COMPUTER SCIENCE

Paper 9608/11
Theory Fundamentals

Key messages

Precision in answers is important. Vague answers and answers that repeat the information given in the question will not gain any credit.

Many candidates would benefit from a better understanding of the different command words used in the questions.

If large parts of a response are crossed out it helps if the replacement answer is written somewhere else in the script rather than above the crossed-out work, and there should be a clear indication of where to find any replacement answer. Using the page numbers printed on the question paper is good practice. It helps if rough work is crossed out.

General comments

Questions may have quite specific instructions and it is essential that candidates follow these instructions. In tick box and line drawing questions, candidates must put the correct number of ticks or lines.

The logic questions were usually completed successfully; questions about networks and translation software were more challenging.

Comments on specific questions

These comments should be read in conjunction with the published mark scheme for this paper.

Question 1

Almost all candidates correctly linked the commercial software licence to its appropriate description. There was considerable confusion between a Shareware licence and an Open Source Licence. Some candidates need to improve their understanding of these software licences.

Question 2

(a) Many of the responses to this question gave a description of the circle, for example, it has a thick black edge, rather than listing its properties, line width and line colour.

(b) There was some confusion between a drawing list and a library of shapes. Many answers described items that would be contained in a shape library, rather than describing a drawing list.

(c) Many candidates were able to describe the pixilation that would occur. Some responses did not make it clear that this happens when the image is enlarged. Describing a second drawback proved more challenging. Candidates who correctly identified that the bitmap would likely require more storage often found it challenging to describe why this was the case.

(d) Almost all candidates were able to correctly give three appropriate security methods. Passwords, biometrics and implementation of a firewall were the most popular correct answers.

Question 3

(a) There were a few completely correct trace tables. Few candidates made a reasonable attempt to trace this straightforward low level language program.

(b) Many candidates need to improve their understanding of indirect addressing. A common incorrect answer was `LDI 200`.

(c) (i) Almost all candidates were able to correctly convert the operand to binary.

(ii) Many candidates understood that the number of op codes was 2^8. A common incorrect answer was 255, the maximum value that can be held in 8 bits, rather than 256, the number of distinct values.

(d) Few candidates understood the flags that can be set in a status register.

Question 4

(a) There were some excellent correct truth tables. There was some confusion with the NOR gates. Some candidates need to improve their understanding of the different symbol for an XOR gate and a NOR gate.

(b) Many candidates were able to correctly identify three logic gates not used in the diagram.

Question 5

(a) The two many-to-many relationships were generally recognised and each one split into the appropriate one-to-many relationships. There was considerable confusion about which was the 'many' end of each relationship. Candidates also need to understand that they must use a recognised notation for the relationships.

(b) Most candidates found this question challenging. Candidates need to improve their understanding of what is meant by referential integrity and be able to apply it to a given set of tables.

(c) Some candidates were able to correctly identify that a table could be created using a DDL statement. Most candidates found it very challenging to then list two other items that could be created using the DDL.

(d) There were a few good attempts at this DML statement. The SELECT and FROM clauses were straightforward. Many candidates found the WHERE clause more challenging because of the need to include leaders who spoke either French or English.

Question 6

(a) Some candidates did not read the question carefully and missed that the value 16 had been entered. Some candidates should be aware that quotation marks are not required on output when it is not part of a programming statement. In questions of this type, it is important that the text is copied exactly from the question, including both case and spelling.

(b) There was some confusion between lines 14 and 15 where the function is defined and line 12 where the function is called.

(c) Many candidates correctly identified the two variables. Some candidates need to understand that JavaScript is case sensitive, so 'hour' and 'Hour' are two different identifiers.

(d) Some candidates correctly identified the first two statements. Many candidates need to improve their understanding of PHP code and what is meant by a conditional statement.

(e) Most candidates selected appropriate terms from the given list.

Question 7

(a) There were some interesting and thoughtful answers to this question. Some candidates need to understand that for two marks, two distinct points must be made. Some answers about Marina's actions were the same point written in two different ways.

(b) There were some insightful answers to this question. Many answers discussed whether Doug had the right to ask this of his team, and whether the team had the right to refuse. Some candidates seemed to assume that it was his fault that the project was behind schedule. Some good answers included statements about work-life balance and the workers' mental health.

(c) There were several interesting responses to this question too. Some candidates need to understand that vague statements which repeat information given in the question such as, 'Unethical because Debbie discusses her concerns on a public forum' are not enough for credit at this level. Why it is unethical must be described.

Question 8

(a) (i) The first two language translators were usually correctly identified. Some candidates need to improve their understanding of the differences between compilers and interpreters.

(ii) The question asked for two benefits. Many of the answers gave descriptions without saying what the benefit was.

(b) (i) Many candidates understood that there is often a limit on the size of a file that can be attached to an email. Some candidates found giving a second reason more challenging. A popular correct choice was the time taken for transmission

(ii) Answers to this question contained a lot of generic descriptions of lossless compression without applying it to the text file described in the stem of the question. Some candidates described the compression of a source code file (program) rather than a text file.

Question 9

(a) Many candidates found this question challenging and need to improve their understanding of public and private IP addresses. Statements such as, 'one is private, and the other is public' will not gain any credit at this level of study.

(b) Some candidates found identifying the most appropriate term very challenging. The one most likely to be correct was a firewall. There is a need for improved understanding of the differences between routers, gateways and servers.

COMPUTER SCIENCE

Paper 9608/12
Theory Fundamentals

Key messages

Precision in answers is important. Vague answers and answers that repeat the information given in the question will not gain any credit.

Many candidates would benefit from a better understanding of the different command words used in the questions.

If large parts of a response are crossed out, it helps if the replacement answer is written somewhere else in the script rather than above the crossed-out work, and there should be a clear indication of where to find any replacement answer. Using the page numbers printed on the question paper is good practice. It helps if rough work is crossed out.

General comments

Questions may have quite specific instructions and it essential that candidates follow these instructions. some candidates had frequently put the incorrect number of ticks or lines in tick box and line drawing questions.

Tracing the low level language program and completing the truth table were usually completed successfully; questions about networks and utility software were more challenging.

Comments on specific questions

These comments should be read in conjunction with the published mark scheme for this paper.

Question 1

(a) Almost all candidates correctly completed the truth table.

(b) There were many correct answers to this question. A common incorrect answer was the use of a NOR gate instead of the OR gate.

Question 2

There were a few completely correct answers to this question. A common incorrect link was from the Current Instruction Register (CIR) to the description 'holds the current value in the Index Register'.

Question 3

(a) A significant number of candidates found tracing this low level language program challenging. There were two distinctly different incorrect trace tables. Some candidates had misinterpreted the `LDI` instruction on line 50 and had loaded the **value** 100 into the accumulator instead of the **contents** of address 100. Other candidates had completed the first iteration correctly, but on the second loop at the `ADD 102` instruction on line 53 had used the original value in address 102 instead of the updated value.

(b) Many candidates need to improve their understanding of the different modes of addressing. A popular incorrect response was `LDD 103`.

(c) (i) Almost all candidates correctly converted the value to binary.

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

(d) Many candidates correctly identified two other modes of addressing. These are technical names, and it is expected that they would be written correctly. Some candidates need to be careful with spelling.

Question 4

(a) The question asked for the drawbacks of a file-based approach. Some candidates instead wrote about the benefits of a relational database. Many statements were too vague and applied equally to a file-based approach or a relational database. There was considerable confusion between security and privacy, with a misconception that security could not be applied to a file.

(b) Most candidates were able to state that there was repetition of data and give an example from the table. Some candidates need to understand that giving the same answer three times with a different example is only one reason not three. Another popular correct answer was the lack of a primary key. Many candidates found it more challenging to give a third reason.

(c) There were some completely correct DDL scripts. Some candidates displayed little or no grasp of writing SQL statements. These skills are best learnt through the practical process of writing and testing SQL statements within a DBMS environment.

Question 5

(a) Many candidates recognised the identifiers in the code. Some candidates need to understand that JavaScript is case-sensitive, so 'Area' and 'area' are not the same identifier.

(b) Some candidates need to improve their understanding of the use of functions in programming code; repeating the text from line 04 of the program is not enough to explain the purpose of the code on line 08.

(c) There were many correct answers to this question. A common error was the omission of the colon. Some candidates should understand that quotation marks are not required on output when it is not part of a programming statement.

(d) Almost all candidates recognised the symbol as logical OR.

(e) (i) Many candidates correctly identified the data validation as a presence check.

(ii) Many candidates were able to identify two other validation checks that could be used.

(f) Many candidates found this question challenging. The question needed to be read carefully as there were quite specific statements about HTML, JavaScript and PHP. Some candidates need to improve their understanding of the use of each of these, particularly the use of HTML tags and JavaScript code.

Question 6

(a) There were some interesting responses to this question. Some candidates need to understand that two distinct points are required for the two marks. Some answers about Latifah's actions were the same point written in two different ways.

(b) There were many interesting responses to this question too. Many answers discussed whether a computer security expert would be required for a banking application.

(c) There were some very insightful answers to this question. Many candidates appear to have a good understanding of the use of personal data by social media websites. Some candidates need to understand that vague statements which repeat information given in the question such as, 'Jason

should take further action' are not enough for credit at this level. The further action must be described.

Question 7

(a) (i) The whole of **Question 7(a)** required the application of knowledge. The stem of the question stated that a photographer was storing images, so the uses of the various utility software needed to be applied to this context. There were many good answers about how backup software would be used to make copies of the images in case of loss.

(ii) Explaining how the photographer would use defragmentation software was more challenging. Many candidates found it difficult to explain how it would improve disc access time. Statements such as, 'it would make the disk work faster' are not precise enough, there needs to be the idea of improving the time taken to access and load files.

(iii) Many of the answers to this part question lacked sufficient detail. Statements such as, 'disk repair software repairs the disk' are too vague for credit.

(b) Many candidates were able to correctly identify three items that would be stored in the header of a bitmap file. Some candidates found it more challenging to identify four **different** items.

(c) This question also required the application of knowledge to the images used on the website. Many responses described lossy or lossless compression in generic terms rather than justifying why one would be more appropriate than the other in this context.

Question 8

(a) There were many correct answers to this question.

(b) Many candidates need to improve their understanding of what is meant by a digital signature. There was considerable confusion between a digital signature and a digital certificate. Some responses described the use of an electronic tablet and stylus when signing something.

Question 9

(a) Many candidates found this question challenging. Stating that wireless networks do not need cables is not a benefit, it is a description of a wireless network. The benefit is that because there are no cables, devices can be mobile. Some responses described drawbacks of wired networks, with no reference to any corresponding benefit of a wireless network.

(b) Some candidates found describing drawbacks of a wireless network more straightforward. Weaker security and interference were popular correct answers. Some responses to this question also described benefits of wired networks, with no reference to any corresponding drawback of a wireless network.

COMPUTER SCIENCE

Paper 9608/13
Theory Fundamentals

Key messages

Precision in answers is important. Vague answers and answers that repeat the information given in the question will not gain any credit.

Many candidates would benefit from a better understanding of the different command words used in the questions.

If large parts of a response are crossed out, it helps if the replacement answer is written somewhere else in the script rather than above the crossed-out work, and there should be a clear indication of where to find any replacement answer. Using the page numbers printed on the question paper is good practice. It helps if rough work is crossed out.

General comments

Questions may have quite specific instructions. It is essential that candidates follow these instructions. Some candidates had frequently put the incorrect number of ticks or lines in tick box and line drawing questions.

The truth table and tracing the low level language program were usually completed successfully. The questions about codes of conduct and video compression were more challenging.

Comments on specific questions

These comments should be read in conjunction with the published mark scheme for this paper.

Question 1

(a) Many candidates were able to correctly identify the four validation types.

(b) (i) Almost all candidates were able to correctly identify another method of protecting the integrity of the data. The most popular correct answer was verification.

(ii) Many candidates were able to correctly identify two correct measures to protect the integrity of the data during transmission. The most popular correct answers were parity and checksum.

Question 2

(a) There were many completely correct answers to this question. Some candidates interpreted the final XOR gate as a second OR gate.

(b) Almost all candidates were able to name the four logic gates used in the circuit.

Question 3

(a) Many candidates were able to correctly trace this code. A very common incorrect response was the inclusion of quotation marks around the output. Some candidates should understand that these are not required on output when it is not part of a programming statement.

(b) There were some good answers to this question. Some candidates need to read the question carefully. The question asked for modes of addressing that had **not** been used in the program.

(c) (i) Almost all candidates correctly converted the value to binary.

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

(iii) Some candidates need to improve their understanding of two's complement binary. A frequent incorrect answer was 217, where candidates had treated the binary value as an unsigned integer rather than a two's complement value.

(d) There were a number of excellent complete answers to this question. Some candidates need to ensure they understand the use of brackets in register transfer notation. Some answers showed the registers correctly but with the brackets omitted.

Question 4

(a) (i) There were some good answers where candidates wrote about installing device drivers and the management of buffers. Almost all candidates found listing three device management tasks challenging. Many answers listed general operating system management tasks rather than tasks to do with device management.

(ii) The most popular correct answer here was dealing with interrupts. Almost all candidates found listing three error detection and recovery management tasks challenging. Many answers referred to errors in code.

(iii) Many candidates were able to correctly list two other operating system management tasks. Some candidates listed hardware management or peripheral management which is the same as the device management task given in the question.

(b) (i) There were many completely correct answers to this question. A common incorrect answer was to identify an IDE as a utility program.

(ii) Almost all candidates were able to correctly give two other utility programs. Disk formatter and file compression were the most common answers.

Question 5

(a) Almost all candidates were able to correctly identify the variables. Care must be taken when writing the variable names in the answer. Variables in PHP are case sensitive and must start with the $ symbol.

(b) Almost all candidates were able to correctly locate the lines where an output was produced.

(c) Candidates found describing the purpose of the code quite challenging. Answers often referred to concatenating the variables, rather than the contents of the variables and omitted the space between the two names. The full line of code also includes the assignment to the `$result` variable. There was little mention of this.

Question 6

(a) There were many completely correct answers to this question. A common incorrect answer was the reversal of steps B and D.

(b) Candidates found this question challenging. Many answers provided great detail about how the DNS converted a domain name into an IP address, which did not answer the question. Few responses included examples of a domain name and an IP address.

Question 7

(a) Many of the answers to this question concentrated on the general features of a compiler rather than the drawbacks of using a compiler. There were lots of statements about an executable file

being produced with little understanding that such a file would not be produced if there were any syntax errors remaining in the code.

(b) Descriptions of the two licences were generally very good. Many candidates were able to write about the difference in the availability of the source code, the cost implications, and whether the source code could be modified or not.

(c) Many candidates found this question challenging. There was a lot of confusion between a professional code of conduct and the production of professional program code.

Question 8

(a) Almost all candidates recognised that there would be two one-to-many relationships, one between `EMPLOYEE` and `PROJECT_TEAM` and one between `PROJECT` and `PROJECT_TEAM`. Many candidates need to improve their understanding of the notation used for E-R diagrams. The 'crows foot' symbol for the many end of the relationship was often on the wrong end of the line.

(b) Identifying the primary keys and corresponding foreign keys was correctly done by almost all the candidates. Explaining how these keys were used to link the tables was more challenging.

(c) There were some good attempts at writing this DDL statement. Candidates must take care when using names given in the question that they write them exactly as given.

(d) This DML statement involved just a single table and there were many completely correct answers.

Question 9

(a) (i) Almost all candidates were able to correctly identify the term needed.

(ii) Almost all candidates were able to correctly identify the term needed.

(b) Candidates found this question challenging. There was considerable confusion between the two terms and a lack of precision in answers. Statements such as, 'if there are different pixels that are the same colour' are not enough. It must be clear when describing Run-Length Encoding (RLE) that the identical pixels must be adjacent to each other within a frame (spatial redundancy) and that for inter-frame compression (temporal redundancy) the identical pixels must be in exactly the same position in consecutive frames.

COMPUTER SCIENCE

Paper 9608/21
Problem-Solving and Programming

Key messages

The emphasis for this paper is on the application of practical skills. Candidates need to have developed these and be able to apply them to the scenarios presented if they are to achieve high marks.

This is a technical subject and makes use of many technical words and phrases. These have specific, defined meanings and it is important that these are used correctly.

It is important that candidates writing program code use the correct syntax for their chosen language.

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

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

General comments

A few excellent programming solutions were seen, but most responses suggest most candidates had little or no programming experience and the majority of candidates attempted few questions. There were a high number of 'no responses' to the programming and pseudocode questions.

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

Candidates need to understand that no marks are awarded for programming answers that do not use one of the three permitted languages.

If answers are crossed out, the new answers must be written clearly so that the text may be read easily and the correct mark awarded.

Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely important to cross out this text.

Comments on specific questions

Question 1

(a) (i) Few candidates achieved both marks. Those achieving a mark stated that ASCII characters are represented by a unique value.

(ii) Few candidates provided a correct response with many not attempting this question. A whole range of incorrect values were seen.

(b) (i) Again, few candidates gave the correct response. The remainder either gave no response or the correct values.

(ii) Few candidates provided a correct response. Various incorrect answers were given.

(c) Most candidates achieved at least one mark. Many identified that there was no error in statement 4 and many identified that there were not enough characters in the string 'Cat' to extract 4 characters from the left. A few candidates simply wrote 'NO ERROR' for each case.

Question 2

(a) The majority of candidates either did not answer this question or produced something unrelated to a program flowchart.

(b) Of the small number of candidates that attempted this question, there seemed to be some confusion as to what was required. Some were comparing consecutive elements with each other rather than setting the initial element as max, then comparing this with each element of the array and setting max to the current element if higher than max. Some mentioned sorting the array to find the highest, which is an acceptable solution, but did not describe the sorting mechanism.

Question 3

(a) Few candidates attempted this question. Many gained marks for correctly identifying some of the parameters. The selection diamond was not seen on any response. A single response correctly showed the iteration arrow for `Module_X`.

(b) (i) Many achieved the mark for identifying the module is a function, but many could not give a correct justification.

(ii) Few candidates achieved any marks. Candidates who correctly wrote the procedure name then lost marks by stating a return value, only required in function headings. `ParZ` was passed by reference and therefore needed `ByRef` in the declaration.

Question 4

(a) Few candidates achieved 3 or more marks for this question, and many did not attempt this question. Of those making an attempt, some good solutions were seen but many lost marks due to writing pseudocode statements or using incorrect syntax for their chosen language.
Common errors were the incorrect type of brackets used in Visual Basic solutions, and an array subscript should be in round brackets ( ) and not square brackets [ ].

(ii) Some Python solutions used functions such as Left and Mid, which are not valid for Python.

(iii) Many incorrect assignment symbols were seen in both Python and VB.

(b) This response required candidates to have had experience of writing programs and therefore having the experience of using an IDE. The lack of correct responses indicated that most candidates are not familiar with coding.

(c) (i) Most candidates were unable to answer this question or gave an unrelated response. Again, candidates must have experience of programming to be able to identify the stages of the program development life cycle. Some gained a mark for giving a description of a task.

(ii) As with the previous question, due to a lack of programming experience, candidates were generally unable to answer this question.

Question 5

(a) (i) 12 per cent of candidates achieved at least one mark for this question. Many did not attempt it. Of these, marks were gained mostly for the prompt and input of a number and for indicating a loop is needed. Many used the incorrect range for the Random function, by using 21. This would select the range 0 to 20. Some omitted the INT function to ensure an integer was used.

Some candidates lost marks for not ending constructs in loops or IF statements. These candidates tended to be those familiar with Python where endings are not required. However, when writing pseudocode, it is essential that candidates clearly indicate where a loop or selection statement ends.

(ii) A few candidates stated correctly that a check should be made that the input is within the range 1 to 20. Many just stated the name of a validation method, such as type check or range check which gained no credit.

(b) (i) Most achieved no marks. The question asked for the type of testing. Those that attempted generally gave any type of testing method they'd heard of, in particular white box/black box testing. Less than 5 per cent stated the correct answer.

(ii) A few candidates mentioned a simple line of code should be entered in the function but did not state that a known value should be returned. Many did not attempt the question.

(iii) Vague responses such as' **turns** program code into machine type code', gained no credit. As with previous questions, it is essential that candidates have experience of writing and executing their code. This helps gain an understanding of the process of producing executable code from a high level language.

Question 6

(a) This question tested candidates understanding and use of selection and logic statements in their chosen language. Most did not attempt to answer this. The responses seen that lost marks, generally used incorrect syntax, or used functions incorrectly.

(b) Most did not attempt to answer this question. Of the few that did, some excellent solutions were seen. A few missed out a loop and as in the previous pseudocode solution, selection and loop statements were not ended correctly.

Candidates should also be reminded that they must close the file that has been opened.

(c) This question required a programming solution which included string and array manipulation. The better candidates produced their solution in Python using a range of different features of the language. Some used slicing to extract substrings and some made use of the split method to separate each substring.

As before, most candidates did not attempt to answer this question.

COMPUTER SCIENCE

Paper 9608/22
Problem-Solving and Programming

Key messages

The emphasis for this paper is on the application of practical skills. These skills involve analysing and understanding the requirement as well as designing and presenting a solution. Requirements are often presented using a scenario description. Candidates need to be able to identify the key elements of each requirement (for example, the need for an iterative structure) when designing their solution. The development of these skills requires practice.

This subject makes use of many technical words and phrases. These have specific, defined meanings and they need to be used correctly.

Answers should be as precise and specific as possible. Candidates should familiarise themselves with the meanings of the command words used in this paper and form their answers accordingly. Candidates should also be aware that answering a question by simply repeating phrases from the question will not gain marks.

TI is important that candidates are familiar with the fundamental programming concepts. Lack of understanding is often illustrated by the confusion between a literal value and an identifier, or the misuse of `OUTPUT` in place of `RETURN`. Many candidates appear unaware of the use of parameters, often replacing parameters to a subroutine with a series of prompts and inputs within the body of the subroutine itself.

Candidates need to read and understand each question before attempting to answer it. Questions may address topics in many different ways, and it is often necessary to apply knowledge in a specific way if marks are to be gained. It should not be assumed that simply because a question contains some recognised terms that it is the same question that has appeared in previous papers.

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. As in previous sessions, no marks were awarded for programming answers that did not use one of the three languages given in the syllabus. A significant number of candidates demonstrated skill levels suggesting they had little programming experience.

If answers are crossed out, the new answers must be written clearly so that the text may be read easily and the correct mark awarded.

Many candidates make use of blank pages for rough work when preparing their final answer. In these cases, it is extremely important to cross out this text.

If typed answers are provided then it is very helpful if these are organised so that individual answers do not span page breaks. This is particularly important for programming answers. If the question involves completing a table, they typed answers should clearly indicate any unfilled rows.

Comments on specific questions

Question 1

(a) Many candidates provided fully correct answers. ‘Corrective and ‘Adaptive’ were the most commonly identified types of maintenance, and where these were given, most answers then went

on to give a correct reason. 'Perfective' was seen much less frequently, and the reason given for this type was often not clear.

Reasons for corrective maintenance often focused on just the testing and omitted than the correction of the error.

Answers referencing computer science terms not related to maintenance were common.

(b) The question asked **why** characters need to be expressed in ASCII or Unicode. Many candidates simply provided an answer describing **what** encoding meant, such as 'each character is represented by a unique number'. Candidates should expect that the emphasis of a question should change from series to series and that simply repeating an answer from a past paper will not always be successful.

Only a small number of answers addressed the fact that the encoding system complied with a standard and would therefore be recognised by all programs.

(c) Successful responses referred to the character's use as a separator between data items. A small number of answers described the use of a special character (one that would not appear in the data). Very few answers adequately described the role of the separator in terms of simplifying the extraction algorithm.

Several answers referred to a separator being used between lines in the file, and in many cases, it was the user that was reading the line and extracting the individual data items, rather than a program.

(d) A wide range of marks was awarded. Some excellent answers were seen, where comprehensive error descriptions were provided. Many descriptions were too brief, to the point where it was unclear whether the actual error had been identified. An example of this would be 'invalid function' which is saying little more than had already been stated in the question.

The second 'No Error' answer was the most often correct, followed by the description for the error in the second statement.

The last statement seemed to be the most challenging. A popular assumption was that the identifier `Index` was an Integer; many stated that operator and could not be used with an integer but failed to mention the value `3`.

Several candidates simply offered 'logic error' or 'run-time error' for every statement.

Question 2

(a) There was a wide range of responses to this question, with many full-mark responses provided.

Most identified 'Overload' as the global variable. The last two answers proved too much of a challenge for many.

(b) Most candidates gained marks for the START and END, followed by the initial assignments. Many went on to also gain the mark for the 'Is Status = TRUE' test together with the corresponding assignments.

Many solutions had incorrect flow lines. Some candidates did not label the lines coming out from decision (diamond) boxes. A common mistake was to omit the 'NO' line from the 'Is Overload = TRUE?' decision (diamond) box.

Question 3

(a) (i) This question part was generally, well answered. 'Selection' and 'Iteration' were the most popular correct answers.

(ii) Many candidates achieved four marks from the available five, with the mark for the selection diamond being the one most often missed.

The double-headed arrow for the BYREF parameter was often missed, or occasionally shown as two separate parameter arrows. A small number of candidates seem unclear as to the significance of the circle on each parameter arrow, and whether it should be shaded or not.

(b) Many candidates seemed unfamiliar with the three stages of the program development cycle. Many different stages were often given, but often the core 'Design – Code – Test' was absent.

Few responses illustrated the cyclic nature of the process.

Some candidates offered a variant of the diagram used for the previous question. There were many blank responses and a variety of concepts from other parts of the syllabus such as Input – Process – Output.

Question 4

(a) This question split candidates into two distinct groups. Some good answers were seen, often gaining full marks for accurate and complete solutions. Many candidates struggled to write the required pseudocode.

While some candidates produced a textbook solution using only six lines, many solutions spilled into a second column.

Several solutions omitted the initialisation (MP1) in favour of an `INPUT` statement.

Although the question asked for a post-condition loop, many did not use a `REPEAT ... UNTIL`, opting instead for either a `FOR` or `WHILE` loop.

The loop condition frequently contained the error as illustrated below, which would stop after the first pass through the loop:

```
num ← 101
REPEAT
   <looped statements>
UNTIL num > = 100 AND num < = 200
```

Solutions using a `FOR` loop frequently included a separate statement to increment the loop counter.

Many candidates demonstrated a lack of understanding of the back-arrow assignment symbol through use of statements such as:

```
UNTIL num ← 201
```

A range of attempts at trying to test for an odd number was seen (MP3). The simple method of starting with the first odd number in the range and adding 2 each time around the loop was common but by no means universal. The use of `MOD()` was common. Some solutions extracted the last digit using string handling functions and others divided the number by 2 then attempting to check if the result was an integer.

(b) There were some very good solutions, especially in Python.

Although many solutions stated VB or Python, the answer given was clearly pseudocode. Some answers contained a mixture.

VB responses commonly lost the first mark point for incorrect syntax: the use of `RETURNS` instead of `AS`.

Many solutions included the re-declaration of the array as a local variable so were prevented from gaining MP2

MP3 was gained by many solutions.

VB solutions regularly lost marks from the use of incorrect brackets for the array index (MP4). Python solutions often incorrectly used a single bracket for the 2D array syntax.

Many solutions included an attempt at MP5. A common mistake was in the logical combination of the tests for Par1 and Par2.

A frequent mistake was to use an `ELSE` statement inside the `FOR` loop which set the return value to –1 each time there was no match with Par1 and Par2, so effectively 'overwriting' a correct value.

Question 5

(a) (i) Pretty print and Indentation were common correct responses. Many general computing terms were offered, and a significant number of candidates made no attempt at this question.

(ii) The context in this question was given as initial error detection but this was missed by many who offered general debugging and fault-finding techniques.

(b) (i) Answers tended to be awarded either both marks or neither. An exception were those answers where the Boolean value was correctly declared but then `OUTPUT` rather that `RETURN` was used.

Many answers included a declaration statement for the first mark, but often this declared `UserID`, which was parameter to the function.

(ii) A small number of excellent solutions were seen. Many answers struggled to achieve any marks, with many candidates making no realistic attempt at answering the question. Many pseudocode answers were seen.

Despite the simple scenario, many solutions lacked any sort of loop to allow the user to 'try again' (MP2). A common mistake when using post-condition loops was to omit the initialisation of the loop counter.

Most gained at least the mark for the prompt and input of the password (MP3)

Many candidates correctly wrote the function heading with `UserID` as the parameter but then went on to prompt and input the `UserID`.

Many solutions attempted to compare the value of `UserID`, using an `IF UserID = 'Guest' THEN...` construct, but several omitted the quotation marks around the string 'Guest' and/or failed to return `TRUE` at that point or to avoid the rest of the algorithm.

Although the line which performs the validation is given in the question, many copied this incorrectly and missed out the ...`AND TODAY()` part (MP4). Many resorted to the simpler, but incorrect:

```
IF password = valid THEN
```

Solutions that did contain the correct validation line usually also gained MP5 and MP6.

Question 6

(a) As for **Question 5 (b) (ii)**, A small number of excellent solutions were seen. Many answers struggled to achieve any marks, with many candidates making little or no attempt at answering the question.

Many pseudocode answers were seen.

A simple scenario based around a linear search of an array; however, many solutions included some form user input.

Many solutions did not initialise the variables used for the `Count` and `Total` (MP1).

Most solutions included a count-controlled loop (MP2). A common mistake was to attempt to use the array itself as a loop counter as illustrated:

```
FOR StockID() = 1 TO 10000
```

The incorrect inclusion of a comma in the number 10 000 was not uncommon.

Many solutions contained successful attempts to skip elements containing the initial value (MP3). Occasionally this took the form of a `WHILE...` rather than an `IF...` which would have generated an endless loop.

Many did not correctly calculate the overall stock and simply gave an expression of the form:

```
Total = Cost * Quantity
```

Many answers omitted the final output (MP5)

(b) The introductory paragraph as for **Question 6 (a)** applies equally here.

Many solutions started with a correct function header but in many cases, there was no function end (MP1). Some candidates attempted to pass a parameter with a name such as `ThisFile.txt`

Solutions that included a reasonable attempt at file handling usually gained MP2.

Many correctly opened the file in read mode but in very many cases the corresponding `CLOSEFILE` was missing (MP3). A common mistake was to treat the parameter name as a literal string by enclosing it in quotation marks when used in the `OPEN` and subsequent file operations.

Addressing MP4, a common mistake was to test the filename itself, rather than the file contents. Another mistake was to compare `EOF(filename)` with an empty string rather than a Boolean value.

Many candidates addressed M6 via a conditional loop.

MP7 was frequently missed due to the use of incorrect parameters when calling `Unpack()`.

MP8 required a check on whether the file was empty or not after reaching the end of the array but few addressed this point.

Some candidates made this question more complicated than it was by trying to write additional functions not required, such as `Unpack()`.

(c) Previous **Question 6** introductory paragraphs apply.

Frequently a loop was missing from the solution (MP1)

MP2 was generally present, although this had to be within a loop to gain the mark. and this was often not the case.

Many correctly checked the length of the file name (MP3). Some lost marks for incorrect range comparisons (< instead of < =) or for incorrect logic (AND instead of OR).

'Impossible' comparisons were seen frequently, for example:

```
IF LENGTH(Filename) < 4 AND LENGTH(Filename) > 10 THEN
```

MP4 was one of the easier marks and often achieved.

Many VB solutions achieved MP5 for extracting a character using MID and the Python solutions using the index reference `filename(x)`.

Mark points 6 and 7 were usually addressed. Solutions based on built-in language functions or methods were generally more successful than those based on range comparisons.

Common errors seen included:

- missing quotes around the characters in comparison statements (both alphabetic and numeric)
- only testing uppercase and missing out lowercase (or vice versa)
- incorrect logic
- incorrect ASCII values used.

Many included the output of a warning message and re-input the file name following an invalid attempt, but then missed out the return of the filename. (MP8)

COMPUTER SCIENCE

Paper 9608/23
Problem-Solving and Programming

Key messages

This paper addresses the application of practical skills. These skills involve analysing and understanding the requirement, often presented in this paper via the use of a scenario description, as well as designing and presenting a solution. Candidates need to be able to identify the key elements of each requirement (for example, the need for an iterative structure) when designing their solution. The development of these skills requires practice.

This subject makes use of many technical words and phrases. These have specific, defined meanings and they need to be used correctly.

Answers should be as precise and specific as possible. Candidates should familiarise themselves with the meanings of the command words used in this paper and form their answers accordingly. Candidates should also be aware that answering a question by simply repeating phrases from the question will not gain marks.

Familiarity with fundamental programming concepts is vital. Lack of understanding is often illustrated by the confusion between a literal value and an identifier, or the misuse of `OUTPUT` in place of `RETURN`. Many candidates appear unaware of the use of parameters, often replacing parameters to a procedure or function with a series of prompts and inputs within the body of the subroutine.

Candidates need to read and understand each question before attempting to answer it. Questions may address topics in many different ways, and it is often necessary to apply knowledge in a specific way if marks are to be gained. It should not be assumed that simply because a question contains some recognised terms that it is the same question that has appeared in previous papers.

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. As in previous sessions, no marks were awarded for programming answers that did not use one of the three languages given in the syllabus. A significant number of candidates demonstrated skill levels suggesting they had little programming experience.

If answers are crossed out, the new answers must be written clearly so that the text may be read easily.

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

Comments on specific questions

Question 1

(a) A few candidates stated pseudocode for the programming language used despite having been told that programming code must be used.

(b) (i) Many candidates did not correctly identify the loop structure. They stated that it was a while or repeat loop which was not enough to gain this mark.

Only a small number of candidates gained both marks available.

(ii) Most candidates gained at least one mark and a significant number gained both marks. A common mistake was giving the reason as 'value not found in array', which was not enough in this instance, and they did not identify that the value was not found when all the array elements had been checked.

(c) This question part was generally well answered. Most candidates achieved at least one mark for this question with a significant number achieving all three marks available.

Question 2

(a) Not well answered. Very few candidates achieved both marks available with the second mark point being the one which proved most elusive.

(b) (i) Many candidates did not give a detailed enough answer to gain the mark for this question. For example, just giving the reason 'as easier to understand' instead of 'easier to understand the purpose of the identifier'.

(ii) A range of marks awarded but few candidates obtained all three marks available. A significant number giving the example already given in **(b)(i)** which was not enough.

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

Question 3

(a) Many candidates referred to program instead of problem in their answer.

(b) Many candidates gaining at least two marks with a significant number gaining all four marks. A common mistake was in the check made for the upper bound of the loop which meant either too few or too many values would have been output.

(c) A wide range of marks was awarded and a significant number of full-mark answers were seen. Common mistakes were the use of a repeat loop instead of a while loop and calling `ReCheck()` multiple times.

Question 4

(a) (i) Most candidates gained some marks for their trace tables with a wide range of marks being awarded. A significant number achieved all five marks available. The mark point for the last stage of the trace table was the one which most candidates did not achieve. They did not set `Result` to `-1` and then `0`.

(ii) Most candidates successfully answered this question part.

(b) (i) Few candidates identified and explained one of the two errors in the program. They were only required to explain one of these errors.

(ii) Few candidates gained the one mark available here for describing how the algorithm could be corrected. This was mainly due to few candidates correctly identifying an error in the algorithm asked for in **(b)(i)**.

(c) A well answered question with many candidates gaining the two marks available.

Question 5

(a) There were some very good solutions where the full 8 marks were awarded. Many candidates made a reasonable attempt at writing the pseudocode required with nearly all of these obtaining some marks.

The most common mark awarded was for the function header (MP1) with many candidates gaining the marks for initialisation (MP2) and for using a loop (MP3). The correct use of the `MID` function (MP4) was also often achieved.

A common mistake was not using the `STRING_TO_NUM` function (MP5) appropriately or not at all or not concatenating (MP7 and MP 8) strings correctly.

(b) Many of the answers candidates gave were too vague or the type of check given was the same for both checks. No mark was given where candidates just stated **type check** with no further details.

Only a few candidates obtained all four marks available with a significant number either being awarded no marks for their answer or making no attempt.

Question 6

(a) Many answers struggled to achieve any marks, with many candidates making little or no attempt at answering the question.

Many solutions included a conditional loop (MP1) but few of these correctly set the loop termination conditions correctly (MP7).

Few candidates correctly opened and closed the file correctly (MP3).

Few candidates correctly achieved the three linked mark points for testing if a file was empty (MP4) and if not empty, confirming overwrite (MP5 and MP6).

(b) A small number of excellent solutions were seen. Many answers struggled to achieve any marks, with many candidates making no realistic attempt at answering the question. Some pseudocode answers were seen.

A simple scenario based around a linear search based on a substring.

Although many solutions correctly used a loop to check each element of the `StockID` array (MP3) only a small proportion of theses extracted the correct substring from the current array element (MP4)

Most candidates who attempted the questions correctly achieved the marks for the procedure heading (MP1).

Many solutions did not initialise the variables used for the `Count` and `Total` (MP2) and misunderstood the requirement to use the `Cost` and `Quantity` arrays when calculating the `Total` (MP5).

Many solutions did not produce the outputs in the required manner.

(c) A small number of excellent solutions were seen. Many candidates made no realistic attempt at answering the question. Some pseudocode answers were seen.

Many solutions started with a correct function header but in some cases, there was no function end (MP1). Some candidates attempted to pass a parameter where none was required.

Many solutions also achieved the declaration and initialisation mark (MP2) along with the correct use of a for loop to iterate through 10 000 elements (MP3).

Most solutions did not skip unused elements in the `Stock` array (MP4).

Although solutions often did not correctly extract the correct group from the current element of `StockID` (MP5). Some did achieve a mark (MP6) for using the `LookUp` function correctly by using the value they had extracted and then a further mark (MP7) for storing the value extracted correctly in the `Summary` array.

COMPUTER SCIENCE

Paper 9608/31
Advanced Theory

Key messages

Candidates need to show an in-depth study of the syllabus topics and make good use of the appropriate technical terminology required to answer questions this paper. Candidates, who have studied the theory and have also practised the precise use of these 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 ensure that they provide the information asked for in the question set. It is not enough to rewrite the information given in the question as an answer.

General comments

Candidates need to read examination paper questions carefully before writing an answer. For example, in **Questions 4(a)(i)** and **4(a)(iv)**, the answer must be shown as a sum-of-products. In **Questions 5(b)** and **5(b)(ii)**, Backus-Naur Form (BNF) is required. In **Question 7(d)(ii)**, only the instructions set out in the question can be used in the answer.

Comments on specific questions

Question 1

(a) (i) Most candidates correctly identified the numbers as positive or negative. A common error seen in the justification was to omit the reference to the mantissa.

(ii) Most candidates converted the binary values to denary for the two exponents. Correct denary values for the two mantissas proved more challenging for some candidates. A common error was to treat the mantissa as an integer instead of a fraction.

(iii) Many candidates correctly calculated the denary values of the numbers.

(b) Most candidates correctly identified the normalised number. A common error seen in the justification was to omit the reference to the mantissa.

Question 2

(a) Most candidates correctly identified at least one of the missing layers. A minority of candidates provided acceptable descriptions for the Application and Internet/Network layers. Common errors included incorrectly identifying the lowest layer as the Network layer rather than the Network Interface Layer and incorrectly including the transport of packets in a description of the application layer.

(b) (i) A minority of candidates explained why communication protocols are necessary. A common error was to define a communication protocol. An acceptable answer is ‘All data is transmitted using the same rules and format meaning that the communication is independent of the hardware and software used.’

(ii) Most candidates could identify one communication protocol and state its purpose. Describing the protocol proved more challenging for some candidates.

Question 3

Most candidates included diagrams showing bus and star network topologies. Some diagrams were clearly labelled. A common error in either the diagram or the description of the bus network topology was to incorrectly name the terminators at each end of the cable as terminals.

Question 4

(a) (i) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks. A common error was to incorrectly write $\overline{P.Q.R.S}$ for $\bar{P}.\bar{Q}.\bar{R}\bar{S}$

(ii) Many responses showed a correctly completed Karnaugh Map (K-map).

(iii) Nearly all responses showed correct grouping.

(iv) Most candidates provided a correct simplified sum-of-products for the answer to **part (a)(iii)**.

(b) Some candidates showed good use Boolean algebra to simplify the expression for **X**. Some candidates did not attempt this part of question 4.

Question 5

(a) Most candidates gave acceptable reasons why each statement was invalid.

(b) Nearly all candidates correctly completed the BNF for `<operator>` and `<digit>`. Many candidates correctly completed the BNF for `<variable>` and `<unsigned_integer>`. A minority of candidates correctly completed the BNF for `<assignment_statement>`.

(c) (i) Many candidates provided an acceptable syntax diagram for the revised **variable**.

(ii) A minority of candidates provided acceptable BNF for the revised **variable**. Some candidates did not attempt this part of question 5.

Question 6

(a) (i) Many candidates provided acceptable explanations of the use of asymmetric key cryptography.

(ii) Stating benefits of asymmetric key cryptography proved more challenging for some candidates. An acceptable benefit is 'Increased message security as the decryption key is only known by the intended recipient.'

(b) (i) A minority of candidates provided acceptable explanations the way in which communication security is provided by Transport Layer Security (TLS). Some candidates did not attempt this part of question 6.

(ii) Many of candidates provided acceptable situations for the use of TLS. Some candidates did not attempt this part of question 6.

Question 7

(a) (i) The majority of candidates correctly identified the system as a control system.

(ii) Providing a justification proved more challenging for some candidates. An acceptable answer is 'The system uses feedback to automatically control the heater and air conditioning unit using actuators.'

(b) Many candidates correctly identified the containers where the temperature was out of range and gave the values in denary. A common error seen was to give the value of the temperature in container 4 as 254 degrees.

(c) This question part was generally correctly answered.

(d) (i) Some candidates showed good understanding of the container control system by clearly explaining the purpose of each instruction it applied to the container control system. For example, a description of the first instruction:

`LDD 300` loads the accumulator with the current status of the heaters and air conditioning units held in location 300

A common error seen was to copy the explanation from the table and not apply it to the container control system.

(ii) Most candidates completed the short assembly language program. A common error was to load the data from another address (304).

COMPUTER SCIENCE

Paper 9608/32
Advanced Theory

Key messages

Candidates need to show an in-depth study of the syllabus topics and make good use of the appropriate technical terminology required at this level. Candidates, who have studied the theory and have also practised the precise use of these 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 ensure that they provide the information asked for in the question set. It is not enough to rewrite the information given in the question as an answer.

General comments

Candidates need to read examination paper questions carefully before writing an answer. For example, in **Questions 4(a)(i)** and **4(a)(iv)**, the answer must be shown as a sum-of-products. In **Questions 6(b)** and **6(b)(ii)**, Backus-Naur Form (BNF) is required. In **Question 8(d)(ii)**, only the instructions set out in the question can be used in the answer.

Comments on specific questions

Question 1

(a) (i) Most candidates converted the binary values into denary for the two exponents. Correct denary values for the two mantissas proved more challenging for some candidates. A common error was to treat the mantissa as an integer instead of a fraction.

(ii) Many candidates correctly calculated the denary values of the numbers.

(b) Most candidates correctly identified the normalised number. A common error seen in the justification was to omit the reference to the mantissa.

Question 2

(a) Most candidates correctly identified at least one of the missing layers. A minority of candidates provided acceptable descriptions for the Application and Transport layers. Common errors included incorrectly identifying the lowest layer as the Network layer rather than the Network Interface Layer and incorrectly including the transport of packets in a description of the application layer.

(b) Most candidates could identify at least one communication protocol and were able to state its purpose.

Question 3

(a) Most candidates included a diagram showing the bus network. Some diagrams were clearly labelled. A common error in either the diagram or the description was to incorrectly name the terminators at each end of the cable as terminals.

(b) Many candidates found describing the way in which a bus network uses Ethernet technology for communication challenging. An acceptable answer is ‘All data is transmitted over the same cable

using CSMA/CD. The data is divided into frames containing the MAC addresses of the source and destination and the data itself.'

Question 4

(a)(i) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks. A common error was to incorrectly write $\overline{P.Q.R.S}$ for $\overline{P}.\overline{Q}.\overline{R}.\overline{S}$

(ii) Many responses showed a correctly completed Karnaugh Map.

(iii) Nearly all responses showed correct grouping.

(iv) Most candidates provided a correct simplified sum-of-products for their answer to **part (a)(iii)**.

(b) Some candidates showed good use Boolean algebra to simplify the expression for **X**. Some candidates did not attempt this part of question 4.

Question 5

(a) Many candidates provided acceptable descriptions of the role of flip-flops in a computer.

(b) Many of candidates provided acceptable descriptions of the differences between an SR flip-flop and a JK flip-flop.

Question 6

(a) Most candidates gave correct reasons why each statement was incorrect.

(b) Nearly all candidates correctly completed the BNF for <operator> and <digit>. Many candidates correctly completed the BNF for <variable> and <unsigned_integer>. A minority of candidates correctly completed the BNF for <assignment_statement>.

(c)(i) Many candidates provided an acceptable syntax diagram for the revised **assignment statement**.

(ii) A minority of candidates provided acceptable BNF for the revised **assignment statement**.

Question 7

(a) Most candidates correctly completed the descriptions of a digital certificate and a digital signature. A common error was to provide the incorrect key.

(b) Most candidates were able to identify at least one encryption protocol.

(c) Most candidates were able to identify two methods used to restrict the effect of malware. Providing an adequate description proved more challenging for some candidates. An acceptable method with description is 'installing anti-virus software and keeping it up-to-date. Using this software to regularly scan for viruses and to remove them.'

Question 8

(a)(i) The majority of candidates correctly identified the system as a control system.

(ii) Providing a justification proved more challenging for some candidates. An acceptable answer is 'The system uses feedback to automatically control the heaters and ventilation using actuators.

(b) Many candidates correctly identified the greenhouses where the temperature was out of range and gave the values in denary. A common error seen was to give the value of the temperature in greenhouse 4 as 255 degrees.

(c) This question part was generally correctly answered.

(d) (i) Some candidates showed good understanding of the greenhouse control system by clearly explaining the purpose of each instruction it applied to the greenhouse control system. For example, a description of the first instruction.
`LDD 700` loads the accumulator with the current status of the heaters and vents held in location 700.

A common error seen was to copy the explanation from the table and not apply it to the greenhouse system.

(ii) Most candidates completed the short assembly language program. A common error was to use another address (702) to load the data from.

COMPUTER SCIENCE

Paper 9608/33
Advanced Theory

Key messages

Candidates need to show an in-depth study of the syllabus topics and make good use of the appropriate technical terminology required to answer questions this paper. Candidates, who have studied the theory and have also practised the precise use of these 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 ensure that they provide the information asked for in the question set. It is not enough to rewrite the information given in the question as an answer.

General comments

Candidates need to read examination paper questions carefully before writing an answer. For example, in **Questions 4(a)(i)** and **4(a)(iv)**, the answer must be shown as a sum-of-products. In **Questions 5(b)** and **5(b)(ii)**, Backus-Naur Form (BNF) is required. In **Question 7(d)(ii)**, only the instructions set out in the question can be used in the answer.

Comments on specific questions

Question 1

(a) (i) Most candidates correctly identified the numbers as positive or negative. A common error seen in the justification was to omit the reference to the mantissa.

(ii) Most candidates converted the binary values to denary for the two exponents. Correct denary values for the two mantissas proved more challenging for some candidates. A common error was to treat the mantissa as an integer instead of a fraction.

(iii) Many candidates correctly calculated the denary values of the numbers.

(b) Most candidates correctly identified the normalised number. A common error seen in the justification was to omit the reference to the mantissa.

Question 2

(a) Most candidates correctly identified at least one of the missing layers. A minority of candidates provided acceptable descriptions for the Application and Internet/Network layers. Common errors included incorrectly identifying the lowest layer as the Network layer rather than the Network Interface Layer and incorrectly including the transport of packets in a description of the application layer.

(b) (i) A minority of candidates explained why communication protocols are necessary. A common error was to define a communication protocol. An acceptable answer is ‘All data is transmitted using the same rules and format meaning that the communication is independent of the hardware and software used.’

(ii) Most candidates could identify one communication protocol and state its purpose. Describing the protocol proved more challenging for some candidates.

Question 3

Most candidates included diagrams showing bus and star network topologies. Some diagrams were clearly labelled. A common error in either the diagram or the description of the bus network topology was to incorrectly name the terminators at each end of the cable as terminals.

Question 4

(a) (i) Those candidates who correctly wrote their answer as a sum-of-products usually gained full marks. A common error was to incorrectly write $\overline{P.Q.R.S}$ for $\bar{P}.\bar{Q}.\bar{R}\bar{S}$

(ii) Many responses showed a correctly completed Karnaugh Map (K-map).

(iii) Nearly all responses showed correct grouping.

(iv) Most candidates provided a correct simplified sum-of-products for the answer to **part (a)(iii)**.

(b) Some candidates showed good use Boolean algebra to simplify the expression for **X**. Some candidates did not attempt this part of question 4.

Question 5

(a) Most candidates gave acceptable reasons why each statement was invalid.

(b) Nearly all candidates correctly completed the BNF for `<operator>` and `<digit>`. Many candidates correctly completed the BNF for `<variable>` and `<unsigned_integer>`. A minority of candidates correctly completed the BNF for `<assignment_statement>`.

(c) (i) Many candidates provided an acceptable syntax diagram for the revised **variable**.

(ii) A minority of candidates provided acceptable BNF for the revised **variable**. Some candidates did not attempt this part of question 5.

Question 6

(a) (i) Many candidates provided acceptable explanations of the use of asymmetric key cryptography.

(ii) Stating benefits of asymmetric key cryptography proved more challenging for some candidates. An acceptable benefit is 'Increased message security as the decryption key is only known by the intended recipient.'

(b) (i) A minority of candidates provided acceptable explanations the way in which communication security is provided by Transport Layer Security (TLS). Some candidates did not attempt this part of question 6.

(ii) Many of candidates provided acceptable situations for the use of TLS. Some candidates did not attempt this part of question 6.

Question 7

(a) (i) The majority of candidates correctly identified the system as a control system.

(ii) Providing a justification proved more challenging for some candidates. An acceptable answer is 'The system uses feedback to automatically control the heater and air conditioning unit using actuators.'

(b) Many candidates correctly identified the containers where the temperature was out of range and gave the values in denary. A common error seen was to give the value of the temperature in container 4 as 254 degrees.

(c) This question part was generally correctly answered.

(d) (i) Some candidates showed good understanding of the container control system by clearly explaining the purpose of each instruction it applied to the container control system. For example, a description of the first instruction:

`LDD 300` loads the accumulator with the current status of the heaters and air conditioning units held in location 300

A common error seen was to copy the explanation from the table and not apply it to the container control system.

(ii) Most candidates completed the short assembly language program. A common error was to load the data from another address (304).

COMPUTER SCIENCE

Paper 9608/41

Further Problem-Solving and Programming Skills

Key messages

Candidates needed to demonstrate their understanding of programming structures, constructs, and their algorithmic thinking for this paper. They needed to read and write pseudocode, program code and assembly language code.

General comments

Many candidates were able to demonstrate their understanding of writing object-oriented code in their chosen programming language and wrote class definitions and constructors accurately.

Some candidates demonstrated an understanding of using hash tables and random access files, but candidates need to make sure that they are opening files in the correct format, and that their code that closes files run in all scenarios.

Comments on specific questions

Question 1

(a) There were mixed responses to this question. Some candidates found this question part challenging. Many candidates could correctly identify several of the requirements, such as the cancel and entering the code. Common errors included a lack of precision in answers, for example in place of 'Dispense item', some responses included 'User purchases an item'; the whole system is about a user purchasing an item; this part of the diagram needed to identify the state of the machine i.e. what the machine was doing at this specific point.

(b) (i) This question required candidates to demonstrate their understanding of object-oriented programming in their chosen high-level language. Many candidates were able to declare the class. A common error included attempting to use inheritance in the class definition, which was not part of the class.

Other common errors included not identifying the private element of these attributes, or not identifying the data types to be used. If candidates use Python, they are still required to identify the data types for these attributes to demonstrate this understanding and must use comments to do this. Some candidates did make use of these comments, stating the data type alongside each attribute's first use. When declaring an array, candidates need to identify the number of elements that it will include and the data type, in this case of type class `foodItem`.

Many candidates were able to use the correct constructor for their chosen language. Fewer passed the required parameters into this constructor. The class diagram describes the requirements of each class and method, and candidates need to make sure they refer to this diagram in each question to make sure they are meeting all the requirements.

(ii) Some candidates found this question challenging. The question required candidates to use the methods to access the attributes of objects within an array, and a common error was using the object in a comparison, instead of access the code of the object.

Many candidates were able to search the array using a linear search and were able to return the appropriate values in sensible places.

Few candidates identified the need to compare the cost with the class attribute `moneyIn`, and some responses attempted to read this value in as a parameter, or input from the user.

(iii) Many candidates were familiar with declaring instances of objects and could declare an object of the appropriate type. Fewer candidates passed the appropriate parameters to the constructor. Common errors included missing out the parameters, passing the strings i.e. 'chocolate' instead of the object `chocolate`, or attempting to add each item manually to the newly instantiated object.

Question 2

(a) Candidates needed to use pseudocode to declare a record data type for `customer`. Many candidates were able to use appropriate code to declare a record, or structure, for `customer`. Candidates also needed to declare the data items within this structure, including the data types as required by a record data structure. Some candidates omitted these data types or used inappropriate data types for example the phone number example include a + symbol and therefore could not be stored as an integer.

(b) (i) Many candidates were able to correctly calculate the hash value for each customer ID. Some candidates attempted different calculations than that given in the question, for example not using modulus division instead using normal division and therefore giving incorrect values.

(ii) Many Candidates were able to demonstrate a good clear understanding of hashing and collisions. Most responses included moving to the next space to check if it was empty, and then continuing this, returning to the start if the last record is found.

Some candidates also described the use of overflow tables and using linked lists (often referred to as arrays) as each record location to store the collisions.

(iii) Some candidates were able to make appropriate use of pseudocode to access the random files to open and access the required value. When declaring a function, candidates need to make sure they are sending the correct parameter data type if they give it in their response, and the correct return value data type. The question stated that the function takes a customer ID as a parameter and not an object, therefore the parameter should have been an integer value for this ID as opposed to accessing the customer ID of a record.

Candidates often made suitable use of the `getRecordLocation()` function and stored the return value. When candidates open a file, they need to consider the type of file that they are opening and make sure this is clear in their pseudocode response. For example, some candidates opened the file without any reference to the random access format.

Many candidates were able to make use of suitable code to access the data in the file, but at times, they were not making use of the calculated address, for example, they read the file and store the value, but did not read the value at the hash address.

Many candidates did close the file, which should always be included in file access, however when a function returns a value the code beyond this will not run, and many candidates returned values and then inserted a close file command at the end of the function, which will never execute and therefore the file will not close.

Question 3

(a) This question was answered well by many candidates who were able to insert the correct tasks and their duration in the correct positions.

(b) This question required candidates to apply their understanding to the tasks in the table instead of giving a generic answer of dividing tasks. Some candidates were able to do this and gave clear identification of tasks that could be divided.

Candidates need to make sure they are reading the question carefully, for example, this question asked how the tasks can be divided between teams, but some candidates described how tasks can be run concurrently without reference to the division between teams.

(c) Candidates found this question challenging. Many attempted to describe the use of PERT charts and the identification of a critical path, which is not relevant to the use of program libraries.

The most common correct answers included a description of saving time, but this was often repeated and not expanded to explain why time can be saved.

(d) Candidates need to be able to differentiate between the different tools on an IDE, for example, the difference between the features of an editor and a debugger. This question required identification of editor features. The most common responses included the use of colour coding and autocomplete.

Question 4

(a) This question was answered well by many candidates who correctly positioned the nodes on the tree. When adding nodes, candidates need to make sure they clearly show if the node is to the left or the right of its parent node.

(b) (i) Many candidates also answered this question well. They accurately identified the pointers for the given diagram. The question stated that null pointers are stored as –1, but this was a common error where candidates attempted to use other values instead.

(ii) Some candidates found this question challenging. It required an array declaration to identify the number of elements and data type. Common errors included attempting to initialise array values instead of declaring the array, or not including the data type. Some candidates declared an array with elements 0 to 100, which would give 101 elements and not the required 100.

(iii) This question required an understanding of how to access and follow the pointers of a linked list stored in an array, as well as write a recursive function. Some candidates were able to accurately produce this algorithm, accessing the required pointers and calling the function with this pointer. Fewer candidates were able to use the required selection to make sure that the node was not empty before calling the recursive function. A common error included the incorrect order of the steps, for example both recursive calls before outputting the data, or outputting the data multiple times.

Another common error was using one ‘if .. else’ statement for the two checks, instead of two separate ‘if’ statements. This would mean that when the recursive call rewound, it would not then check the right link.

Question 5

(a) Responses were mixed to this question. Some candidates were able to apply their understanding of a binary search and update the values appropriately. Common errors included replacing lower and upper with mid, instead of the mid with the addition or subtraction of 1, as well as putting the lower and upper changing values in the incorrect location.

(b) This question required candidates to rewrite the algorithm as a recursive function. Some candidates did not include the required recursive calls, instead rewriting the algorithm in the same format as the previous question. Candidates needed to include the remainder of the algorithm so that it was functional. Some candidates did not include the first comparison to check if the data was not found and returned –1.

Question 6

Some candidates were able to demonstrate a good understanding of assembly language programs and completed the missing code accurately. Candidates were also required to use the masks appropriately. A common mistake was using an incorrect opcode or accessing the incorrect operand. Candidates need to make sure they are using the appropriate Op codes from the given list, including the correct access mode.

Question 7

(a) This question was often answered well, with candidates looping through the values accurately. Some candidates attempted to access an object as a parameter, and updated a pointer for each array element, which was not given in the question. There were two parameters: one storing the integer values and the top of stack pointer storing a singular value.

(b) This question required an understanding of how data items are removed from a stack. Some candidates did not use the top of stack pointer to check if it was empty, but often the appropriate value was then returned. Fewer candidates accurately returned the top data item and decremented the pointer, often returning the value and either missing the decrement, or doing this after the return statement in the function which meant that this code would never be executed.

COMPUTER SCIENCE

Paper 9608/42

Further Problem-Solving and Programming Skills

Key messages

This paper required candidates to demonstrate their understanding of programming structures, constructs, and their algorithmic thinking. Candidates need to be able to read and write pseudocode, program code and assembly language code.

General comments

Many candidates were able to demonstrate their understanding of writing object-oriented code in their chosen programming language and were able to write class definitions and constructors accurately.

Some candidates demonstrated an understanding of using hash tables and random access files, but candidates need to make sure that they are opening file in the correct format, and that their code to close the file runs in all scenarios.

Where a question includes a context, candidates need to make sure they are referencing this appropriately in their responses, for example, there is a difference between explaining how to generically search a binary tree and searching for a specific value in a given binary tree.

Comments on specific questions

Question 1

(a) This question was answered well by many candidates who were able to correctly identify the root node of the given tree.

(b) This question was also answered well by many candidates. A common error was identifying Donkey or Kangaroo as the leaf node, although these are children from the root node; the leaf node does not have any children.

(c) Responses to this question part were mixed. Many candidates were able to correctly insert the nodes. Some candidates did not follow the alphabetical order of the values in the tree and positioned the nodes in different places. Some additions by the candidates showed the nodes below the appropriate node but positioned directly underneath and therefore not identifying if the node was inserted to the left or the right. The question states that the tree is a binary tree and therefore candidates need to demonstrate their understanding that each node has two nodes beneath, one to the left and one to the right.

(d) This question required candidates to explain how the given tree will be searched to find Elephant. This required an application of knowledge to this scenario. Some candidate gave a generic description of searching a binary tree, or wrote an algorithm, which did not explain how Elephant would be searched for in this tree.

Question 2

(a) Candidates needed to write a pseudocode response to this question. This needed to be logical and follow a suitable structure to meet the requirements. This included the declaration of a record, or structure etc. A common error was writing a subroutine or just using the identifier `Booking` without any reference to declaring it as a record type.

The question stated that all four values were to be integers and candidates needed to use this in their response to declare each as integer values.

(b) (i) There were mixed responses to this question part. The function needed to take the booking ID as a parameter. In the previous question part, candidates were told that this was an integer. Some candidates passed a `Booking` record to the function, which did not meet the requirements. Similarly, the function needed to return the final value, there are a variety of ways this could be completed depending on the language chosen, but many candidates did not include the return of this value, with some outputting it instead.

Candidates needed to identify that modular division was required to perform the calculation. Some candidates were able to identify the modulus division syntax for their chosen language, but often this was given as a standard division, or an attempt to cast as an integer for integer division which did not give the correct result.

(ii) Many candidates were able to correctly calculate the hash value for each booking ID. Some candidates gave incorrect values by attempting different calculations than that given in the question.

(c) Some candidates were able to make appropriate use of pseudocode to access the random files to open and access the required value. When declaring a function candidates need to make sure they are sending the correct parameter data type if they give it in their response, and the correct return value data type. The question stated that the function takes a record as a parameter; a common error was sending a booking ID instead.

The hash function was often called; an inaccurate value was often given as a parameter. Some candidates attempted to rewrite the hash function instead of calling it.

When candidates open a file, they need to consider the type of file that they are opening and make sure this is clear in their pseudocode response. For example, some candidates opened the file without any reference to the random access format.

Many candidates were able to make use of suitable code to access the data in the file, but at times they were not making use of the calculated address, for example, they access the file and store the return value to check if it was free but did not identify the movement to the address to access the record.

Many candidates did close the file (which should always be included in file access), however, when a function returns a value the code beyond this will not run, and many candidates returned values and then inserted a close file command at the end of the function. This will never execute and therefore the file will not close.

(d) Candidates were able to demonstrate an understanding of what exception handling is and why it is used. Fewer candidates used this understanding to explain how it can be used when reading from a file. This question required application of their understanding to file reading, and therefore needed to reference a reason why file handling is used in this situation such as in case the file does not exist.

Question 3

(a) This question required candidates to demonstrate their understanding of classes. Many candidates were able to correctly declare the class and could use their constructor appropriately. Fewer candidates declared the attributes appropriately. The question stated that all attributes are to be declared as private. Common errors included not identifying the private element of these attributes, or not identifying the data types to be used; if candidates are using Python, they are still required to

identify the data types for these attributes to demonstrate this understanding and therefore are instructed to use comments to do this. Some candidates did make use of these comments, stating the data type alongside each attribute's first use. When declaring an array, candidates need to identify the number of elements that it will include and the data type. Some candidates did not include the number of elements, whilst others using a different data type such as string in place of `QuestionClass`.

Some candidates attempted to declare a constructor method in a language where the constructor has a designated identifier and therefore this will not be the inbuilt class constructor.

(b) This question stated that the method takes a question object as a parameter, this required candidates to identify that the data type is the class type and not, for example, a string value.

Candidates need to make sure they are revisiting the class descriptions when they answer each part of a question. The class has an attribute that stores the number of questions, which allows identification of a full array by there being 20 questions. Some candidates attempted to iterate through the array to check if there was an object in each position. On many occasions, they checked if the array item was '' when the item is an object and therefore does not have a value for comparison in this way.

Many candidates were able to insert the object into the array, the method differing depending on how they checked if the array was empty, and whilst many did return true, they did not consider the need to increment the number of questions parameter to make sure this was appropriately updated.

(c) Many candidates were able to declare instances of the given objects. Fewer used the appropriate parameters within the constructor to create the new question object required. Where candidates have been given identifiers to use, they need to make sure that they are using these accurately.

Fewer candidates were able to use the appropriate method to assign the question to the first quiz, some candidates attempted to assign the question, then answer etc. separately instead of assigning the object they had created.

A common error included mixing the identifiers with the class names, for example creating an instance of `FirstQuiz` with the identifier `QuizClass` instead of vice-versa.

(d) This question was answered well by many candidates who were able to correctly identify the use of containment. The most common incorrect answers given was inheritance.

(e) (i) This question required candidates to identify **when** the translators should be used, and **not why**. Some candidates explained the benefits of compilers and interpreters or gave features of them instead of when they would use them during the development process.

Candidates need to make sure they are giving more information in their answer than that provided in the question. In this example, they are asked when during the development process the translators are used, therefore stating they are used in the development is repeating the question and not providing further information.

(ii) Candidates need to be able to differentiate between debugging and identifying errors. This question required facilities for debugging, i.e., working out what the error is and correcting it. The most common answers included breakpoints and stepping.

(iii) An editor is a feature of an IDE, and this question wanted candidates to consider features within this editor. This includes features to support the writing of the programs, and not the debugging tools asked for previously.

Question 4

(a) Candidates found the use of a circular queue challenging. Few candidates were able to accurately enqueue and dequeue the data, manipulating the head and tail index as they went. Some candidates attempted to move the array elements down to start at index 0 instead of enqueue at the end and dequeuing at the front.

(b) (i) Candidates found elements of this question challenging. A common error was identifying if number of elements in queue was greater than 10, which would mean that if there were already 10 elements in the queue then another element could be added which would error as there are not enough spaces in the array.

Where candidates are completing gaps in programs, they need to make sure they are following the spelling and case used in the question; this demonstrates their understanding that `numberinqueue` is different to `NumberInQueue`.

(ii) Most candidates made a reasonable attempt at checking if the queue was empty. A common error was to check each element in the array as opposed to making use of the head and tail index. Another common error was checking if the tail index was less than the head index, which would not work in this instance where a circular queue is being incremented.

Many candidates were able to increment the head index, and some correctly updated this as a circular queue by resetting the value to the first queue elements.

Where candidates need to return a value in a function, they need to understand that the code after this return statement will not run. Some candidates returned the value at the head of the queue and then attempted to update the pointers.

Some candidates attempted to change the head index before accessing the value to be returned, for example, they incremented the head index and then accessed the value at this new location, which is now pointing to the next queue element instead of the one to be accessed.

Question 5

Candidates found this question challenging. Most candidates could identify the comparison to `Temp`, but the comparison to `Counter` was often mixed with the variable `Count` instead.

Question 6

(a) Most candidates were familiar with completing decision tables and were able to identify at least several of the actions correctly. Candidates need to be clear when giving their responses, for example using Y, or X consistently.

(b) Candidates found this question more challenging. Candidates needed to identify the conditions that were did not impact the resulting actions. This needed to be identified in the conditions using an appropriate symbol to identify that either Y or N could be used. Many candidates did not identify these features, instead giving one set of conditions that gave an action without identifying those conditions that do impact the results.

Question 7

This question was answered well by many candidates. They were able to identify the missing statements and presented these appropriately. A common error was not identifying that the animal strength and health needed to be > = 10 and not just > 10.

Question 8

Candidate responses were mixed to this question. Most candidates were able to identify some correct op code and operands. Fewer candidates were able to identify the correct location for the label `LOOP`. Candidates need to carefully check the Op codes that they are using; specifically, the different addressing modes that need to be used to access the data.

A common error included the mixing up of left shift and right shift, candidates needed to identify that left shift multiplies the binary values, whilst a right shift will divide.

COMPUTER SCIENCE

Paper 9608/43

Further Problem-Solving and Programming Skills

Key messages

Candidates needed to demonstrate their understanding of programming structures, constructs, and their algorithmic thinking for this paper. They needed to read and write pseudocode, program code and assembly language code.

General comments

Many candidates were able to demonstrate their understanding of writing object-oriented code in their chosen programming language and wrote class definitions and constructors accurately.

Some candidates demonstrated an understanding of using hash tables and random access files, but candidates need to make sure that they are opening files in the correct format, and that their code that closes files run in all scenarios.

Comments on specific questions

Question 1

(a) There were mixed responses to this question. Some candidates found this question part challenging. Many candidates could correctly identify several of the requirements, such as the cancel and entering the code. Common errors included a lack of precision in answers, for example in place of 'Dispense item', some responses included 'User purchases an item'; the whole system is about a user purchasing an item; this part of the diagram needed to identify the state of the machine i.e. what the machine was doing at this specific point.

(b) (i) This question required candidates to demonstrate their understanding of object-oriented programming in their chosen high-level language. Many candidates were able to declare the class. A common error included attempting to use inheritance in the class definition, which was not part of the class.

Other common errors included not identifying the private element of these attributes, or not identifying the data types to be used. If candidates use Python, they are still required to identify the data types for these attributes to demonstrate this understanding and must use comments to do this. Some candidates did make use of these comments, stating the data type alongside each attribute's first use. When declaring an array, candidates need to identify the number of elements that it will include and the data type, in this case of type class `foodItem`.

Many candidates were able to use the correct constructor for their chosen language. Fewer passed the required parameters into this constructor. The class diagram describes the requirements of each class and method, and candidates need to make sure they refer to this diagram in each question to make sure they are meeting all the requirements.

(ii) Some candidates found this question challenging. The question required candidates to use the methods to access the attributes of objects within an array, and a common error was using the object in a comparison, instead of access the code of the object.

Many candidates were able to search the array using a linear search and were able to return the appropriate values in sensible places.

Few candidates identified the need to compare the cost with the class attribute `moneyIn`, and some responses attempted to read this value in as a parameter, or input from the user.

(iii) Many candidates were familiar with declaring instances of objects and could declare an object of the appropriate type. Fewer candidates passed the appropriate parameters to the constructor. Common errors included missing out the parameters, passing the strings i.e. 'chocolate' instead of the object `chocolate`, or attempting to add each item manually to the newly instantiated object.

Question 2

(a) Candidates needed to use pseudocode to declare a record data type for `customer`. Many candidates were able to use appropriate code to declare a record, or structure, for `customer`. Candidates also needed to declare the data items within this structure, including the data types as required by a record data structure. Some candidates omitted these data types or used inappropriate data types for example the phone number example include a + symbol and therefore could not be stored as an integer.

(b) (i) Many candidates were able to correctly calculate the hash value for each customer ID. Some candidates attempted different calculations than that given in the question, for example not using modulus division instead using normal division and therefore giving incorrect values.

(ii) Many Candidates were able to demonstrate a good clear understanding of hashing and collisions. Most responses included moving to the next space to check if it was empty, and then continuing this, returning to the start if the last record is found.

Some candidates also described the use of overflow tables and using linked lists (often referred to as arrays) as each record location to store the collisions.

(iii) Some candidates were able to make appropriate use of pseudocode to access the random files to open and access the required value. When declaring a function, candidates need to make sure they are sending the correct parameter data type if they give it in their response, and the correct return value data type. The question stated that the function takes a customer ID as a parameter and not an object, therefore the parameter should have been an integer value for this ID as opposed to accessing the customer ID of a record.

Candidates often made suitable use of the `getRecordLocation()` function and stored the return value. When candidates open a file, they need to consider the type of file that they are opening and make sure this is clear in their pseudocode response. For example, some candidates opened the file without any reference to the random access format.

Many candidates were able to make use of suitable code to access the data in the file, but at times, they were not making use of the calculated address, for example, they read the file and store the value, but did not read the value at the hash address.

Many candidates did close the file, which should always be included in file access, however when a function returns a value the code beyond this will not run, and many candidates returned values and then inserted a close file command at the end of the function, which will never execute and therefore the file will not close.

Question 3

(a) This question was answered well by many candidates who were able to insert the correct tasks and their duration in the correct positions.

(b) This question required candidates to apply their understanding to the tasks in the table instead of giving a generic answer of dividing tasks. Some candidates were able to do this and gave clear identification of tasks that could be divided.

Candidates need to make sure they are reading the question carefully, for example, this question asked how the tasks can be divided between teams, but some candidates described how tasks can be run concurrently without reference to the division between teams.

(c) Candidates found this question challenging. Many attempted to describe the use of PERT charts and the identification of a critical path, which is not relevant to the use of program libraries.

The most common correct answers included a description of saving time, but this was often repeated and not expanded to explain why time can be saved.

(d) Candidates need to be able to differentiate between the different tools on an IDE, for example, the difference between the features of an editor and a debugger. This question required identification of editor features. The most common responses included the use of colour coding and autocomplete.

Question 4

(a) This question was answered well by many candidates who correctly positioned the nodes on the tree. When adding nodes, candidates need to make sure they clearly show if the node is to the left or the right of its parent node.

(b) (i) Many candidates also answered this question well. They accurately identified the pointers for the given diagram. The question stated that null pointers are stored as –1, but this was a common error where candidates attempted to use other values instead.

(ii) Some candidates found this question challenging. It required an array declaration to identify the number of elements and data type. Common errors included attempting to initialise array values instead of declaring the array, or not including the data type. Some candidates declared an array with elements 0 to 100, which would give 101 elements and not the required 100.

(iii) This question required an understanding of how to access and follow the pointers of a linked list stored in an array, as well as write a recursive function. Some candidates were able to accurately produce this algorithm, accessing the required pointers and calling the function with this pointer. Fewer candidates were able to use the required selection to make sure that the node was not empty before calling the recursive function. A common error included the incorrect order of the steps, for example both recursive calls before outputting the data, or outputting the data multiple times.

Another common error was using one ‘if .. else’ statement for the two checks, instead of two separate ‘if’ statements. This would mean that when the recursive call rewound, it would not then check the right link.

Question 5

(a) Responses were mixed to this question. Some candidates were able to apply their understanding of a binary search and update the values appropriately. Common errors included replacing lower and upper with mid, instead of the mid with the addition or subtraction of 1, as well as putting the lower and upper changing values in the incorrect location.

(b) This question required candidates to rewrite the algorithm as a recursive function. Some candidates did not include the required recursive calls, instead rewriting the algorithm in the same format as the previous question. Candidates needed to include the remainder of the algorithm so that it was functional. Some candidates did not include the first comparison to check if the data was not found and returned –1.

Question 6

Some candidates were able to demonstrate a good understanding of assembly language programs and completed the missing code accurately. Candidates were also required to use the masks appropriately. A common mistake was using an incorrect opcode or accessing the incorrect operand. Candidates need to make sure they are using the appropriate Op codes from the given list, including the correct access mode.

Question 7

(a) This question was often answered well, with candidates looping through the values accurately. Some candidates attempted to access an object as a parameter, and updated a pointer for each array element, which was not given in the question. There were two parameters: one storing the integer values and the top of stack pointer storing a singular value.

(b) This question required an understanding of how data items are removed from a stack. Some candidates did not use the top of stack pointer to check if it was empty, but often the appropriate value was then returned. Fewer candidates accurately returned the top data item and decremented the pointer, often returning the value and either missing the decrement, or doing this after the return statement in the function which meant that this code would never be executed.