates.

Question 2

(a) Most candidates gained full credit for this question, but there were some candidates who did not follow the instructions given in the question.

(b) (i) Many candidates chose to define broadband rather than answer the question that was asked. Most candidates found this a difficult question to answer.

(ii) Many candidates did not read the question properly and did **not** state applications that must have a high speed connection, instead giving basic browser applications or email which do not require such a high speed connection. Other candidates gave hardware devices as their answers which are not applications. Other candidates had not read the question properly and talked about higher bandwidth just making the Internet speed faster.

(c) (i) Many candidates could not explain, even in the most basic way, what the term circuit switching meant. This was a 2 mark question but most candidates gave only a simple and usually incorrect answer.

(ii) This is another question about how the Internet works with many candidates having a lack of knowledge of what a packet header would contain. Many candidates gave answers that were simply not connected to the question such as integer, character and string i.e. data types rather than the items the header actually holds.

Question 3

This question proved to be relatively straight forward for the candidates to answer with many obtaining full credit and the majority gaining more than half of the mark that were available. Serial access, archiving and disc formatting surprisingly proved to be the terms that were most often identified incorrectly. Some candidates drew two lines from the left hand side boxes when the question asked for **a** (one) line to be drawn. Candidates are reminded to read the question and follow the instructions if they are to gain full credit for their answer.

Question 4

(a) Working out the correct output for a truth table is a skill that is either very well known or completely unknown. The answers rarely fall into the middle ground. Some Centres should make more effort with Boolean logic.

(b) When asked to write a logic statement from the circuit shown a number of candidates did not write a logic statement but a verbal description which does not usually actually state the appropriate logic of the circuit.

Question 5

(a) When asked what the MAX function will return when applied to a range of cells, many candidates gave answers that were not about that function but some other, often wildly different function.

(b) When asked to create a spreadsheet function to determine whether a candidate had reached a certain standard, almost no answer gave a function which had the correct logic and syntax. Answers were awarded some credit for describing the logic in words, even though this was not was asked for.

(c) This was a question that above all required the candidates to think about the features of a spreadsheet and apply that to the analysis of candidate performance. Very few candidates mentioned appropriate ways the teacher could use these functions to show their analysis.

(d) This question is about off the shelf software that has appeared in previous papers. Common answers regarding advantages such as it is cheaper where not acceptable without more detail. Many of the answers were like this giving only part of the expected answer. Ideas such as immediately available and more widely tested were well known.

Question 6

(a) Candidates are asked to name two different network topologies. There were also asked to give one advantage and one disadvantage. It really works to the candidates disadvantage if they go on to give more than one advantage especially if one of the ideas is incorrect.

(b) This question asked about hardware that is used in a LAN or WAN, and if named should be common to all types of LAN. Many candidates incorrectly described normal computer hardware components such as a CPU or memory. The most common device named was a router, but few stated clearly why such a device is needed.

Question 7

(a) This question names four fact finding methods used in the system life cycle at the analysis and design stages, and asks how they could be used in this specific scenario. Many answers simply gave generic responses that were not connected to the given scenario and as a result gained less credit. The answers expected were the basic methodology, **but** connected to the scenario and the question clearly stated "... to this collection and delivery service".

(b) (i) This was a question about a specific HCI and candidates are asked to state why it is an inefficient method. Most answers covered just one point. The question scenario clearly says the telephone operator keys in all the data, **but** many answers were about what the customer keyed in, which is not relevant.

(ii) Candidates are asked here how the interface could be improved. Answers such as validation checks are not relevant to improving the interface. Answers regarding changing the colour add nothing to improving the efficiency of the interface. Many answers did mention the idea of drop down boxes but did not link it to an appropriate data entry field that would benefit from it.

Question 8

(a) (i) The majority of candidates could work out the appropriate binary for the integer value given.

(ii) More candidates made errors with this answer as they moved the binary pattern too far to the left usually by one or two extra places. Other candidates also placed the encoding value into their answer which was not asked for and so made an otherwise correct response incorrect.

(b) **T**he majority of candidates were able to use the information given to produce the correct result.

(c) Candidates often spotted that the coding value was too great, but very few could explain its effect of the alphabetic data.

Question 9

(a) What makes a number odd or even parity is well known by most candidates.

(b) The idea that sensor do not send signals direct to actuators was not well-known by most candidates.

(c) **C**andidates often did not do what the question asked i.e. say what was wrong with the statement. Instead they tried to correct the statement.

(d) The operation of a stack appears to be well known by the majority of candidates.

(e) The difference between a GUI and a CLI is well understood and usually very clearly expressed by most candidates.

COMPUTING

Paper 9691/21
Written Paper

Key messages

This was the last time this syllabus was offered for examination. It is being replaced by Computer Science 9608. It remains the case that to succeed in this paper it is essential that candidates have practical experience of programming using a high-level procedural language. Programming and pseudocode questions from past examination papers provide an ideal starting point for practical work.

General comments

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

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

Comments on specific questions

Question 1

This question required candidates to write procedures to print a formatted name label in the programming language of their choice.

(a)-(c) The more able candidates produced good solutions. Many candidates were not able to write simple print statements and formulate a procedure heading in their chosen programming language.

(d) Few candidates were able to list benefits of modular solutions that related to the scenario given in the question and listed the standard text-book answers. The suggestions that subroutines could be stored in a program library were acceptable. Many people being able to work on different parts of the solution was not appropriate in the scenario of printing a name label.

(e) Meaningful identifiers and indentation were the most common correct answers.

Question 2

This question guided candidates through the stages of designing a solution to a simple ticket price calculator.

(a) The more able candidates wrote program code to declare and initialise an array to store the distances between stations. Candidates need to have plenty of practice with this type of task.

(b) The question gave an example of test data and the reason for choice. Some candidates correctly chose other types of test data, rather than just repeating similar values and reasons. A particular important set of test data was to check that travelling in the opposite direction should give a correct result.

(c) Candidates need to understand that the solution suggested in a given flowchart is the method to be used for writing the required program code. Many candidates were not able to translate the pseudocode given in the flowchart symbols into program statements in their chosen programming language. This should be seen as a basic skill.

Question 3

The scenario given in this question referred to a simple restaurant table booking system, initially involving three 1D arrays.

(a), (b) Candidates completed some of the gaps in the pseudocode well. The parts that seemed to challenge candidates involved the use of array indexing. Python programmers need to be aware that in pseudocode they still need to be able to declare variables and their corresponding data types, even when this is not implemented in Python.

(c) Candidates need to understand that standard algorithms can be used as design templates and adapted for specific requirements. The `AvailableTablesReport` procedure required iterating through a 1D array and checking each element in the `Booked` array. This is similar to a sequential search and should be well rehearsed.

(d) Being able to declare a record structure and a corresponding array of records is a syllabus requirement. Very few candidates were able to make a successful attempt.

(e) Being able to save the contents of an array of records in their chosen programming language requires detailed knowledge of the mechanism of opening a file in the correct mode, assigning the given file name and saving individual array records. There was very little evidence that candidates had previous experience of such a task.

COMPUTING

Paper 9691/22
Written Paper

Key messages

This was the last time this syllabus was offered for examination. It is being replaced by Computer Science 9608. It remains the case that to succeed in this paper it is essential that candidates have practical experience of programming using a high-level procedural language. Programming and pseudocode questions from past examination papers provide an ideal starting point for practical work.

General comments

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

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

Comments on specific questions

Question 1

Candidates were required to complete a trace table. Some candidates did seem to notice that the `FOR` loop iterates from 6 to 1. The more able candidates worked sufficiently methodically to get the correct values in the `Sum`, `Carry` and `Result` columns.

Question 2

This question considered a function to calculate the result of a number raised to a power.

(a) The recursive definition was well understood by the more able candidates.

(b) Most of the candidates were able to use the code to calculate the result for the calls given.

(c) Many candidates were able to deduce that the call `Power(3, -1)` would not return the correct result. The reasons given varied widely and only the more able candidates correctly noted that the base case would never be reached. Other candidates suggested that there would be an error message, even though the code clearly does not include such. The solutions offered to this problem were often very unrealistic. It must not be done at the expense of what was working correctly initially. The simple suggestion of checking for a negative exponent 0 and returning an error code gained full credit. An alternative creditworthy answer was that the function definition should be extended, so that it manages negative exponents. Some candidates correctly noted that these would return a real value.

(d) Most candidates struggled to write the `Power` function as an iterative algorithm. Candidates need to analyse how this could be done with a simple loop. Here is a sample solution:

```
FUNCTION Power(Number : INTEGER, Exponent : INTEGER) RETURNS INTEGER
    Result ←1
    WHILE Exponent > 0
        Result ← Result * Number
        Exponent ← Exponent - 1
    ENDWHILE
    RETURN Result
ENDFUNCTION
```

(e) Some candidates could provide benefits of iterative and recursive solutions, such as: Iterative solutions are easier to write/debug and have smaller overheads.
Recursive solutions can be mathematically intuitive and usually contains fewer lines.

(f) It appears candidates do not have practical experience of using debugging tools such as breakpoints, variable watches and stepping. Many misconceptions were apparent. Candidates need to be aware that usually one breakpoint is sufficient when testing a program. The program will execute normally up to the breakpoint (in the context of this question the breakpoint should be set in the main program just before the `Power` function is called. The program code is then executed by stepping. This means one instruction is executed at a time. The variable watch window is checked to see if the values are as expected before stepping to the next instruction. In the `Power` function suitable variables for the watch window are: `Result` (this is the value to be returned at the end of each call) and `Exponent` (this has a different value each time the function is called).

Question 3

The scenario given in this question involved numbering squares in a crossword grid where a horizontal or vertical word can start.

(a) Even though the question stated that the white squares may have a number in them (if it is a starting square), many candidates suggested using a positive integer to represent a black or empty white square. Candidates need to understand that they are required to make sensible choices that are workable. 0 and −1 would have been appropriate choices, being of the correct data type. The more able candidates were able to write program code to declare and initialise a 2D array. Candidates need to have plenty of practice with this type of task.

(b) Candidates completed some of the gaps in the pseudocode well. The parts that seemed to challenge candidates involved the use of array indexing. Most candidates were able to state which parameters would be passed by reference and which would be passed by value.

(c) Candidates need to understand that the solution suggested in a given pseudocode is the method to be used for writing the required program code. Many candidates did not seem to be able to translate the pseudocode given into program statements in their chosen programming language. This should be seen as a basic skill.

(d) Meaningful identifiers, comments, uppercase keywords and indentation were the most common correct answers.

(e) Being able to save the contents of an array in their chosen programming language requires detailed knowledge of the mechanism of opening a file in the correct mode, assigning the given file name and saving individual array elements. There was very little evidence that candidates had previous experience of such a task.

(f) The more able candidates were able to write a sensible solution for this function. Although the function heading was given, many candidates did not take notice of it, did not re-write it, or did not make use of the given parameters. The function also returns a value, but this was also often forgotten. Here is a sample solution:

```
FUNCTION CountSquaresAcross(Puzzle, ThisRow, ThisColumn) RETURNS INTEGER
    DECLARE WordLength : INTEGER
    WordLength ← 2 // this was the minimum word length
    WHILE Puzzle[ThisRow, ThisColumn + WordLength] = WHITE
                                 AND (ThisColumn + WordLength) <= 11
        WordLength ← WordLength + 1
    ENDWHILE
    RETURN WordLength
ENDFUNCTION
```

COMPUTING

Paper 9691/23
Written Paper

Key messages

This was the last time this syllabus was offered for examination. It is being replaced by Computer Science 9608. It remains the case that to succeed in this paper it is essential that candidates have practical experience of programming using a high-level procedural language. Programming and pseudocode questions from past examination papers provide an ideal starting point for practical work.

General comments

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

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

Comments on specific questions

Question 1

The topic of this question was the production of a conversion table from Fahrenheit to Celsius.

(i) Most candidates were able to complete the identifier table and noticed that the only variable that was not an integer was `Result`.

(ii) The more able candidates were able to complete the gaps in the given pseudocode. Some candidates did not use the input values as starting and end values for the `Fahrenheit` variable. The steps for conversion were given in the question, but some candidates did not translate these into pseudocode statements.

Question 2

This question guided candidates through the stages of designing a solution for a vending machine change calculator.

(a) Candidates need to understand that the solution suggested in a given flowchart is the method to be used for writing the required program code. Many candidates did not translate the pseudocode given in the flowchart symbols into program statements in their chosen programming language. This should be seen as a basic skill.

(b) (i) The more able candidates were evaluated the statements involving DIV and MOD, and explained what the results represented:

```
ChangeDue DIV 50 represents the number of 50-cent coins.
ChangeDue MOD 50 represents the remaining change due (after 50-cent coins
were removed).
```

(b) (ii) Some candidates were able to use **Part (b)(i)** to write the procedure OutputChange. Some candidates made the task more difficult by using meaningless identifiers for the intermediate results. An example solution (in pseudocode) is:

```
Coins50 ← ChangeDue DIV 50
Print("Number of 50c coins: ", Coins50)
LeftOver ← ChangeDue MOD 50
Coins20 ← LeftOver DIV 20
Print("Number of 20c coins: ", Coins20)
LeftOver ← LeftOver MOD 20
Coins10 ← LeftOver DIV 10
Print("Number of 10c coins: ", Coins10)
```

Question 3

The scenario given in this question referred to completion scores, initially involving three 1D arrays.

(a) (i), (ii) Candidates completed some of the gaps in the pseudocode well. The parts that seemed to challenge candidates involved the use of array indexing. Python programmers need to be aware that in pseudocode they still need to be able to declare variables and their corresponding data types, even when this is not implemented in Python.

(a) (iii) There seemed to be a lot of confusion over the meaning of the BYREF keyword. **Part (a)(i)** showed that Average was assigned a value just before the end of the procedure. This should have been sufficient a hint that the value would need to be returned to the calling program. Candidates need to understand that generally parameters can be passed by reference or by value, even if some programming languages do not support this.

(b) (i) The given pseudocode was a standard bubble sort algorithm. Most candidates successfully completed the trace table. Candidates need to be aware that when an algorithm is dry run, it is helpful for each new iteration to enter values in a new row. Values that are not re-assigned do not need to be entered again. Many candidates did not complete the second round of iterations when nothing changes in the array, but x iterates again from 1 to 9 with NoMoreSwaps remaining TRUE.

(ii) The more able candidates noticed that the reason for the error was that only the points totals were swapped and not the corresponding name and score.

(c) Being able to declare a record structure and a corresponding array of records is a syllabus requirement. Very few candidates were able to make a successful attempt.

(d) This part question required candidates to complete the pseudocode for a sequential search. The more able candidates completed this task well. Candidates need to understand that learning algorithms by rote is very little use when an outline structure is given that is to be completed using the intended method. It is important to understand the essence of standard algorithms. For example, the sequential search required to find a competitor's name using the REPEAT UNTIL construct is a standard method and candidates should be able to use the identifiers given in the question.

(e) Being able to save the contents of an array of records in their chosen programming language requires detailed knowledge of the mechanism of opening a file in the correct mode, assigning the given file name and saving individual array records. There was very little evidence that candidates had previous experience of such a task.

COMPUTING

Paper 9691/32
Written Paper

Key messages

This was the final series for this examination paper. The 9691 syllabus has now been replaced by the 9608 Computer Science. Centres would be advised to note much of the content of this report in preparing candidates for the new 9608 syllabus. There are many topics which have been retained and any feedback contained in this report should be considered relevant in preparing candidates for the new syllabus papers.

General comments

It is clear that weaknesses seen on previous examination papers have been addressed and there are clear signs of improvement. This was apparent for **Question 2**, where clear answers describing the issues with Second Normal Form and Third Normal Form were seen.

Weaknesses with the need to read the key words in the question rubric remained and were a direct result of loss of marks. **Question 1** was assessing the understanding and application of Backus Naur Form (BNF). One or more questions on previous papers had required the candidate to simply list, as a number sequence, the rules which had to be used to prove the validity or otherwise of a given expression. In **question 1**, the key word in three parts for **(b)** was to "*explain*". The consequence was typically that an answer such as "Use rules 3 and 4" gained no credit. An answer such as ""A `<nounPhrase>` can be a `<noun>`, and `dog` is a `<noun>`, therefore dog is a valid `<nounPhrase>`" gained the full two marks.

Question 1

(a) Most candidates were able to recognise rule 6 as recursive, but the statements often did not gain credit with explanations which were using the term recursive in the context of programming using functions.

(b) A common weakness for answers seen for the first three parts of **(b)** was to attempt to describe the rules, but not to relate the various structures back to the given statement.

(i) See the earlier comment made in the General section.

(ii) Many answers were not clear, simply stating that the word `dog` did not appear in any of the rules. To gain full credit, an explanation of the given statement, `a puppy sat` was required. One of the available marks would then be for stating that puppy would need to be a `<noun>` for this to be a valid sentence.

(iii) Most candidates scored some marks. Weaker responses correctly recognised a cat slept as a valid `<verbPhrase>` and hence a valid `<sentence>`. Stronger answers then explained the need for the recursive rule 6.

(iv) The majority of candidates introduced a new rule `<adverb>` consisting of the list of given words. The most common answer (and the simplest) was then to amend the existing rule `<verbPhrase>` to include the alternative `<nounPhrase><adverb><verb>`. Other answers involving an amended `<sentence>` rule were possible and gained credit.

Question 2

(a) The majority of candidates stated that the table had a repeated group of attributes to score the mark. Candidates should not confuse this statement to mean that certain data values are repeated within the table.

(b) (i) This was well answered. Candidates either stated simply “many-to-one” or gave an expanded statement including the two table names where the meaning was clear, “one manager manages many bands”.

(ii) This was well answered. Candidates correctly used the terminology primary key and foreign key.

(c) See the comment made earlier in the General section. This was well answered. For the first mark candidates either used the terminology partial dependencies or gave a clear description of this. For the second mark reference to the specific attributes in the given table was required, namely that the `Genre`, `NumberInBand` and `SetFee` would be determined by knowing only the `BandName` (and not the `BookingDate`).

(d) (i) This was well answered. Candidates stated that a new `AgreedFee` attribute had been added to the `BOOKING` table. Some candidates did not state the table and were penalised.

(ii) This was well answered. Candidates stated that a new `BookingTime` attribute had been added to the `BOOKING` table. Candidates were penalised if they omitted the table name.

(iii) This was well answered with candidate stating the addition of a new table `VENUE`.

(iv) Answers where candidates scored all the available four marks were rare. The two marks which were more elusive were for stating that `BandName` and `VenueName` were two foreign keys in the `BOOKING` table, and that the primary key for the `BOOKING` table would need to be `BandName` + `BookingDate` + `BookingTime`.

Question 3

(a) (i) This was well answered. The majority of candidates scored the one mark available.

(ii) One mark was the most common outcome for both parts. For **(ii)**, candidates generally wrote the correct form for “p squared”.

(b) (iii) The common error was to omit the brackets around the expressions to indicate the correct multiplication by 3.

(c) This part required the application of knowledge from two parts of the syllabus. This was generally well answered, with many candidates scoring at least seven of the available eight marks.

Question 4

Parts (a), (b) and **(c)** only required the conversion of one form of representation to another. All three parts were generally well answered.

For **part (d) (i)**, most candidates were able to state that the number can be recognised as negative as the most significant bit of the mantissa was a 1 bit. Answers seen for **parts (ii)** and **(iii)** were generally weak. Many answers stated −88 as the value of the mantissa, undermining any understanding of the use of floating point representation. For part **(iii)**, an attempt was made by the marker to follow through from an incorrect answer given previously for **(ii)**; conditional on the mantissa stated in **(ii)** being a fraction.

Part (e) was well answered with candidates stating that the first two digits of the mantissa will be different, or for the given number the mantissa starts with digits 10.

Two correct answers for **part (f)** was rarely seen. The second number generally was the better answered with candidates appreciating that the largest number would have the largest possible positive mantissa and exponent.

Question 5

Most candidates scored for **(a) (i)**, stating that a static structure is one which has a fixed size, sometimes supporting this with the example of an array declaration. For the explanation of a dynamic structure, the statement that the size is effectively changing at execution time was often missing and so did not gain credit. Answers for **part (ii)** were often unclear with the omission of the key word 'memory'. A common misunderstanding was that 'fixed size' referred to the number of characters/bytes which are used for a record type structure as shown in the stem of the question.

Part (b) was well answered with the majority of candidates gaining the two available marks. **Part (c)** was well answered with the candidates showing the correct head pointer and the conceptual linking of the nodes. A not uncommon error was to interchange GREENE and HASAN, presumably where candidates thought H was alphabetically before G.

Answers for **(d) (i)** were varied. To score the three marks, the candidate needed to show the value 130 in the data position of the empty node. Then, complete the link value of 28 with an arrow linking it to the 180 node. The final mark was for crossing out in some way the 28 pointer value and then drawing an arrow to show the new link from the 20 node to the new one.

Although answers were varied, there was evidence that the answers did show considerable improvement on those seen in an earlier paper. The key points looked for were to start at the Head pointer, use the pointer at each node to follow through the nodes until the first value was found, which was smaller than the value to insert, two pointer changes then needed to be made.

Question 6

The question framework used here was different to that used on previous papers where the candidate was given a definition for a robot, but the scenario should have been one familiar to the candidate. Candidates generally struggled to secure three or four of the available marks.

Key points expected were that:

Mechanical device
The structure is made up of motors, actuators and stepper motors which make it a mechanical device. Some candidates were side-tracked here with statements such as there is no human involvement and that the robot can operate 24/, none of which gained credit.

Movable
Since the robotic arm must travel to all parts of the car surface both in a horizontal and vertical direction.

Able to Sense their surroundings
It is fitted with various sensors designed to detect typically, the presence of a car body and distance of the paint nozzle from the surface of the car.

Controlled by a computer program
Answers seen here often did not score the mark, but there were numerous statements which would have secured the mark. The program is made up of instructions which determine the action of the sprayer based on the continual inputs received from the sensors.

For **part (b)**, answers which scored the maximum four marks were rare. Examples which all scored included a named sensor, actuator, motor, the central processing unit, main memory or secondary storage.

Question 7

(a) (i) Most candidates scored the mark with statements such as the original code written by the programmer, probably in a high-level language, but for part **(ii)**, the term object code was not well understood. Many candidates focussed on the word 'object' and described code which was written using object-oriented programming.

(b) (i) This was generally well answered. Better answers described a table containing all the key reserved words of the particular programming language, with an associated token used for replace, at compile time, each key word.

(ii) Most candidates were able to study the given program code and identify three reserved words.

(iii) A significant minority of candidates were still describing a table which contains all the language symbols such as the arithmetic operators which would be used within a source program. Only the very best answers described the storage in the table of the actual memory address allocated for each identifier.

(iv) **C**andidates who realised that the table contained the identifiers used by the programmer were able to list three constants, variables or the procedure name. Credit was not given for answers which wrote a complete statement from the code containing one or more of the identifiers.

(c) Proved to be a good discriminator of the candidates' understanding of lexical analysis. Weaker answers were often unable to suggest more than the removal of white space and comments. Weak answers often wrongly described additions being made to the keyword table. Answers often vaguely described the process of producing a 'string of tokens' which gained no credit.

(d) **(i)** Many candidates were able to score the two available marks. The popular answers for part **(i)** were to state that the process is designed to make the code execute faster or to use less memory space. Vague answers that talked of 'faster code' or 'shorter code' did not gain credit.

(ii) Candidates identified the final two lines where calculating the value of x + y twice could be avoided.

COMPUTING

Paper 9691/33
Written Paper

Key messages

This was the final series for this examination paper. The 9691 syllabus has now been replaced by the 9608 Computer Science. Centres would be advised to note much of the content of this report in preparing candidates for the new 9608 syllabus. There are many topics which have been retained and any feedback contained in this report should be considered relevant in preparing candidates for the new syllabus papers.

General comments

It is clear that weaknesses seen on previous examination papers have been addressed and there are clear signs of improvement. This was apparent for **Question 2**, where clear answers describing the issues with Second Normal Form and Third Normal Form were seen.

Weaknesses with the use of SQL remain. Candidates are confident with the use of Structure Query Language to perform a query, but scripts for the addition, deletion and amendment of data are not well understood.

Question 1

(a) (i) This question part was answered well. Most candidates were able to identify `BandName` and `VenueName` in the `BOOKING` table as foreign keys.

(ii) This was answered well. There were three relationships which were valid:

- One `BAND` gets many `BOOKING`s
- One `VENUE` is used for many `BOOKING`s
- Many `BAND`s play at many `VENUE`s

(b) (i) See the comment made in the General section about the improvement in the answers seen for this topic area. Candidates identified the `BAND` table as the one not in 2NF. Candidates then either used the correct terminology describing partial dependences, or explained this in terms of the attributes in the table; namely that the `NumberOfMusicians` value would be determined by only knowing part of the primary key, the `BandName` value. The revised table for the final mark required the removal of this attribute from the `BOOKING` table.

(ii) See the comment made in the General section about the improvement in the answers seen for this topic area. Candidates correctly identified the `BAND` table and then described non-key attributes `ManagerName` and `ManagerPhoneNumber` which were dependent on `ManagerID`. The re-design required the removal of these attributes to a new `MANAGER` table with `ManagerID` as the primary key.

(c) This question part was answered well by candidates.

(d) The syntax for the SQL was known only by a few candidates. Candidates often scored one mark for a correct `WHERE` clause requiring the `AND` operator.

Question 2

(a) Most described the operation of a stack as "last item added will be the first to leave". A minority of candidates used "First item added will be the last to leave".

(b) Candidates answered this well, describing a queue as "the first item added will be the first to leave".

(c) (i) Most candidates were able to score at least two of the available marks. A common error was a not initialising Head and/or Tail or initialising each with a value of zero.

(ii) This was well answered. Most candidates described a loop counter or array index.

(d) (i)(ii)(iii) All parts were well answered with candidates, placing the items in the correct positions and correctly labelling the Head and Tail pointers.

(d) (iv) Most candidates recognised that the final Queue-A had the items in reverse order.

(e) (i) All candidates scored some of the available marks with able candidates securing the full four marks.

(ii) This question required knowledge and understanding from two areas of the syllabus; namely the operation of the stack data structure and object oriented programming. The question required familiarity with object oriented notation and practice, the use of an identifier for the object and the dot notation to denote the three methods `InitialiseStack`, `Push` and `Pop`. The candidate then needed to appreciate that the `Push` method will have a single parameter, the various data values given in the stem of the question, whereas the `Pop` method does not require a parameter. It would be helpful if centres exposed candidates to practical programming exercises like this.

Question 3

(a) (i) Most candidates correctly described the instruction as the number 129 being copied to the Accumulator.

(ii) This question was answered well. Candidates who omitted the leading zero from the four digit representation were penalised.

(iii) There was evidence of an improvement in the standard of answers. Candidates simply needed to state *"there will be fewer digits to write"* to secure the mark.

(iv) This was answered well. Candidates needed to read the 8-bit binary code for the 'store' instruction and then convert denary 90 to an 8-bit binary value.

(v) Most candidates correctly stated that this was a true statement and followed this with one of the examples from the given instructions; `OUTCH`, `IN` or `END`.

(b) All candidates were able to score marks from the trace table.

Question 4

(a) This was a question framework which was used for the first time to assess this area of the syllabus and was generally well answered. Most candidates were able to secure four or more of the available marks.

(b) (i) Candidates often correctly identified this instruction as an example of 'case 2' but then were not able to clearly express why this was so. A clear answer would have been *"the address held in the Current Instruction Register must now be loaded onto the address bus so that the data value at this address can be retrieved"*.

(b)(ii) Candidates intuitively decided 'case 1' but then the explanation did not support this. A statement such *as "the operand is a register and therefore requires no further access of the main memory"* was required.

Question 5

(a) (i), (ii) Most candidates scored marks stating that the identifiers required would be an array index (used as a loop counter) and a Boolean variable to flag when the item is found. The question asked for pseudocode and so it was the syntax documented extensively in the syllabus which was expected.

The key points looked for in the algorithm were:

- The input of `SearchItem`
- The initialisation and incrementing of the array index
- The test for `MyList[i]` equal to `SearchItem`
- A post or pre-condition loop structure correctly formed
  Note, candidates who used a `FOR` loop structure did not gain credit
- The test for the end of the loop: `IsFound` = `TRUE OR i = 7` (or similar)
- On exit from the loop, a test for 'found' and output message if not found

(iii) This was answered well. Candidates understood that for a sequential search, the average number of comparisons would be 125.

(b) Candidates correctly stated that the items need to be in order to carry out a binary search.

(c) This was the first series where an insertion sort has had been examined. The question was well answered, with many candidates scoring the full three marks.

Question 6

(a) (i) Most candidates were able to score marks. Weaker answers simply gave an explanation of each of the three states. Candidates gained more marks if they followed the question rubric to "Describe how the state of a process changes…"

(ii) Answers here often gained no credit. The candidate may have understood what was meant by an interrupt but was not able to apply this knowledge of when an interrupt would be used as a process changes state.

(b) There were some vague responses here. They described little more than hard disk storage is needed. The answers expected were to state that the OS must maintain a file directory for all the files stored and that it will be the task of the OS to manage the unallocated storage units using a File Allocation Table (or similar, if the candidate's practical experience was in the use of a non-Windows OS).

(c) The final part should have been straightforward; the other tasks of any OS will be to provide a user interface, the management of the main memory and control of all input/output devices.

COMPUTING

Paper 9691/04
Project

General comments

This report provides the 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.

The projects submitted covered a wide variety of topics with stronger responses 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: Computing project**. There is a useful checklist for teachers and candidates that sets out the expected contents of each section in **Appendix 8.2 Guidance on marking the Computing Project**.

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 teachers at the Centres and Moderators to easily check that work for all sections has been included. It is also essential that the pages of the report are clearly numbered by the candidate.

Project assessment and marking

Most Centres used the marking grid on pages 45-52 of the current syllabus to provide a breakdown of marks showing the marks given for each sub-section of the report. In order to aid the process of moderation, the completed grid should include references to the appropriate pages in the 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 specific questions

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

Candidates who clearly documented client and user involvement in their investigation and considered carefully the evidence obtained from interviews, observation of the existing system and study of documents currently in use; then asked follow up questions to fill in any gaps in the knowledge obtained about the current system or requirements needed for the new system gained good marks. Alternative approaches need to be discussed in depth as they would be applied to the candidate’s proposed system.

(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, more able candidates used pseudocode and/or flowcharts to provide a detailed description of the processes to be implemented. Evidence from the solution is not required here. Gantt charts are not required.

Few candidates obtained marks in the top two bands for this sub-section, by obtaining evidence that their client had seen and commented on the design work, and then showing what had changed as a result of these comments.

(ii) Intended benefits

Candidates, who obtained good marks for this sub-section, described the benefits of their intended system, rather than providing a list of general statements that could apply to any system.

(iii) Limits of the scope of solution

Candidates who described the limitations of their intended system and included an estimate of the size of any files, based on information provided by the client, scored full marks.

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

This section should include evidence of development shown by program listings of code written by the candidate, the 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 was annotated by the candidate.

(ii) Programming

Candidates gained good marks if they showed that they could apply the programming skills developed at AS level in Paper 2 to a real situation.

(iii) Testing

Evidence of testing, supported by a well-designed test plan for all parts of the system that included the identification of appropriate test data, including valid, invalid and extreme cases, together with expected results for all tests gained full marks.

(iv) Installation

Candidates gained full marks if they provided an implementation plan containing details of user testing, user training and system changeover together with written evidence from the client and/or user(s) to show that the system had been seen, used and tested, and the candidate's plans had been agreed.

(e) Documentation

(i) Systems maintenance documentation and (ii) User guide

These sections were completed to a good standard by most candidates.

(f) Evaluation

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

Candidates gained good marks if they considered each objective set out in **(c)(i)** and explained how their project work met the objective, cross referenced to evidence in their report, or explained why the objective was not met.

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

Candidates gained good marks if they obtained written responses directly from their client and/or user(s) showing that they have used the system, and then evaluated those responses.