A-level

Computing

COMP3/Unit 3: Problem Solving, Programming, Operating Systems, Databases and Networking

Mark scheme

2510

June 2013

Version: 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

To Examiners:

1. **When to award '0' (zero) when inputting marks on CMI+:** A mark of 0 should be awarded where a candidate has attempted a question but failed to write anything creditworthy. Insert a hyphen when a candidate has not attempted a question. By these two actions the Principal Examiner will be able to distinguish between the two (nothing credit worthy/unattempted) when analysing any statistics.

2. This mark scheme contains the correct responses which we believe that candidates are most likely to give. Other valid responses are possible to some questions and should be credited. Examiners should refer off mark scheme responses that they believe are creditworthy to a Team Leader.

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 - means "Don't penalise twice". 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>Sub-part</th><th>Marking Guidance</th><th>Marks</th><th>Comments</th></tr>
<tr><td>1</td><td>(a)</td><td></td><td>One mark per correct response.

<table>
<tr><th>Construct</th><th>Example</th><th>Valid ?</th></tr>
<tr><td>identifier</td><td><code>Player2name</code></td><td>No;</td></tr>
<tr><td>parameter</td><td><code>x,y:bool</code></td><td>Yes;</td></tr>
<tr><td>procedure-def</td><td><code>procedure square(s:real)</code></td><td>No;</td></tr>
<tr><td>procedure-def</td><td><code>procedure rect(w:int,h:int)</code></td><td>No;</td></tr>
</table>
A alternative clear indicators of Yes/No such as Y/N, True/False and Tick/Cross.</td><td>4</td><td></td></tr>
<tr><td>1</td><td>(b)</td><td>(i)</td><td>The <code><type></code> rule has an extra type <code>char</code>;
The <code><procedure-def></code> rule does not allow a procedure without parameters // cannot be just an identifier;</td><td>2</td><td></td></tr>
</table>

			Accept answers comparing the figures the other way around, i.e. • The type rule does not allow a char • The procedure does not have to have parameters / can be just an identifier		

1	**(b)**	**(ii)**	Required as there can be a list of parameters // required as there can be more than one parameter; BNF does not support iteration // BNF can only achieve iteration through recursion // would need infinite number of rules otherwise // recursion allows for more than one parameter; **MAX 1** **A.** Input for parameter **NE.** Rule needs to loop	**1**	

2	**(a)**		One mark per correct answer: **Value description** / **Correct letter (A-D)** A negative value. – D; The smallest positive value that can be represented. – A; A value that is not valid in the representation because it is not normalised. – C; If a letter is used more than once then mark as correct in the position that is correct.	**3**	

2	**(b)**		**1 method mark** for either: • showing correct value of both mantissa and exponent in denary • showing binary point shifted 6 places to right in mantissa • indicating that final answer calculated using answer = mantissa x $2^{exponent}$ Mantissa = 0.625 // 5/8 Exponent = 6 **1 mark** for correct answer Answer = 40 *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**	

2	**(c)**		**2 marks** for working: Correct representation of 7.75 in fixed point binary: 111.11; **A.** leading and trailing 0s. Bits flipped: 000.00 // 1000.00; **A**. leading 1s Correct representation of -7.75 in fixed point twos complement: 1000.01; **A**. leading 1s Showing the correct value of the exponent in denary (3) or binary (11) // showing the binary point being shifted 3 places; *Note: Award both working marks if bit pattern 1.00001 is shown anywhere* **MAX 2** **1 mark** for correct mantissa and exponent together: 1 ● 0 0 0 0 1 0 0 Mantissa 0 0 1 1 Exponent *If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.* *Working marks can be awarded for work seen in the final answer e.g. correct exponent.*	**3**	

2	**(d)**	**(i)**	0.025 // 6.9-6.875 // 1/40 **R.** -0.025 **A.** award **BOD** mark if correct method has been shown i.e. 6.9-6.875 but candidate has then made an error performing the subtraction operation	**1**	

2	**(d)**	**(ii)**	0.003623 // 0.025/6.9 // 1/276 **A.** 0.3623% **A.** answers rounded to at least two significant figures **A.** follow-through of incorrect answer to part 2di **A.** award **BOD** mark if correct method has been shown but candidate has then made an error performing the division operation **R.** if shown that incorrect method used e.g. dividing by 6.875, even though this arrives at an answer that is the same when written to 2 significant figures	**1**	

2	**(d)**	**(iii)**	**Alternative 1:** Adjust the mantissa; To use more bits; **A.** "longer" for "more bits" but **R.** "larger", "increase size" **Alternative 2:** Reallocate (one) bit; from the exponent to the mantissa; **A.** bits **Alternative 3:** Infer one of the two bits on either side of the binary point (from the other, as they must both be different); use the freed up bit to store one more significant digit in the mantissa// use the freed up bit to represent mantissa more accurately;	**2**	

3	**(a)**		Serial send one bit at a time / after each other whereas parallel sends multiple bits simultaneously/at same time; **A.** "data" for "bits" in the context of parallel transmission Serial uses a single wire/cable/path/line whereas parallel uses several/multiple wires/cables/paths/lines; **R.** answers that refer to multiple channels achieved by sharing bandwidth *Both sides of point must be made.*	**2**	

3	**(b)**		*Parity Bit:* 1; *Start bit, Stop Bit :* Can be either 0 or 1, but must both be different to get mark;	**2**	

3	**(c)**		Receiver and transmitter (clocks) do not need to be/are not (exactly) synchronised // transmission of data without use of external clock signal // receiver and transmitter clock only synchronised at start of/for length of transmission // start bit used to synchronise clocks of sender and receiver // data sent as soon as available rather than waiting for clock pulse/synchronisation symbol; **NE** data sent as soon as possible without waiting for receiver to be ready // receiver does not know when data will arrive	**1**	

4	**(a)**		+; 4, 9, 6; (in any order)	**2**	

4	**(b)**		A: Store the data/value (in the vertices/nodes); **A.** holds the expression B: Left pointer // points to the left child / left sub tree; C: Right pointer // points to the right child / right sub tree; **A** “indicates”, "index" or other synonym for “points” / “pointer” **R.** Stores left/right subtree	**3**	

4	**(c)**		The node has no left child / sub tree; **A** there is nothing to the <u>left</u> **A** this is a null <u>pointer</u>	**1**	

4	**(d)**		One mark for each area outlined with a dark rectangle. Lines that are not outlined can be missed out. Alternative 1 **Pos** / **Output** 1 / 2 / 4 1 / 3 / 4 / 9 3 / 5 / 6 3 / * 1 / + Alternative 2 **Pos**: 1, 2, 1, 3, 4, 3, 5, 3, 1 **Output**: 4, 9, 6, *, + Mark against whichever alternative gives the highest mark. **Stop marking as soon as incorrect output is given.**	**4**	

4	**(e)**		Post-order; **A.** Depth-first **A.** Depth-first search as BOD **TO.** Depth-first pre/in-order	**1**	

4	**(f)**		(4 + 9 * 6 in) Reverse Polish (Notation) // Postfix (Notation) // RPN;	**1**	

5	**(a)**	**(i)**	192.168.0.x where x is not 0 or 255; Must be a specific IP address **R.** addresses that include port numbers	**1**	

5	**(a)**	**(ii)**	192.168.1.x where x is not 0 or 255; Must be a specific IP address **R.** addresses that include port numbers	**1**	

5	**(a)**	**(iii)**	192.168.1.y where y is not 0 or 255 and is not the same as x in (ii); Must be a specific IP address **R.** addresses that include port numbers	**1**	

5	**(b)**		Bus (topology/network); **A** Line	**1**	

5	**(c)**		255.255.255.0 / FFFFFF00 / 11111111 11111111 11111111 00000000;	**1**	

5	**(d)**		(An operating system that is optimised to) provide (one or more specialised) services to (network) clients; **A.** description of examples of services e.g. logging on, sharing printers, but just the example of accessing files is not enough as this is in the question there needs to be additional explanation if files is used as an example, e.g. managing quotas, security of files. **R.** answers that imply that server does all processing i.e. confusion with thin client.	**1**	

5	**(e)**	**(i)**	Use of Wired Equivalent Privacy/WEP/ WPA/WPA2/WiFi Protected Access; (Strong) encryption of transmitted data // use of Advanced Encryption Standard/AES; **R** encoding Use of Extensible Authentication Protocol/EAP; User/computer must enter/send a passphrase/certificate at start of communication before laptop allowed to connect; **A** key for passphrase **A** only allow password if used in correct context ie for accessing network, not for logging on to a sever or just having a password Access point checks MAC/hardware address of laptop and only allows computers with a MAC/hardware address in a list of approved addresses to connect; **R** IP address Disable broadcast of SSID/identity; Reduce / limit power of transmitter; Use of two/multi-factor authentication;	**1**	

5	**(e)**	**(ii)**	Longer range // faster transmission speeds // higher bandwidth // more simultaneous connections; **A.** reverse of points e.g. "Bluetooth only has a short range" **R.** Bluetooth can only connect two devices at once	**1**	

5	**(f)**		**SUBJECT MARKING POINTS:** **Internal:** • Student's computer uses subnet mask (and destination/web server's IP address) to determine if destination computer/web server is on same subnet // identify not on same subnet • Up to two marks from description (in separate section below) of how subnet mask is used • Packet is sent (from student's computer) to Router (1) • Router 1 identifies that destination is outside the LAN so forwards packet to Gateway **External:** • Hierarchical organisation of routers • Example of hierarchical organisation of routers e.g. passed up to a national router, transferred internationally and then passed back down a hierarchy • Path to take selected by each router (not determined at start) **NE** passed from router to router • Route may change as a result of e.g. congestion, technical problems **Either:** • (Possible) repackaging of packet to use different protocol (e.g. Gateway may change protocol) • Route determined using the (Network ID part of the destination) IP address (Note: can infer "IP address" if just "address" is stated, if previously candidate has written about an IP address) • Use of router tables / criteria to determine next hop / (step of) path • Router decrementing "time to live" of packet • Source and destination MAC addresses changed at each router // MAC addresses used for each "hop" **How subnet mask used (MAX 2 points):** • AND operation of subnet mask with student's computer's IP address • AND operation of subnet mask with web server's IP address • Result (of AND operation) is the network ID; • Network IDs compared • If they are the same, then the computers are on the same subnet	**8**	

A interchangeable use of subnet ID and network ID

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 language criteria (QLx).* *SUB* Candidate has covered both internal and external routing in detail and 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 language criteria (QLx).* *SUB* Candidate has covered both internal and external routing, although one may be in more detail than the other and 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 language criteria (QLx). *SUB* Candidate may not have covered both internal and external routing, but has covered at least one of them. Up to four relevant points have been made. *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 language 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.		

5	**(g)**		Any two points from the list below. Candidate only needs to make one side of point, the other can be implied. Do not award marks for two sides of same point. **Routable** / **Non-Routable** (Globally) unique. / Many computers/devices may have same address. Allocated by a central/regional issuing authority (**A** example). / Not allocated centrally // allocated by a home user/company/ISP. Can be connected to directly over the Internet / from outside private network / Difficult/impossible to connect to over Internet // from outside of network. Owner can be looked up using WHOIS protocol / Owner cannot be looked up using WHOIS protocol **A.** non-routable IP addresses more secure as cannot be connected to over Internet/from outside network **A.** can identify location from a routable IP address	**2**	

6	**(a)**		Most efficient: C // O(n) **A.** n B // O(n^2) **A.** n^2 Least efficient: A // O(a^n) **A.** a^n	**1**	

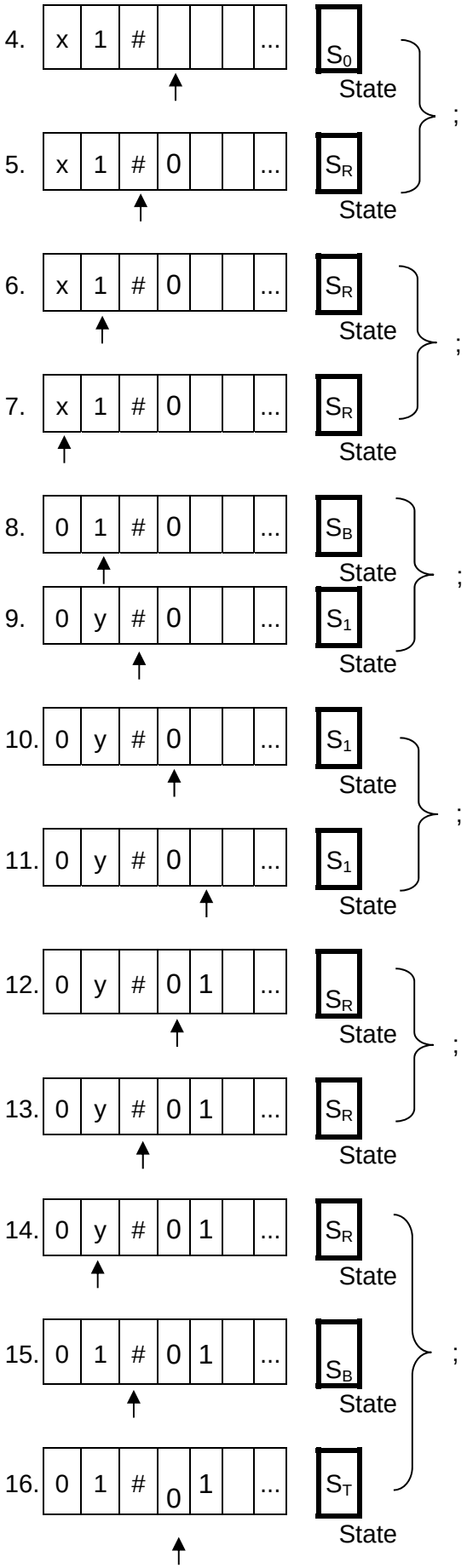
6	**(b)**	**(i)**	The problem can be solved; But not in polynomial time // only in exponential (or worse) time // it takes an unreasonable amount of time to do so // can't be solved quickly enough for it to be useful; **A** takes too long for a computer to solve but **NE** just takes a long time **A** "algorithm exists" for can be solved **A** answers relating to space rather than time **TO** of the solving mark, if states that can be solved in polynomial/reasonable time	**2**	

6	**(b)**	**(ii)**	**Problem** / **Intractable? (Tick One)** The travelling salesman problem. / ✓; The problem of sorting a list of names into alphabetic order. / The Halting problem. / **A** alternative indicators for ticks **Do not award mark if more than one box is ticked.**	**1**	

7	**(a)**	**(i)**	❷ S_1 **A.** 1, State 1 ❸ S_T **A.** T, State T Both answers correct to get mark;	**1**	

7	**(a)**	**(ii)**	$\delta (S_B, 0) = (S_0, x, \rightarrow)$; **A.** 0, x, → or 0 | x | → **R** if additional rules listed **I** minor transcription errors e.g. missing , (δ	**1**	

7	**(a)**	**(iii)**	$\delta (S_R, x) = (S_B, 0, \rightarrow)$ and $\delta (S_R, y) = (S_B, 1, \rightarrow)$; **A.** x, 0, → or x | 0 | → and y, 1, → or y | 1 | → **R** if additional rules listed **I** minor transcription errors e.g. missing , (δ	**1**	

7	**(b)**		One mark per bracketed section.  	**6**	

			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, even if the read/write head is not drawn.		

7	**(c)**	**(i)**	Mark symbol currently being copied // to indicate how much of the string has been copied so far // to indicate where to return to (to copy next symbol); **A.** placeholders **NE.** x represents 0, y represents 1	**1**	

7	**(c)**	**(ii)**	Copy a string//copy a binary number // copy a bit pattern; **A.** Repeat	**1**	

8	**(a)**		Static structures have fixed (maximum) size whereas size of dynamic structures can change // Size of static structure fixed at compile-time whereas size of dynamic structure can change at run-time; Static structures can waste storage space/memory if the number of data items stored is small relative to the size of the structure whereas dynamic structures only take up the amount of storage space required for the actual data; Dynamic data structures (typically) require memory to store pointer(s) to the next item(s) which static structures (typically) do not need // Static structures (typically) store data in consecutive memory locations, which dynamic data structures (typically) do not; **MAX 2** **A** just one side of points, other side is by implication **NE.** Dynamic data structures use pointers	**2**	

8	**(b)**		Not possible to simply insert item into middle of list; Must move all items that should come after the new process down in the array; **NE** move all data Moving items is time consuming; In a dynamic implementation, insertion achieved by adjusting pointers; **MAX 2**	**2**	

8	**(c)**		Priority (queue);	**1**	

8	**(d)**	**(i)**	Memory allocated/deallocated at run-time/for new items (to dynamic data structure); (Provides a) pool of free/unused/available memory; **NE** to store new items **MAX 1**	**1**	

8	**(d)**	**(ii)**	(Memory) address // memory location // position in memory; **NE** position or location without reference to memory **R** index	**1**	

8	**(d)**	**(iii)**	**OVERALL GUIDANCE:** Solutions should be marked on this basis: • Up to **4 marks** for correctly locating the position to insert the new process at. • Up to **4 marks** for creating a new node and storing the correct data into it and the associated pointers. Some marks can be awarded for this even if the locating process is incorrect/missing. The full 7 marks should only be awarded for a complete fully working solution. If any steps are missed out, then award a maximum of six marks (MAX 6). The addition of any unnecessary steps that do not stop the algorithm working should not result in a reduction in marks. Responses should be accepted in pseudo-code or structured English. ***If you are unsure about the correctness of a solution please refer it to a team leader. Also, responses in prose should be referred to team leaders.*** **SPECIFIC MARKING POINTS:** *Correctly locating insertion point (MAX 4):* 1. Initialising current node pointer to start pointer; 2. Use of loop to attempt to move through list (regardless of correct terminating condition); 3. Advancing current node pointer within	**7**	

			loop; 4. Correctly maintaining a pointer to the node before the position that the new node should be inserted at; 5. Sensible condition to identify place to insert (suitable terminating condition for loop or condition in selection statement); *Correctly inserting new process (MAX 4):* 6. Create a new node / obtain new node from heap; 7. Store new process name and priority (in new node); 8. Update NextNodePointer of node before newly inserted one to point to new node; 9. Set NextNodePointer of new node to point to node after it; Mark point 2 can only be awarded if, within the loop, current node pointer is being changed (even if not correctly changed). Mark point 4 can only be awarded if mark point 3 had been awarded. Mark point 5 can be awarded if there is a sensible condition, even if current node pointer is not correctly updated. Mark points 8 and 9 can only be awarded if the correct insertion point has been found. *For any solution:* **A** use of either while or repeat loops, as long as logic is correct. **A** storage of values into new node in any order, and regardless of whether the node has been created or not. **A** use of ^ symbol to indicate the value stored at an address referenced by a pointer, for example CurrentNodePointer^.Priority indicates the value stored in the Priority field of the node pointed to by the pointer CurrentNodePointer. **A** use of alternative variable names so long as the meaning is clear. **EXAMPLE SOLUTIONS AND MARKS:** These four examples show where marks should be awarded in some possible solutions (subject to a maximum mark of 7): **Example 1:**		

```
CurrentNodePointer = StartPointer;
Repeat
  PreviousNodePointer =
   CurrentNodePointer;
  CurrentNodePointer =
   NextNodePointer of current node;
Until priority of process in current
 node < priority of process to add
//
 priority = "Low";;
Create new node;
Store new process name (and
priority)
 in new node;
New node's NextNodePointer = Next
 NodePointer of item at position
 PreviousNodePointer;
NextNodePointer of item at position
 PreviousNodePointer = Address of
new
 node;
```

Example 2:

This is an alternative way of expressing Example 1:

1. Load the Start Pointer value into the Current Node Pointer;
2. Copy value from Current Node Pointer into Previous Node Pointer;
3. Set Current Node Pointer to Next Node Pointer of current node;
4. If the priority of the data item at the current node is higher than or the same as the priority of the process to be added; then go back to step 2;
5. Create a new node;
6. Store the name of the new process (and its priority) in the new node;
7. Copy value from Next Node Pointer of list entry at position stored in Previous Node Pointer into the Next Node Pointer of the new node;
8. Set the Next Node Pointer of the list entry at position stored in the Previous Node Pointer to point to the new node;

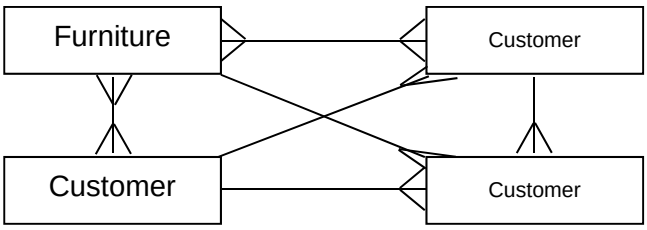
Example 3:

```
CurrentNodePointer = StartPointer;
Inserted = False
While; Inserted = False Do
  If Current Node's priority < new item
   priority // = "Low"; Then
    Create new node;
```

			Store new process name (and priority) in new node;
			New node's NextNodePointer = CurrentNodePointer;
			NextNodePointer of item at position PreviousNodePointer = Address of new node;
			Inserted=True
			End If
			PreviousNodePointer = CurrentNodePointer;
			CurrentNodePointer = NextNodePointer of current node;
			End While
			
			Example 4:
			
			CurrentNodePointer = StartPointer;
			While; not at end of list // While CurrentNodePointer <> Nil // While priority of process at CurrentNodePointer >= priority of process to add Do
			If Current Node's priority is the required priority // = "Normal"; Then
			LastNodeOfCurrentPriorityPointer = CurrentNodePointer;
			End If
			CurrentNodePointer = NextNodePointer of current node;
			End While
			Create new node;
			Store new process name (and priority) in new node;
			New node's NextNodePointer = Next NodePointer of item at position LastNodeOfCurrentPriorityPointer;
			NextNodePointer of item at position LastNodeOfCurrentPriorityPointer = Address of new node;		

9	**(a)**		**What means:** every attribute (in relation) is dependent on the key; the whole key; and nothing but the key; **R.** Everything OR (relations) contain no repeating groups (of attributes) // data is atomic; no partial dependencies; no non-key dependencies; **R** No repeated columns/attributes/data OR every determinant (in the relation) is a candidate key;;	**4**	

			MAX 2 **Why important:** Eliminate update anomalies; **A** Example Eliminate insertion anomalies; **A** Example Eliminate deletion anomalies; **A** Example Eliminate data inconsistency // improve consistency // avoid inconsistency problems; *Minimise data duplication // no unnecessary repeated data; **A** Reduce for minimise **R** eliminate *Eliminate data redundancy; **A** Reduce/minimise for eliminate **NE** Easier to update/insert/delete without concrete example or good explanation **NE** Less errors when updating/inserting/deleting without concrete example or good explanation **NE** Saving space/memory **NE** Easier to query **Award marks to points made anywhere across 9(a)** **Can only award one of the two marks indicates by asterisks (*)** **MAX 2**		

9	**(b)**		One mark per correct relationship.  **MAX 2 if any incorrect relationships drawn** **MAX 3**	**3**	

9	**(c)**		FurnitureID INT PRIMARY KEY NOT NULL // FurnitureID INT PRIMARY KEY(FurnitureID) Optional FurnitureName VARCHAR(30) Category VARCHAR(10) Price SMALLMONEY SupplierName VARCHAR(20) **1 mark** for FurnitureID, with sensible data type and identified as primary key **1 mark** for two other fields with sensible data types and lengths *OR* **2 marks** for all four other fields with sensible data types and lengths **A** any sensible types. Lengths do not need to be specified.	**3**	Allow lengths after numeric types e.g. INT(11) as these are allowed in MySQL.

			Valid alternative SQL types are: • Alternative types For *FurnitureID*: smallint, mediumint, integer, any text type (see below) • Alternative types for *Price*: money, float, real, decimal, double, numeric, int, smallint, mediumint, integer, currency • Alternative types for *text fields*: char, varchar, nchar, nvarchar, ntext, longvarchar, varchar2, nvarchar2, text, tinytext, mediumtext, longtext **Answers using a syntax that is clearly not SQL should be awarded zero marks. But:** • **ignore punctuation errors e.g. unnecessary colons or commas.** • **answers in SQL style syntax but using non-SQL data types can be credited but MAX 1 of 2 for data types if any non-SQL types used.**		

9	**(d)**		SELECT CustomerName, TelephoneNumber FROM Customer, CustomerOrder, CustomerOrderline WHERE FurnitureID=10765 AND Customer.CustomerID= CustomerOrder.CustomerID AND CustomerOrder.OrderID= CustomerOrderLine.OrderID ORDER BY CustomerName (ASC) **1 mark** for correct two fields in SELECT clause **1 mark** for correct three tables in FROM clause **1 mark** for FurnitureID = 10765 **1 mark** for Customer.CustomerID = CustomerOrder.CustomerID, joined to other conditions with AND **1 mark** for CustomerOrder.OrderID = CustomerOrderLine.OrderID, joined to other conditions with AND **1 mark** for ORDER BY CustomerName, ASC is optional --- OR --- SELECT CustomerName, TelephoneNumber FROM Customer INNER JOIN CustomerOrder ON Customer.CustomerID=CustomerOrder.CustomerID INNER JOIN CustomerOrderLine ON CustomerOrder.OrderID=CustomerOrderLine.Or	**6**	

			derID WHERE FurnitureID = 10765 ORDER BY CustomerName (ASC) **1 mark** for correct two fields in SELECT clause **1 mark** for correct three tables in FROM clause **1 mark** for INNER JOIN using Customer.CustomerID=CustomerOrder.CustomerID **1 mark** for INNER JOIN using CustomerOrder.OrderID=CustomerOrderLine.OrderID **1 mark** for FurnitureID = 10765 **1 mark** for ORDER BY CustomerName, ASC is optional Marks for SELECT and FROM statements should not be awarded if additional fields/tables included. Marks can be awarded for the conditions in the WHERE statement even if the required tables are not present in the FROM. Accept FurnitureID with no quotation marks, single quotation marks or double quotation marks. Accept table names before fieldnames. Accept use of Alias/AS command e.g. FROM Customer AS C then use of C as table name. Accept insertion of spaces into fieldnames Ignore unnecessary clause CustomerOrderLine.FurnitureID=Furniture.FurnitureID **I** unnecessary brackets **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.**		

9	**(e)**		One mark for tick in correct row. Do not award mark if more than one row is ticked. **Command** / **Correct? (Tick One)** ALTER TABLE ✓ CREATE FIELD INSERT COLUMN	**1**	

Statistical data and information on grade boundary ranges www.aqa.org.uk/over/stat.html

UMS conversion calculator www.aqa.org.uk/umsconversion