A-LEVEL
COMPUTING

Unit 3: Problem Solving, Programming, Operating Systems, Databases and Networking
Mark scheme

2510/COMP3
June 2014

Version/Stage: V1.0 Final

Mark schemes are prepared by the Lead Assessment Writer and considered, together with the relevant questions, by a panel of subject teachers. This mark scheme includes any amendments made at the standardisation events which all associates participate in and is the scheme which was used by them in this examination. The standardisation process ensures that the mark scheme covers the students' responses to questions and that every associate understands and applies it in the same correct way. As preparation for standardisation each associate analyses a number of students' scripts: alternative answers not already covered by the mark scheme are discussed and legislated for. If, after the standardisation process, associates encounter unusual answers which have not been raised they are required to refer these to the Lead Assessment Writer.

It must be stressed that a mark scheme is a working document, in many cases further developed and expanded on the basis of students' reactions to a particular paper. Assumptions about future mark schemes on the basis of one year's document should be avoided; whilst the guiding principles of assessment remain constant, details will change, depending on the content of a particular examination paper.

Further copies of this Mark Scheme are available from aqa.org.uk

The following annotation is used in the mark scheme.

; - means a single mark
// - means alternative response
/ - means an alternative word or sub-phrase
A - means acceptable creditworthy answer
R - means reject answer as not creditworthy
NE - means not enough
I - means ignore
DPT - in some questions a specific error made by a candidate, if repeated, could result in the loss of more than one mark. The **DPT** label indicates that this mistake should only result in a candidate losing one mark, on the first occasion that the error is made. Provided that the answer remains understandable, subsequent marks should be awarded as if the error was not being repeated.

<table>
<tr><th>Qu.</th><th>Part</th><th></th><th>Marking guidance</th><th>Mark</th></tr>
<tr><td>1</td><td>a</td><td></td><td>
<table>
<tr><th>Situation</th><th>Most likely to be Parallel</th><th>Most likely to be Serial</th><th>Could be either Serial or Parallel</th></tr>
<tr><td>Sending data to a peripheral, such as a printer, that is plugged directly into a desktop computer.</td><td></td><td>A. ✓</td><td>✓</td></tr>
<tr><td>Transferring memory addresses between the processor and the main memory of a desktop computer.</td><td>✓</td><td></td><td></td></tr>
<tr><td>Transmitting an e-mail across a WAN from a computer in England to an e-mail server in Scotland.</td><td></td><td>✓</td><td></td></tr>
</table>
One mark per row with a correct tick
Do not award marks for any row which has more than one tick
A. alternative indicators for ticks e.g. crosses, Y, Yes
</td><td>3</td></tr>
<tr><td>1</td><td>b</td><td></td><td>To check that a (receiving) device is connected;
To check that a (receiving) device is ready to receive data // to inform a (transmitting) device that a (receiving) device is/is not ready to receive data;
To tell a (receiving) device that data is ready to be transmitted;
To negotiate/agree how the transmission will take place // to agree the system to be used for transmission ; A. an example of a setting that might be agreed during a handshake eg bit rate, parity
To ensure successful communication;
MAX 1</td><td>1</td></tr>
<tr><td>1</td><td>c</td><td></td><td>Time delay between the moment something is initiated and the moment its effect begins;
A. time delay between signal being transmitted and arriving
A. time taken for transmitted data to arrive at the receiver
A. lag for time delay
NE. delay in transmission, transmission time</td><td>1</td></tr>
<tr><td>2</td><td>a</td><td>i</td><td>Zero or more bs followed by a/one c;
A. answers by example but must be at least c, bc, bbc and indicate the sequence continues.</td><td>1</td></tr>
<tr><td>2</td><td>a</td><td>ii</td><td>Zero or one bs followed by (a/one) c // the strings c or bc;</td><td>1</td></tr>
</table>

2	**b**		Correct expression: `b(cd)*(e|fg)` **A.** use of incorrect bracket types **A.** accept brackets around `fg` **A.** `(cd)+?` for `(cd)*` **I.** ^ at start, $ at end of expression **Two marks** for the full correct expression. **One mark** for including either `(cd)*` or `(e|fg)` in an incorrect expression.	2

3	**a**		<table><tr><th>Algorithm Name</th><th>Requires Sorted List? (Tick one box)</th></tr><tr><td>Binary search</td><td>✓</td></tr><tr><td>Linear search</td><td></td></tr></table> **One mark** for having a tick in the "Binary search" row. **A.** alternative indicators for tick eg "Yes" **A.** a tick for "Binary search" and a cross for "Linear search" **R.** answers where two ticks have been used.	1

3	**b**		<table><tr><th rowspan="2">List Length</th><th rowspan="2">Outer Pointer</th><th rowspan="2">Current Value</th><th rowspan="2">Inner Pointer</th><th colspan="4">List</th></tr><tr><th>[1]</th><th>[2]</th><th>[3]</th><th>[4]</th></tr><tr><td></td><td></td><td></td><td></td><td>9</td><td>8</td><td>5</td><td>6</td></tr><tr><td>4</td><td>2</td><td>8</td><td>1</td><td></td><td>9</td><td></td><td></td></tr><tr><td rowspan="2"></td><td rowspan="2"></td><td></td><td>0</td><td>8</td><td></td><td></td><td></td></tr><tr><td>3</td><td>5</td><td>2</td><td></td><td></td><td>9</td><td></td></tr><tr><td></td><td></td><td></td><td>1</td><td></td><td>8</td><td></td><td></td></tr><tr><td rowspan="2"></td><td rowspan="2"></td><td></td><td>0</td><td>5</td><td></td><td></td><td></td></tr><tr><td>4</td><td>6</td><td>3</td><td></td><td></td><td></td><td>9</td></tr><tr><td></td><td></td><td></td><td>2</td><td></td><td></td><td>8</td><td></td></tr><tr><td></td><td></td><td></td><td>1</td><td></td><td>6</td><td></td><td></td></tr></table> Award **one mark** for each of the highlighted rectangles which has the correct values written in it in the unshaded cells. Accept responses in which correct values are unnecessarily written out again. Do not award a mark for any rectangle which has an incorrect value written in it.	3

3	**c**		The value being moved / CurrentValue / 6 does not need to be put at the start of the list // should be inserted at position 2 not position 1;	1

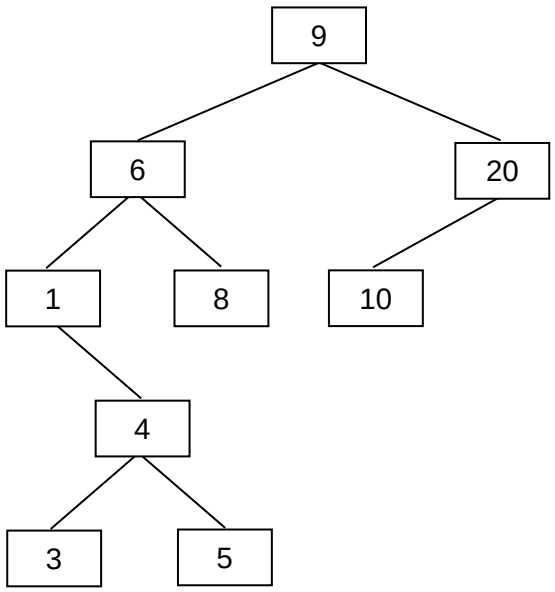
			Because the second condition (in the While statement) is not satisfied; **MAX 1**	

3	**d**		<table><tr><th>Order of Time Complexity</th><th>Tick one box</th></tr><tr><td>O(n)</td><td></td></tr><tr><td>$O(n^2)$</td><td>✓</td></tr><tr><td>$O(2^n)$</td><td></td></tr></table> **A.** alternative indicators instead of a tick eg a cross, Y, Yes **R.** responses in which more than one box is ticked	1

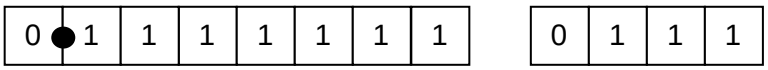
3	**e**		Insertion sort; **A.** Insert sort	1

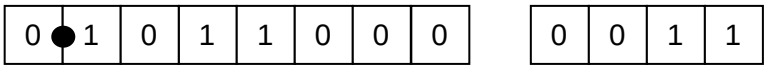
3	**f**	**i**	9, 6, 8; Must be in the order above. Can be separated by any character or a space.	1

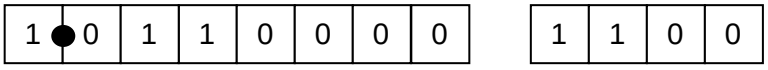
3	**f**	**ii**	9, 20,10; Must be in the order above. Can be separated by any character or a space.	1

3	**g**		 **1 mark** for inserting number 4 in the correct place **1 mark** for inserting both numbers 3 and 5 in the correct place	2

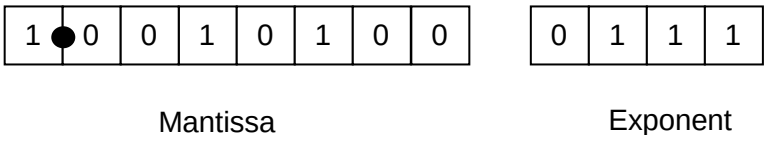
			relative to 4 **MAX 1** if any numbers added in the wrong place / any extra numbers added	

4	**a**		 Mantissa Exponent **1 mark** for correct mantissa **1 mark** for correct exponent	2

4	**b**		 Mantissa Exponent **1 method mark** for either: • showing correct value of both mantissa and exponent in denary (Mantissa = 0.6875 // 11/16, Exponent = 3) • showing binary point shifted 3 places to right in binary number • indicating that final answer calculated using answer = mantissa x $2^{exponent}$ **1 mark** for correct answer Answer = 5 ½ // 5.5 *If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.*	2

4	**c**		 Mantissa Exponent **1 method mark** for either: • showing correct value of both mantissa and exponent in denary (Mantissa = -0.625 // -5/8, Exponent = -4) • showing binary point shifted 4 places to left in binary number • indicating that final answer calculated using answer = mantissa x $2^{exponent}$ **1 mark** for correct answer Answer = -5/128,-0.0390625 **A.** rounded to at least 2dp *If answer is correct and some working has been shown, award*	2

			two marks, even if working would not have gained credit on its own.	

4	**d**		**2 marks** for working: Correct representation of 108 in binary: 1101100; **A.** any number of preceding 0s or succeeding 0s after a binary point Correct representation of -108: 10010100; **A.** any number of preceding 1s Showing the correct value of the exponent in denary (7) or binary (0111) // showing the binary point being shifted 7 places; Showing the correct value of the mantissa in floating point binary: 1.0010100; **MAX 2** **1 mark** for correct mantissa and exponent together:  *If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.* *Marks for working can be awarded in the answer.*	3

4	**e**	**i**	The results of a calculation; is a number that is too large to store; in the available storage space/number of bits; Must get the middle point (about the number being too large) to be awarded any marks. **MAX 2**	2

4	**e**	**ii**	<table><tr><th>Operation</th><th>May cause overflow? (Tick One)</th></tr><tr><td>Subtracting a very small number from a large number.</td><td></td></tr><tr><td>Dividing a large number by a very small number.</td><td>✓</td></tr><tr><td>Multiplying a large number by a very small number.</td><td></td></tr></table> **A.** alternative symbol which clearly indicates just one box eg cross, Y, Yes **R.** answers in which more than one row is ticked	1

5	**a**		Values/cards need to be taken out of the data structure from the opposite end that they are put in // cards removed from top/front and added at end/bottom/rear; Values/cards need to be removed in the same order that they are added; **A.** It is First in First Out // It is FIFO; **A.** It is Last in Last Out // It is LILO; **MAX 1**	1

5	**b**	**i**	`FrontPointer = 11`
			`RearPointer = 52`
			`QueueSize = 42`
			1 mark for all three values correct
				1

5	**b**	**ii**	`FrontPointer = 11`
			`RearPointer = 2`
			`QueueSize = 44`
			
			1 mark for all three values correct
			A. incorrect value for `FrontPointer` if it matches the value given in part (b) (i) and incorrect value for `QueueSize` if it is equal to the value given for `QueueSize` in part (b) (i) incremented by two (follow through of errors previously made)
				1

5	**b**	**Iii**	`If DeckQueue is empty then`
			`  Report error`
			`Else`
			`  Output DeckQueue[FrontPointer]`
			`  Decrement QueueSize`
			`  Increment FrontPointer`
			`  If FrontPointer>52 Then`
			`  FrontPointer = 1`
			`EndIf`
			
			1 mark for If statement to check if queue is empty – alternative for test is `QueueSize = 0`.
			1 mark for reporting an error message if the queue is empty // dealing with the error in another sensible way – this mark can still be awarded if there is an error in the logic of the If statement, as long as there is an If statement with a clear purpose.
			1 mark for only completing the rest of the algorithm if the queue is not empty – this mark can still be awarded if there is an error in the logic of the If statement, as long as there is an If statement with a clear purpose.
			1 mark for outputting the card at the correct position
			1 mark for incrementing FrontPointer and decrementing QueueSize
			1 mark for If statement testing if the end of the queue has been
				6

			reached **1 mark** for setting FrontPointer back to 1 if this is the case – this mark can still be awarded if minor error in logic of If statement, eg >= instead of = **A.** `FrontPointer = (FrontPointer` MOD 52) + 1 for **3 marks** or FrontPointer = (FrontPointer MOD 52) for **2 marks**, both as alternatives to incrementing and using and the second If statement - **deduct 1 mark** from either of the above if QueueSize has not been decremented **A.** any type of brackets for array indexing **I.** Additional reasonable EndIf Statements **MAX 5** unless all of the steps listed above are carried out	

5	**c**		Flow of program/execution sequence determined by events // program executes relevant code-handling block/procedure/sub-routine in response to events; Example of event such as clicking a button; Message sent to program when event occurs; System/message loop executes until application closes; this receives and processes messages // use of event-listener/handler; If several events occur they are queued; **MAX 2**	2

5	**d**		The smartphone operating system: • Will not have to support as wide a range of hardware devices / peripherals // may not support external storage devices; • Will not / less likely to need to support the addition of new hardware devices to the system; • Will have minimising power consumption as a higher priority; **A.** will have more sophisticated power management features • Will run application software in a sandbox // will (more tightly) restrict access to resources by application software; • Must be capable of running on a device with less processing power // less RAM/memory // smaller memory footprint; • Needs to work with specialised hardware devices eg GPS receiver, accelerometer (Note: A relevant example must be given); **A.** Points made in reverse, but do not give two marks for one point and its reverse. **MAX 2**	2

6	**a**		*Declaring PolicyNumber as primary key:*
			`PolicyNumber INT PRIMARY KEY(NOT NULL)`
			//
			`PolicyNumber INT`
			`PRIMARY KEY(PolicyNumber)`
			Optional
				3

Declaring RegistrationNumber as foreign key:

```
RegistrationNumber CHAR(7) FOREIGN KEY
REFERENCES Vehicle(RegistrationNumber)
//
RegistrationNumber CHAR(7)
FOREIGN KEY (RegistrationNumber) REFERENCES
Vehicle(RegistrationNumber)
```

Declaring three other fields:

```
DateStarted DATE
PolicyType VARCHAR(13)
ExcessAmount SMALLMONEY
```

1 mark for PolicyNumber with sensible type and length (if required), and identified as primary key. Type can be either numeric or text.

1 mark for two other fields from RegistrationNumber, DateStarted, PolicyType, ExcessAmount with sensible data types and lengths (if required by the type) *OR* **2 marks** for all four other fields with sensible data types and lengths (if required by the type)

- Length of RegistrationNumber, if specified, must be 7.
- Length of PolicyType, if specified, must be at least 13.

1 mark for identifying RegistrationNumber as a foreign key.

MAX 3

Valid alternative SQL types are:

- Alternative types For *PolicyNumber*: smallint, mediumint, integer, any text field type (see below)
- Alternative types For *DateStarted*: smalldatetime, datetime, datetime2, datetimeoffset
- Alternative types For *PolicyType*: ENUM('Comprehensive', 'Third Party') - accept any type of quotation marks around values - accept data values in any order - accept if ENUM defined as a type separately first
- Alternative types for *ExcessAmount*: money, currency, float, real, decimal, double, numeric, int, smallint, mediumint, integer
- Alternative types for *text fields*: char, varchar, nchar, nvarchar, text, ntext, longvarchar, varchar2, nvarchar2, text, tinytext, mediumtext, longtext

Sensible non-SQL data types can also be credited but MAX 2 if any non-SQL types used.

6	**b**		`UPDATE Vehicle`
			`SET Colour = "pink"`
			`WHERE RegistrationNumber = "DF24JUT"`
			
			1 mark per correct line
			A. double or single quotes around pink and DF24JUT
			A. table names before fieldnames.
			A. pink written in any case
			DPT no quotes
			DPT for fieldname before table name.
			DPT for unnecessary punctuation – allow one semicolon at the very end of the statement, but not at the end of each clause.
			MAX 2
				2

6	**c**		`SELECT Model, Colour, Forename, Surname`
			`FROM Owner, Vehicle`
			`WHERE RegistrationNumber = "AB72XHC"`
			`AND Owner.OwnerID = Vehicle.OwnerID`
			
			1 mark for correct four fields in SELECT clause
			1 mark for correct two tables in FROM clause
			1 mark for WHERE RegistrationNumber = "AB72XHC"
			1 mark for Owner.OwnerID = Vehicle.OwnerID, joined to other condition with AND
			
			--- OR ---
			
			`SELECT Model, Colour, Forename, Surname`
			`FROM Owner INNER JOIN Vehicle ON Owner.OwnerID = Vehicle.OwnerID`
			`WHERE RegistrationNumber = "AB72XHC"`
			
			1 mark for correct four fields in SELECT clause
			1 mark for correct two tables in FROM clause
			1 mark for INNER JOIN using Owner.OwnerID = Vehicle.OwnerID
			1 mark for WHERE RegistrationNumber = "AB72XHC"
			
			Marks for SELECT and FROM statements should not be awarded if additional fields/tables included.
			Accept table names before fieldnames.
			Accept use of Alias/AS command eg FROM Vehicle AS V then use of V as table name.
			Accept insertion of spaces into fieldnames
			DPT for unnecessary punctuation – allow one semicolon at the very end of the statement, but not at the end of each clause.
			DPT for fieldname before table name.
			
			Refer responses using nested SQL queries to team leaders.
				4

6	**d**	**i**	Sequence of instructions / program / code; **NE** programming language **Note:** Do not award mark for program if candidate clearly means HTML which is executed/run/interpreted on the server (instead of the client); executed/run/interpreted when a web page is requested; to generate a web page (and its contents) / result which the server returns to the client // generating of dynamic web pages; **MAX 2**	2

6	**d**	**Ii**	**1 MARK for this point:** Retrieve RegistrationNumber/value input by user and store in variable; **R.** responses that suggest the command makes the user input the values at the point in time when the script is run **MAX 1 point from this list:** from the web page/web site/form/web server/browser/url/ request; using POST/GET methods;	2

6	**d**	**Iii**	Output the forename and surname; Back to the web server/web browser/client/terminal; **A.** display forename and surname on web page (or alternative)for both marks **R.** responses that imply output is made directly to screen	2

6	**e**		Create a new table // suitable table name given eg SafetyCertificates; with CertificateNumber as the primary key ; Include these fields in new table: CertificateNumber, DateIssued, GarageName; Add RegistrationNumber into the new table as a foreign key // as link to Vehicle table; **A.** relation for table **A.** different fieldnames for new fields if meaning the same **A.** adding the extra field ExpiryDate, but not as an alternative to DateIssued **A.** answers by example eg writing out the new table definition, SQL script to achieve changes **R.** a composite key in new table **Do not award any marks unless it is clear that a new table has been created**	3

7	**a**		Omitting unnecessary details (from a representation) // Storing only those details which are necessary (in the context of the problem being solved); **R.** responses that do not refer to abstraction in the context of data or modelling	1

7	b		**SUBJECT MARKING POINTS:** **Representation as a graph:** Vertex/node represents a station; **A.** junction between railway lines Edge/arc/line represents a (direct) connection/railway line between two stations; **R.** vector Graph must be weighted // edges have weights; Distance between two stations must be written on edge // stored with edge // weights will be distances; Could be more than one direct route between two stations; in which case shortest of the distances would be stored as the weight; Graph would be undirected as trains can travel in both directions between each station; OR Graph would be directed as some lines may only be traversable in one direction; **Note:** Only accept one of the above two points about whether the graph would be directed or undirected. Must have reason. **Implementation as array:** Each station assigned a (unique) number (to be used as array index); - **This mark available regardless of how the rest of the implementation is done** *Using an adjacency matrix:* The (adjacency) matrix would be a <u>two-dimensional</u> array (of numbers); Array contains one row and one column for each station // An n x n array is required to represent n stations; **A.** rows and columns labelled with stations for BOD mark If there is a (direct) connection between the two stations, store the distance between the two stations at the intersection of the row/column for the stations; If there is no (direct) connection between the two stations, store a value to indicate this at the intersection of the row/column for the stations; **A.** examples of values eg 0, ∞, NULL that could not be valid distances including any alphanumeric indicator *Using an adjacency list as a 2d array of numbers:* Adjacency list could be stored in a <u>two-dimensional</u> array (of records or similar); In one dimension there would have to be n rows/columns for n stations // one row/column per station; In the other dimension the number of columns/rows would be	8

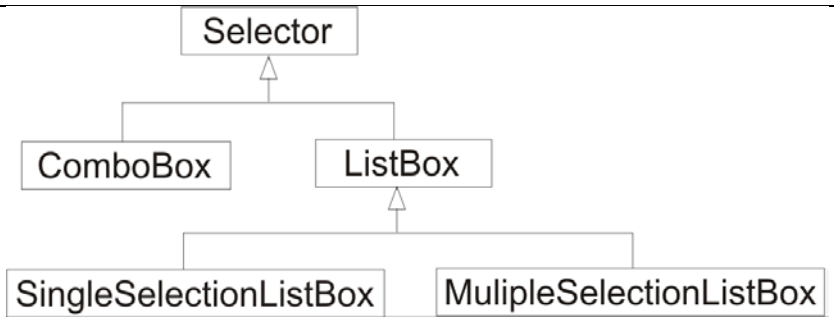
			determined by the highest degree of any vertex // the maximum number of neighbours a vertex has // the maximum number of (direct) connections that any station has; If a station is (directly) connected to another station then in the row/column for the first station, a new entry (record) would be made consisting of the number of // an identifier for the second station and the distance to it; **NE.** just to state identifier, must have distance as well **Note:** Also accept implementation in two two-dimensional arrays, with one storing the stations and the other the distances, as long as made clear station identifiers and distances stored in corresponding positions. *Using an adjacency list as a 1d array of strings:* Adjacency list could be stored in a (one-dimensional) array of strings; One row per station; The string for a station contains, for each station that it is connected to, the station identifier/name and distance; Use of a delimiter between values; **REFER ANY OTHER WORKABLE SOLUTION TO A TEAM LEADER** **If comparison made between adjacency matrix and adjacency list (not asked for):** Adjacency list might be more efficient (in terms of storage space) as graph is likely to be sparse // as few edges between vertices // as most stations only (directly) connected to a small number of other stations; Adjacency matrix might be more efficient (in terms of speed) as shortest route finding algorithm is likely to need to lookup many distances when computing a route; **Note on use of diagrams:** Candidates may choose to use diagrams to help clarify their responses. When marking, use may be made of such diagrams to help clarify understanding of the written description, however as this question assesses quality of written communication, marks should be awarded for the written description, not directly for the diagrams themselves.	

HOW TO AWARD MARKS:

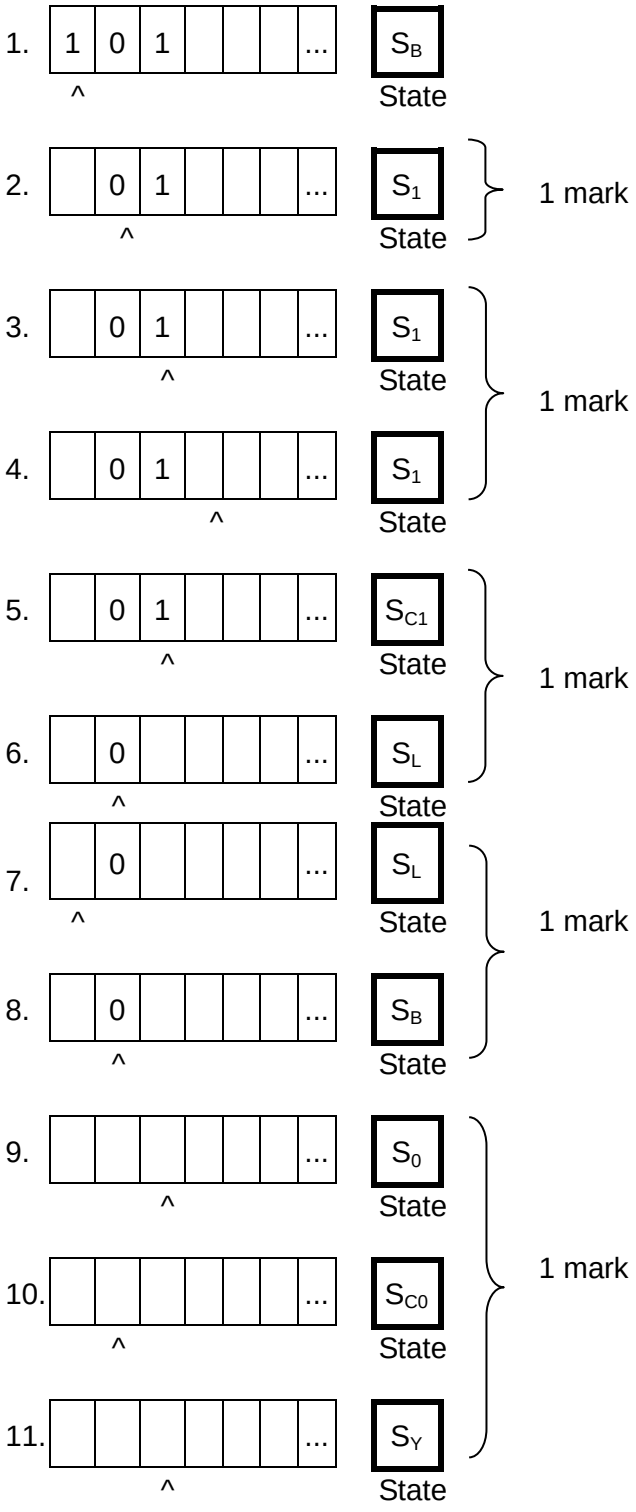
Mark Bands and Description	
7-8	*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and all 5 of the quality of written communication criteria (QWCx).* *SUB* Candidate has covered the graph representation and array implementation in detail and all or almost all of the required detail for an implementation is present. The candidate has made at least seven subject-related points. *QWC1* Text is legible. *QWC2* There are few, if any, errors of spelling, punctuation and grammar. Meaning is clear. *QWC3* The candidate has selected and used a form and style of writing appropriate to the purpose and has expressed ideas clearly and fluently. *QWC4* Sentences (and paragraphs) follow on from one another clearly and coherently. *QWC5* Appropriate specialist vocabulary has been used.
5-6	*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of written communication criteria (QWCx).* *SUB* Candidate has covered both the graph representation and array implementation, making some valid points for each, but the level of detail may not be sufficient to implement. The candidate has made at least five subject-related points. *QWC1* Text is legible. *QWC2* There may be occasional errors of spelling, punctuation and grammar. Meaning is clear. *QWC3* The candidate has, in the main, used a form and style of writing appropriate to the purpose, with occasional lapses. The candidate has expressed ideas clearly and reasonably fluently. *QWC4* The candidate has used well-linked sentences (and paragraphs). *QWC5* Appropriate specialist vocabulary has been used.
1-4	*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of written communication criteria (QWCx).*

			SUB Candidate has made some relevant points but the description is either lacking in detail or only covers one of the graph representation or array implementation. *QWC1* Most of the text is legible. *QWC2* There may be some errors of spelling, punctuation and grammar but it should still be possible to understand most of the response. *QWC3* The candidate has used a form and style of writing which has many deficiencies. Ideas are not always clearly expressed. *QWC4* Sentences (and paragraphs) may not always be well-connected. *QWC5* Specialist vocabulary has been used inappropriately or not at all. 0 Candidate has made no relevant points. Note: Even if English is perfect, candidates can only get marks for the points made at the top of the mark scheme for this question. If a candidate meets the subject criterion in a band but does not meet the quality of written communication criteria then drop mark by one band, providing that at least 4 of the quality of language criteria are met in the lower band. If 4 criteria are not met then drop by two bands.	

8	**a**		A class/subclass has/shares/can access properties and methods of the (parent) class it is derived from; Building a hierarchy of classes with each child class inheriting access to its parent class' methods and properties; Relationship between two object (types) in which one object (type) is a kind of the other; **MAX 1** **A** Just one of properties and methods, do not need both. **A** Use of the word "inherits" in the response only if the relationship between parent and subclass is stated explicitly otherwise it is **NE** **A** The following as alternatives to properties: fields, attributes, characteristics, data. **A** The following as alternatives to methods: procedures, functions, code. **A** The following as alternatives to parent: base, super. **A** The following as alternative to child: descendent, subclass, derived.	1

8	**b**		 **1 mark** for Selector at top of diagram with ComboBox and ListBox directly underneath it and linked to it and no other labels linked to it; **1 mark** for ListBox with SingleSelectionListBox and MultipleSelectionListBox directly underneath it and linked to it, and no other labels linked to it (except Selector above); **MAX 1** of the above two marks if any additional links drawn in **1 mark** for correctly styled diagram, i.e. lines drawn as arrows and boxes (any shape) around labels; - ***This mark is only available if candidate has already achieved at least one mark for correct contents of the diagram.*** **A.** arrows drawn as:  **A.** any type of arrowheads. **A.** diagram rotated through 90/180/270 degrees. **A.** arrows draw wrong way round (but cannot get mark for correctly styled diagram). **A.** class diagrams.	3

8	**c**		ComboBox = Class (Selector)
			  Public
			  Procedure SelectItemFromList
			  Procedure Display
			  Procedure KeyPressed
			  Function GetTextTyped
			  Function GetSelectedItemNumber
			  Procedure SetAllowNonListInputs
			  Private
			  TextTyped: String
			  SelectedItemNumber: Integer
			  AllowNonListInputs: Boolean
			End
			
			Accept answers that use different notations, so long as meaning is clear.
			Accept any sensible names for subroutines, except SelectItemFromList which must have this name as it overrides a procedure in the parent class.
			
			1 mark for correct header including name of class (ComboBox) and parent class (Selector);
			1 mark for overriding the SelectItemFromList procedure (it is not necessary to state that overriding is occurring but must be public);
			2 marks for defining all 5 other extra functions/procedures needed, all identified as being public (keyword public is optional if they are declared before variables);; **OR 1 mark** if at least 2 of them defined;
			1 mark for defining all 3 extra variables, with appropriate data types and identified as being private;
			
			A. Array of characters as alternative to string for TextTyped
			A. Any sensible numeric types for SelectedItemNumber (must be whole numbers)
			A. Answers that indicate separately that each variable is private or each method is public
			A. Two procedures instead of one for setting the value of AllowNonListInputs by result, eg, Procedure AllowTextInputs and Procedure OnlyAllowSelection
			A. Procedure instead of Function and vice-versa
			I. parameters to methods, minor changes to names that do not affect clarity.
			R. do not award marks for functions/procedures with the same name as variables
			DPT if any additional functions/procedures/variables declared do not award the first of the three marks for correctly defining new functions and variables, but award subsequent marks. However, do not penalise answers that include any of the following procedures/functions: `GetAllowNonListInputs, SetTextTyped, SetSelectedItemNumber`
				5

9	**a**		One mark per bracketed section.  **Must have correct tape contents and state for each mark** **A.** blank symbols instead of empty cells **DPT** If the read/write head is not drawn on some rows, this should result in the loss of the mark on the first occasion that it is missing only. Marks should be awarded for subsequent rows,	5

			even if the read/write head is not drawn.	
9	**b**		(After a 0 has been read,) the rules keeps moving the read/write head to the right (preserving the contents of the tape); Until a blank symbol is encountered / the end of the number is reached, then the state is changed to S_{C0} (and the head is moved left/direction reversed); **Note:** To achieve the first mark, it must be clear that the head moves right regardless of whether a 0 or 1 is read and also that this is a repeated process ie not just moving one place right. **Note:** If it is stated that the process of moving continues until the end of the number is reached, then it can be inferred that the head was moving right for the first mark, if this was not explicitly stated. **Note:** Marks should not be awarded for just explaining what the rules do individually.	2

9	**c**		It reads instructions one at a time // reads instructions in sequence // deals with instructions line by line; And executes these instructions; Instructions are / transition function is stored on the tape; **A.** "rules" for "instructions" **MAX 2**	2

10	**a**	**i**	192.168.0.x where x is not 0 or 255;	1

10	**a**	**ii**	192.168.2.x where x is not 0 or 255;	1

10	**a**	**iii**	192.168.2.y where y is not 0 or 255 and is not the same as x in (ii);	1

10	**b**		**Reason:** To reduce (network) congestion//improve throughput//*to cut the number of collisions**; **A** faster operation/transmission; **Explanation:** *by cutting the number of collisions**//by reducing the number of stations/computers connected to each section of cabling// because two computers in one segment can communicate at the same time as two computers in another segment; **Note:** * = Do not award two marks for cutting the number of collisions – only award one for either reason or explanation. **Reason:** To improve security; **Explanation:** by localising packet transmission to one segment; **Reason:** To improve reliability; **Explanation:** By limiting effect of cable failure to one segment; **R.** answers referencing the computers not working at all **Award marks for either:** • **one reason + explanation** • **two reasons** • **two explanations**	2

10	**c**	**i**	No need for maintenance // no need to upgrade // no need to install patches for software // could employ fewer technical staff; Lower hardware requirements for computers (as processing done on web server); **A.** examples of lower hardware requirements but **R.** just cheaper hardware No (high) one-off purchase cost;Platform independence // can access the software on many devices; **A.** examples eg PC and tablet. Software can be used from anywhere that there is an Internet connection // from outside of office; **Note:** To award this point must be clear that can be accessed from outside of office, just "can be accessed from any computer" is not enough. Can still access software and data if a specific computer is not working; **A.** reduced management cost/effort when a reason is given, such as no need to install software on each computer, but just "does not need to be installed on each computer" which is not enough on its own. **MAX 2**	2

10	**c**	**ii**	Reliance on Internet // unreliable internet connection may mean software inaccessible; Reliance on the company that develops the software to keep providing the service; Slow connection speed may make software difficult/annoying to use; Concern over security of saved documents // security of transmission; May be an ongoing cost to pay for using the software; Lack of control over which version to use / when upgrades happen; Software may slow when used by many users simultaneously; Higher cost (to company) of fast internet connection to connect many computers to SaaS; **MAX 1**	1

10	**d**		LAN usually baseband whilst WAN broadband* // only one communication can take place at a time on a LAN whereas multiple communications can take place simultaneously on a WAN; LAN communication links have higher speeds than WAN; LAN has lower latency than WAN; Lower error rates on LAN than WAN; Communications medium in LAN likely to be privately owned, whereas likely to be leased/publicly owned in a WAN; Use different protocols (at link layer / hardware level); Different hardware required to connect (**A.** examples); WAN may have greater security risks (as data transmitted over larger area, on public system, through more servers or devices); **A.** WAN may use satellites / microwave whilst LAN may use cables / radio / WiFi* - this point cannot be awarded for just saying WAN uses cables on LAN radio or vice-versa as both LAN and WAN can use either of these **MAX 2** **Only one side of the difference needs to be provided (as the other is implicit) except for the points marked with an * for which both sides are needed.**	2