COMPUTER SCIENCE

Paper 9608/12
Written Paper

Key messages

Candidates need to use precise terms related to the subject at this level of study.

Each question needs to be read carefully. A question that begins with 'Describe' requires a different type of answer from one that begins with 'Explain'.

Candidates are required to respond appropriately in the given context of the question. Generic answers that apply to any situation will not gain full credit.

General comments

In general, candidates performed well on low level languages question, as well as the question on logic gates and truth tables. Many candidates found the questions about hardware and Database Management Systems (DBMS) more challenging.

Some candidates write their answer first in pencil and then over-write in ink. These candidates must ensure that the pencil marks are completely removed.

Comments on specific questions

Question 1

(a) (i) There were many correct answers to this question. The majority of candidates were able to give a valid IPv4 address. The most common incorrect response either used a colon as a separator, or included values above 255.

(ii) This question asked ‘why’ there is a need for IPv6 addressing. Many of the answers did not answer this question, but described IPv6 addressing in general terms. Some candidates need to understand that IPv6 is required because IPv4 addresses are running out.

(iii) Most candidates recognised that the address given would be invalid as an IPv4 address because the value in each group of numbers was out of range. Some candidates need to understand that the maximum value used in IPv4 addressing is 255, not 256. A common incorrect statement was that IPv4 addresses cannot be represented in Hexadecimal notation.

(b) (i) A minority of candidates answered this question well. Candidates need to improve their understanding of the transmission of data using a Public Switched Telephone Network (PSTN). Many responses just re-worded the stem of the question by saying that switches were used in this type of transmission, which is insufficient for credit at this level of study.

(ii) Many of the benefits listed were very appropriate and the majority of candidates answered this part well. Some candidates need to be more precise in their suggestions for drawbacks. Answers such as, 'It is expensive' are not detailed enough for credit at this level. There needs to be a reason why it is more expensive. An example of a good answer is, 'A drawback would be that a leased line is expensive to install and maintain'.

(c) This question asked for the role of routers and gateways in a network. There were some very good answers explaining both the similarities and the differences between routers and gateways. Other candidates incorrectly described the role of a router in connecting a single device to the Internet, rather than the role of a router in a network.

(d) Almost all candidates could identify three other types of server. The most common incorrect answer was a client-server.

Question 2

(a) (i) Almost all candidates correctly stated the purpose of a language translator.

(ii) Many candidates were able to state that easier de-bugging is a benefit of using an interpreter during program development. Fewer candidates were able to correctly state a second benefit. Many answers described how an interpreter operates rather than stating benefits of use. There was some confusion between an interpreter and a compiler.

(iii) Almost all candidates correctly named an assembler.

(b) This question begins with '*Explain why*', so a generic description of library files does not answer the question. Candidates should understand that to answer this type of question it is insufficient to say, for example, '*Library files will have been used by a lot of people so will be well-tested*'; there needs to be a statement as to why this fact aids software development.

Question 3

(a) Most candidates answered this question well. A small number of candidates need to take care when drawing the arrows in the statements to ensure that they point in the right direction.

(b) There were a small number of excellent answers to this part question. Many candidates need to improve their understanding of processor instruction groups.

(c) Tracing of assembly language programs is improving steadily and this one was very well done, with many candidates providing completely correct solutions, neatly set out in the table provided. The most common error was the inclusion of the word `END` in the `OUTPUT` column after the final statement.

(d) (i) Almost all candidates correctly converted the denary value to 8-bit binary.

(ii) Almost all candidates correctly converted the denary value to Hexadecimal.

(iii) Most candidates answered this question well. The most common error was the omission of the two leading zeros on the final answer i.e. 0044.

Question 4

(a) (i) A minority of candidates provided a good response to this question. Other responses demonstrated considerable confusion between the internal operations of the microphone, that is, how the microphone captures the sound, and the encoding of the analogue sound to digital form.

(ii) Many candidates found this part question challenging. There was considerable confusion between the operation of a laser printer and the operation of a scanner. Candidates need to improve their understanding of how a laser printer works. Many answers incorrectly stated that the laser engraved the text onto the paper, or that the laser melted the toner ink onto the paper.

(b) Some candidates were able to describe how all the data for a single frame was recorded or displayed at the same time. Many candidates need to improve their understanding of what is meant by progressive encoding. Some candidates confused progressive encoding with interlaced encoding and vice versa.

(c) There were a number of good answers to this question. Many candidates need to improve their understanding of SRAM and DRAM. The question begins with, 'Explain the differences'. Some candidates need to understand that in order to answer this question each statement needs to refer to both SRAM and DRAM. An example of a good answer is, 'SRAM is used in cache memory, while DRAM is used in main memory'. Answers such as, 'SRAM is more expensive than DRAM' or, 'SRAM is faster than DRAM' are not detailed enough for credit at this level. There needs to be a reason why it is more expensive and how it is faster. A better answer is, 'SRAM has faster access times than DRAM'.

(d) (i) There were a number of good answers regarding the idea of legal ownership or the need for permission to use. Some answers to this question did not give a clear definition of copyright. There was some confusion between copyright and plagiarism.

(ii) There were some excellent answers to this question where the responses were given in context. Many candidates gave a generic description of an open source licence and had not applied their answer to the context given in the question. This is a question where the precise use of terminology is important too. It is insufficient to say that Shazia does not want anyone to see the 'software'; it is the *source code* that she wants to keep private.

(iii) The most popular correct answers were Shareware and a Commercial licence, and the majority of candidates correctly stated these two types of licence. A small number of candidates responded with a name only, despite the question asking for a name **and** description.

Question 5

(a) (i) The majority of candidates correctly stated that data redundancy referred to the duplication of data.

(ii) Many responses included a statement about normalisation and there were a small number of excellent descriptions of the normalisation process. Some candidates need to improve their understanding of relational databases. There were many answers that mentioned multiple tables but few descriptions of how the data was stored and linked. This is another question where the precise use of terminology is important. There was considerable confusion between tables and databases, with the word *database* frequently being incorrectly used to refer to a table within the database.

(b) (i) Many candidates correctly explained the difference between security and integrity. A small number of candidates need to improve their understanding of this topic. It is also not enough, at this level of study, to simply repeat the wording of the question. It is not enough to say, 'security is keeping data secure'. There needs to be some explanation of the meaning of 'keeping data secure'.

(ii) This question is one that needs to be read carefully. It asks for two security features of a DBMS. There were some very good answers. Other responses referred to security measures such as firewall and anti-virus software that are not part of the DBMS.

(iii) The majority of candidates need to improve their understanding of the terms query processor and developer interface.

Question 6

(a) This question was answered very well. The majority of candidates drew a logic circuit that correctly represented the logic expression given. Many of the solutions demonstrated excellent understanding by using a 3-input OR gate as the final gate. There is an improvement in the standard of the diagrams over the previous year. There were far fewer instances where `AND` and `OR` gates could not easily be distinguished.

(b) Almost all candidates provided a completely correct truth table.

COMPUTER SCIENCE

Paper 9608/22
Written Paper

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to have developed these and need to be able to apply them to the scenarios presented to them to access the full range of marks.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and candidates should ensure that they use these correctly. It is also important that candidates writing program code use the correct syntax for their chosen language.

The understanding of fundamental programming concepts is essential. Examples of confusion include the difference between a literal and an identifier and the difference between `OUTPUT` and `RETURN`.

Candidates should read each question carefully before attempting to answer it. Questions may address individual topics in a number of different ways.

General comments

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

A significant number of candidates demonstrated skill levels suggesting they had little programming experience.

Comments on specific questions

Question 1

(a) (i) The majority of candidates indicated a good understanding of the concept of an algorithm as a sequence of steps. A frequent mistake was to refer to specific programming features.

(ii) Candidates responded well to the activity types. There were some mixed responses for the pseudocode example. A common mistake was to give `INPUT`, followed by a prompt, but without a variable name as in the following example:

```
INPUT "Enter a value"
```

A small but significant number of responses *described* the activity instead of giving an example. As stated in the Key Messages above, candidates need to read the question carefully to ensure they understand it before making their response.

(b) (i) Most candidates gained some credit on this question. A common fault was to omit the quotation marks for the final answer, which are needed to differentiate between a string and an identifier name.

(ii) This type of question has appeared relatively frequently, and virtually all candidates were able to achieve high marks. Common mistakes included `STRING` instead of `DATE`, and `STRING` instead of `CHAR`.

Question 2

(a) (i) The question used the key word 'algorithm'. The majority of responses gave answers that incorrectly related to the code itself.

(ii) Correct responses usually referred to reusable code and the ability to assign modules to different programmers or teams. A significant number of candidates gave answers that related to library routines rather than to subroutines in general.
A number of candidates used a response, such as "Easier to..." that only vaguely related to the question.

(iii) Most candidates correctly identified the module as a function and were able to give a correct justification. A relatively common mistake was to state that `DoSomething()` was a procedure because it "performed a task".

(b) At this level, candidates were required to use the correct technical terms when describing the activities. For example, just 'code' is too vague when referring to the source code. Additionally, candidates should appreciate that simply stating, "the Editor is used to edit" is not sufficient at this level.

Candidates who used the correct technical terms usually gained full credit. Translating source code to object code was a frequent and accurate response for Translator. A response such as "converting code to a language understandable by the computer" is too vague at this level of assessment.
A small number of responses suggested that low-level code would be translated into high-level.

Those achieving one mark often gained it for their response given for Debugger, with most identifying its use during program testing to detect errors.

(c) Most candidates identified the correct control structure. Few responses adequately described the function of the code. In many cases, responses simply contained copies of the statements from the question rather than descriptions of their actions.

Many candidates correctly stated that the code would only carry out the actions when `Result` is less than 20. Few mentioned the parameters being passed. Candidates often attempted to describe the terminating condition (i.e. `Result >= 20`) and it was common in these cases for the "or equals" part of the comparison to be omitted.

Question 3

(a) (i) Many candidates correctly used parameters A and B. A significant number of responses suggested a function rather than a procedure.
Only a small number of candidates recognised that parameter C was passed by reference.

A small but significant number of candidates were not able to answer this question.

(ii) Many candidates gained full credit on this question part. Candidates appeared to be much more familiar with functions than they are with procedures.

(b) 'Iteration' and 'Selection' were the most popular responses. Some responses lost marks for incorrect terminology such as stating that the structure chart contained `IF` statements.

A small number of responses repeated the features given in the question.

Question 4

(a) (i) Almost all candidates gained credit for the first column, and most also gave the correct characters for the second column. Many candidates did not gain credit for the second column due to not enclosing values in the column in quotation marks to identify them as characters. The third and fourth columns were often correct, although many contained an erroneous leading zero. A mixture of responses was seen for the fourth column. The most common mistake was not restarting the string after each "number”, and therefore ending up with a copy of the original parameter string.

(ii) There were a significant number of correct responses to this question part. Many other candidates gained this mark if they had correctly carried forward the final value for the variable `Selected` from their trace table.

(b) (i) Those candidates that gained credit, usually identified that there was no space at the end of the string. Fewer went on to explain that this meant that the final comparison would not be made.

A large number of weaker responses referred to either logical errors in the code, the use of invalid data types or incorrect type conversion. A common mistake was to attempt to add one to the loop count in an attempt to check the last number. Some responses simply repeated the phrase from the question, "the function did not return the largest value".

A number of responses gave pseudocode rather than a required explanation.

(ii) There were some very good responses that stated another selection statement should be added before the final value is returned, to compare `Selected` with the value of string `NewValue`.

Many candidates gained credit by stating that a space should be added to the end of the string. Often they did not specify where in the algorithm this should be done. A number of responses suggested that the function should be called with a different value string, which did not represent a change to the algorithm as the question required.

Many weaker responses simply suggested the use of white-box or black-box testing.

Question 5

(a) Many responses gained partial credit, usually for opening the file and looping until the End of File.

Answers often referred to the use of a counter variable. Few stated that it should be initialised to zero.

Many solutions included incrementing a counter within a loop. Reading a line from the file was frequently omitted.

Some solutions suggested the use of an existing function to perform the counting.

Response content varied widely, from simple descriptive prose to pseudocode or program code.

(b) A small number of candidates provided a full correct response. The majority of candidates recognised the need to open the file, perform a count-controlled loop, and finally make some form of output.

Many candidates correctly opened the file for reading. They often did not close the file.

A common mistake was to attempt to use `EOF()` and `CLOSEFILE()` without providing a file name, and as in **part 5 (a)** many solutions did not include reading the file or did not include a variable to hold the line read.

Although the question asked for a procedure, many solutions gave a function header. The requirements specified in the three bullets were often not satisfied, particularly regarding the provision of the file name as a parameter. Many solutions used a literal file name string in the procedure, in some cases, despite including a parameter in the module header.

Question 6

(a) Most responses included a function header and some form of loop. Many candidates gained some credit for declaring the local variables and for the prompt and input. Some responses used the correct code for the final concatenation and return. Some candidates incorrectly used `OUTPUT` rather than `RETURN`.

Fewer candidates provided correct solutions for the tests in the loop. Many solutions did not include code to check the length.

Solutions often correctly included simple comparisons statements such as:

```
FirstChar >= 'A' AND FirstChar <= 'A
```

In some cases, the comparison operators and/or the logical operator were incorrect. Many solutions converted the character to its ASCII code value, making a more complex solution and often the values used for comparison were not the correct ones. Many candidates attempted to use functions which convert to upper case such as Python's `upper()`, resulting in many incorrect comparisons such as:

```
if ThisChar == upper(ThisChar):
```

Few solutions checked the last four characters correctly. Many solutions included an incorrect comparison such as:

```
LastChars >= "0000" AND LastChars <= "9999"
```

Many good responses included loops that extracted each individual character (four right characters) and tested each one individually. Many responses correctly used substring functions from the candidate's chosen language. In a significant number of cases, functions were from another language was used, for example, the attempted use of `MID()` in a Python solution.

Many answers seemed to suggest a failure to understand the key similarities and differences between subroutines and functions. For a good solution for **parts 6(a)** and **6(b)**, it was essential that candidates appreciated the detail given in the table in the question rubric for the three subroutines given.

The contents of many weaker responses bore little relation to the question and some candidates appeared to mix up the requirement of this question with that of **part 6 (b)**.

(b) Most good responses recognised the need for a procedure header and some form of loop.

A prompt and input was included in many responses. Some responses did not first check on the return value from `WriteInfo()`.

A minority of candidates recognised that `GetInfo()` and `WriteInfo()`were functions. The following construct was widely seen:

```
Call GetInfo()
```

Candidates appeared to be much more familiar with functions than they are with procedures.

Many solutions used native `INPUT` statements, sometimes in addition to `GetInfo()`.

Many correctly recognised the need to call function `WriteInfo()` in both cases. Often only a single parameter was passed.

A significant number of responses included native file handling operations, which were not required.

Many responses indicate a lack of programming experience and solutions that combined program code and pseudocode were common. A small number of very weak responses did not include any recognisable program code.

(c) The majority of candidates gained almost full credit for this question.

Some responses suggested that candidates do not understand what is meant by the ‘module declaration’, evidenced by the fact that many of these responses attempted to include all the code for the function.

COMPUTER SCIENCE

Paper 9608/32
Written Paper

Key messages

Candidates need to show an in-depth study of the topics in the syllabus and make good use of appropriate technical terminology for this advanced theory paper. Candidates, who have studied the theory and have also practised the precise use of appropriate tools and techniques, were able to demonstrate successfully how they could be used to solve the problems set on the examination paper.

Candidates need to show good examination technique by responding to each question after carefully reading each in order to understand the exact requirements. Questions that state ‘Explain how….’ require a technical explanation of how to perform the task described in the question. Questions that state ‘Show your working.’ require candidate to show the intermediate steps in the calculation, as well as the final answer.

General comments

Candidates need to read questions very carefully before attempting to write an answer. For example, in **Question 5(c)(i)**, the instruction is ‘Describe **two** vulnerabilities that a malware can exploit.’, and not describe two possible effects of malware.

Comments on specific questions

Question 1

(a) (i) Many candidates calculated the denary value of the floating-point number correctly and showed their working. A common incorrect answer given was 6.875, provided by candidates who incorrectly assumed that the binary point was at the start of the mantissa.

(ii) The majority of candidates provided a correct reason.

(iii) The majority of candidates gave the correct normalised mantissa and exponent. A common incorrect answer was `0101` for the exponent.

(b) (i) Most candidates correctly converted +11.625 to binary. Normalisation proved more of a challenge with some candidates not showing the required working for obtaining the exponent.

(ii) This part of **Question 1** proved to be the most challenging. To gain full credit, candidates needed to show how –11.625 would be converted to a normalised binary number, either by calculating the binary values, or converting +11.625 to a negative number. A common error was not to mention the exponent in the conversion.

(c) Many candidates understood the problem of not being able to convert some denary numbers to exact binary values. Some candidates also correctly identified that the difference in value between the denary and the binary would increase, and become significant enough to be seen, if two such numbers were multiplied together and the answer output.

Question 2

(a) Most candidates were able to name a correct alternative method.

(b) Many candidates gave a suitable example of a situation where circuit switching would be used. Most candidates correctly identified the need for a dedicated circuit to be made available. A minority of candidates were able to successfully justify their choice.

Question 3

(a) (i) Most candidates were able to successfully complete the Karnaugh Map.

(ii) Many candidates correctly identified one group of four ones, $\overline{A}$, from the Karnaugh map. The second group of four ones, $\overline{B}$, proved more challenging for some candidates.

(iii) The majority of candidates correctly gave $\overline{A}$; a common wrong sum-of-products given was $\overline{A.B}$.

(b) (i) This question was generally answered well.

(ii) This question was generally answered well.

(iii) This question was generally answered well.

Question 4

(a) There were some excellent descriptions of the main steps in the evaluation of a Reverse Polish Notation (RPN) expression using a stack. Candidates who gave clear descriptions used the correct terminology of push and pop related to the process of evaluating the given expression. Common incorrect answers given by candidates included using more than one stack, or placing the operators on the stack.

(b) Many candidates correctly identified the contents of the stack at each stage of the evaluation and gained good marks. A few candidates did not demonstrate the process and gained zero or very few marks.

Question 5

(a) Understanding of why public and private keys were needed was shown by many candidates. Fewer candidates provided a technical explanation of how these keys were used to ensure that the email message remained private.

(b) Some candidates showed a good understanding of how asymmetric encryption was used to send a verified message. Candidates' explanations that included encrypting the message digest with the government department's private key and sending this well as the message, so that the sent message digest could checked against the message digest recreated by Sanjeet's computer gained full credit.

(c) (i) Many candidates were awarded full credit for this question. Some candidates incorrectly described the effects of malware rather than vulnerabilities that malware could exploit.

(ii) This question was generally well answered. The majority of candidates were awarded the mark.

Question 6

Many candidates correctly identified two other hardware devices. A common incorrect response was to include a device already mentioned in the question, for example, a sprinkler or a processor. The purpose of a sensor is to measure an identified property rather than sensing or detecting.

Question 7

(a) Many candidates correctly attributed at least four of the statements to a RISC or CISC processor.

(b) (i) Many candidates were awarded full credit for this question. Some candidates incorrectly identified parts of a computer.

(ii) Many candidates correctly identified that a large number of processors were required. Fewer candidates went on to describe that the processors worked collaboratively on the same task and they needed to communicate via a messaging interface.

Question 8

(a) Many candidates correctly identified two states for a computer process. Most candidates gave a description of the identified state. A minority of candidates provided full descriptions.

(b) This question proved challenging to many candidates. Those candidates who provided descriptions of how the use of the resource stated in the question could be maximised gained better marks.

COMPUTER SCIENCE

Paper 9608/42
Written Paper

Key messages

Candidates need to demonstrate their programming skills and algorithmic thinking in a variety of different ways. They need to show their ability to write object-oriented code and to use the methods appropriate to complete the algorithms described.

Candidates also need to show their understanding of different abstract data types, both by demonstrating how items were added and/or removed from them, as well as writing code to implement the given structures.

General comments

Many candidates clearly demonstrated their understanding of abstract data types and implemented these appropriately.

Some candidates found the object-oriented programming questions more challenging, with some candidates showing a lack of experience in writing constructors, get and set methods, and using these methods to write programs.

Candidates need to read the questions carefully to make sure they have met the required criteria, for example, where a question asks for a value to be returned, this should not be output instead. If a question asks candidates to use specific identifiers and names of parameters, then candidates should make sure they are using these in their answers.

Comments on specific questions

Question 1

(a) (i) Many candidates were able to add the nodes in the correct places, and many added the appropriate null pointers. Those candidates, who put the nodes in the incorrect places, often gained credit for putting in the correct null pointers. Some candidates did not follow the order of the data given and instead attempted to enter them numerically which produced an incorrect tree structure.

(ii) Many candidates answered this question well. Those who gave an incorrect tree structure for part **(i)**, were often able to produce the correct table for their tree. Some candidates incorrectly used the data values as pointers instead of using the array indexes. Many candidates gave appropriate null pointer values. One common error was to give index the value 0 as a null pointer. The number zero cannot be used as a null pointer because there is a data item with the index 0, therefore those nodes would link to this data item.

(b) **(i)** Candidates were required to demonstrate their knowledge of constructors in their chosen language. Many candidates understood the need to use their own constructor as described in the class diagram. Some incorrectly used a create method. It is important that candidates read the requirements of the question correctly, and they refer back to the information at the start of the question. In the class diagram, candidates were told that the `Grade` should be initialised to "`Fail`" and that the `FinalMark` should be initialised as `0`. Some candidates attempted to read data input by the user within the diagram, instead of storing the data required. The question also required the constructor to take the centre number and candidates number as parameters and then concatenate these; some candidates attempted to take the whole `PaperID` as one parameter, or take this as an input from the user, which did not meet the specification provided.

(ii) Many candidates were able to demonstrate a good understanding of the purpose of get and set methods. Fewer candidates were able to explain how they supported security and integrity, with many candidates providing generic description of these. A common error was for candidates to state that the methods could not be accessed out the class; this is inaccurate because the values can still be accessed. They cannot be accessed directly, and set and get methods need to be used.

(iii) A significant number of candidates were aware of the purpose of get methods and were able to write at least part of them accurately. Common errors included the output of values within the methods instead of returning them. Another included declaring a procedure to return a value instead of a function returning the value. Some candidates seemingly confused get and set methods, by reading in values from the user and attempting to assign these to the attributes.

(iv) In this question, candidates needed to return either true or false; these were often output instead, or assigned to a variable that was not then returned within the function. This question required a straight forward `IF` statement making use of the parameter as given in the question. Some candidates were unable to use an `IF` statement correctly, or did not make use of the parameter instead attempting to input a value into the function. Candidates must make sure they are using appropriate operators for their language, for example the use of <= instead of the mathematical operator ⩽.

(v) As with part **(iv)** this question required candidates to make use of the parameter values within an `IF` statement. Candidates needed to revisit the original specification at the beginning of **part (b)**. This question required the setting of the `Grade` attribute based on the value of the attribute `FinalMark`, and not the return of a value. Some candidates did not make use of this attribute. They attempted to read in a value to the method. A common error was for candidates to attempt to redefine the class's attributes within the set method, i.e. defining a new `Grade` and new `FinalMark` instead of using the attributes.

(vi) Candidates were required to use the methods they had defined in the earlier questions. Candidates were expected to create a main procedure to input values and then create an instance of their class. Many candidates were able to create an instance of an object in their chosen language. Fewer candidates were able to put together the information from the previous questions to send the appropriate data to the constructor. Some candidates were unable to use the get and set methods correctly in their language, for example, they attempted to send the object to the get and set methods as a parameter.

(c) Many candidates were able to demonstrate a clear understanding of the differences between linked lists and hash tables, and applied this appropriately to the context. The most common responses involved identifying that records could be directly accessed; this was quicker than a linked list. Fewer candidates were able to expand on this, for example, they explained how it was slower to follow the pointers and perform a linear search on the linked list. Some candidates confused the use of a hash table with hashing for encryption and discussed the security of the data. A common answer involved the use of memory, with candidates stating that a linked list required more memory because of the pointers, but they did not take into account the number of empty spaces that require declaring for a hash table to allow for sufficient space for all of the data.

Question 2

(a) The majority of candidates correctly identified that the first statement described a stack data structure.

(b) (i) (ii) There were mixed responses to this question. Some candidates correctly identified the stack structure as last in first out, but did not apply this to the given stack.

(iii) Candidates were given the requirements in the question i.e. the parameter that was required, and the values to return. It is important that candidates refer to back to the question whilst writing their answers to make sure they are meeting all the requirements. Some candidates attempted to read a value from the user instead of using a parameter, and many candidates did not actually return a value; either outputting the result, or assigning it to a variable, such as `Valid`, and then not returning it.

Candidates found the use of the `Top` global variable challenging, and often compared the top array element to `Top` instead of using it as the index.

A significant number of candidates were able to use the appropriate comparison to check the top of the array, using either its length, or the values assigned from previous questions to identify the size.

A number of candidates correctly assigned the new value to the top of the array, but did not then increment the `Top` counter.

Question 3

(a) This question was answered well by many candidates, who gave suitable features that they had used within their own programming experience. The most common responses included the use of colour coded text, auto-indent and auto-correct. Some candidates identified the use of indentation, and others, the use of comments as being features of an editor – whereas these are features provided by the programming language itself and not the editor.

(b) Many candidates identified the three types of test data using a variety of different names for each. The most common error was to identify testing strategies such as alpha and beta testing instead of types of test data.

Question 4

(a) (i) Answers were mixed to this question. The most common correctly identified error was line 26 (or the swapping of lines 28 and 30). Fewer candidates were able to correctly identify the other errors, giving a mixture of attempted corrections. When answering these questions, candidates should test the algorithm e.g. with example data, so they can trace each step and then see where and when the errors occur. If candidates have time, it is also recommended that they test their corrected algorithm.

(ii) The majority of candidates correctly identified a different searching algorithm.

(b) (i) Many candidates were able to gain marks for the completion of the trace table. The final mark was awarded least, with candidates stopping after the iteration and not providing the final iteration when the `FOR` loop runs for the second time. Candidates should refrain from writing ‘-’ in spaces where values are not changed because these indicate that that value ‘-’ is being assigned to the variables.

(ii) Many candidates found this question challenging. The question required the `FOR` loop in the algorithm to be converted into a `WHILE` loop. No other parts of the original algorithm required changing. Many candidates attempted to combine the `WHILE` loop with the `REPEAT UNTIL` loop by using two comparisons, and some candidates replaced the `IF` statement with the `WHILE` loop instead. Of those candidates who correctly used a while loop, often the criteria was incorrect, e.g. only using less than 4 instead of less than or equals to. Some candidates put the initialisation of the counter inside the `WHILE` loop which stopped the loop from working, and some candidates only incremented it when the IF statement condition was true. As with the error detection question, candidates should be testing their algorithms as they write them. They had already created a test table for the original algorithm, therefore could use the same data to test their amended algorithm.

(iii) Many candidates correctly identified the algorithm as a bubble sort. A significant number identified it as a selection sort.

(iv) This question was answered well with many candidates being able to give a different sorting algorithm to their first answer to **part (iii)**.