A-LEVEL

# Computing

COMP3/Problem Solving, Programming, Operating Systems, Databases and Networking
Mark scheme

2510
June 2015

Version 1: Final Mark Scheme

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

<table>
<tr><th>Qu.</th><th>Part</th><th>Sub-part</th><th>Marking guidance</th><th>Mark</th></tr>
<tr><td>1</td><td>a</td><td></td><td>
<table>
<tr><td>Current State</td><td>$S_3$</td><td>$S_3$</td><td>$S_3$</td></tr>
<tr><td>Input Symbol</td><td>a</td><td>b</td><td>c</td></tr>
<tr><td>Next State</td><td>$S_6$</td><td>$S_6$</td><td>$S_4$</td></tr>
</table>
1 mark for all six correct values in the bold rectangular area<br>The columns do not have to be in the same order as shown, but the pairings must be correct i.e. (a - $S_6$, b - $S_6$, c - $S_4$).<br>A. 4 for $S_4$ and 6 for $S_6$
</td><td>1</td></tr>
<tr><td>1</td><td>b</td><td></td><td>$S_3$<br>A. 3<br>I. An additional name given to the state eg “State 3”</td><td>1</td></tr>
<tr><td>1</td><td>c</td><td></td><td>To ensure that a non-valid string is trapped // prevent the accepting state being reached;<br>A. To capture invalid input<br>A. To capture strings that are too long / have extra characters<br>N.E. Infinite loop / state cannot be left</td><td>1</td></tr>
<tr><td>1</td><td>d</td><td></td><td>a(bc)*a // a(bc)+?a // (aa)|(a(bc)*a) // (aa)|(a(bc)+?a) // (aa)|(a(bc)+a)<br><br>1 mark for recognising an a at both ends<br>1 mark for correctly recognising 0 or more repetitions of bc<br>I. ^ and $ at start and end of expression<br>A. Any type of bracket</td><td>2</td></tr>
<tr><td>1</td><td>e</td><td></td><td>Turing Machine has (an infinite/unlimited amount of) memory / storage;<br>Turing Machine can read and write / input and output (data) to/from a tape;<br>Turing Machine has infinitely long tape; N.E. Turing Machine has a tape<br>MAX 1</td><td>1</td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| **2** | **a** | | Qualification(<u>SubjectName, Level</u>)<br>Module(<u>ModuleCode</u>, ModuleName, UMSPoints, SubjectName, Level)<br>Entry(<u>CentreNumber, CandidateNumber, ModuleCode, ExamSession</u>)<br><br>**1 mark** for naming all three relations correctly (Qualification, Module, Entry) – if alternative names are used, the purpose of the tables must be clear to award this mark.<br>**1 mark** for one relation containing the correct attributes OR **2 marks** for all three relations containing the correct attributes.<br>**1 mark** for correct primary key identified in one relation OR **2 marks** for correct primary key identified in all three relations.<br><br>**A.** Alternative names for relations and attributes, so long as meaning is clear.<br>**A.** Spaces in relation and attribute names.<br>**A.** Introduction of a QualificationID field into the Qualification relation with it being used as the primary key (and therefore link to the Module table as well)<br>**A.** Introduction of an EntryID field into the Entry relation with it being used as the primary key.<br>**A.** ExamSession attribute split into parts e.g. Season, Year or just as a Date.<br>**A.** Creation of a fourth relation for sessions, containing a SessionID as the primary key and the ExamSession as the other attribute then the use of the SessionID in the Entry relation instead of ExamSession.<br>**A.** Accept the addition of other attributes to relations that are not asked for, as long as they are relevant e.g. 'Extra Time' in the Entry relation or 'Entry Restrictions' to the Qualification relation **but** a relation should not be considered to have the correct attributes if additional attributes are added that are irrelevant or would mean that the design is not normalised.<br>**I**. If any unnecessary relations are created for the first mark on relation names.<br>**I.** Any representation for foreign keys.<br>**R.** Attribute names that are the same as relation names. | 5 |

| | | | | |
|---|---|---|---|---|
| **2** | **b** | | Composite (key);<br>**A.** Compound (key) | 1 |

| | | | | |
|---|---|---|---|---|
| **3** | **a** | | 192.168.0.x (where x is not 0, 2 or 255); | 1 |

| | | | | |
|---|---|---|---|---|
| **3** | **b** | | Star (topology);<br>**A.** Star network<br>**I.** additional writing that does not talk out the response | 1 |

| | | | | |
|---|---|---|---|---|
| **3** | **c** | | Devices are not directly connected to the Internet; **N.E.** all computers on a private network<br><br>So that LAN devices cannot be connected to (directly) by computers outside of the LAN / on the Internet // for increased security; **A.** relevant examples of increased security<br><br>Don't need to be allocated by a central authority // would be difficult to organise for each device to have a unique (routable) IP address // easier to allocate if do not need to be unique //( as devices not directly connected to Internet) IP addresses don't need to be globally unique; **N.E.** routable IP addresses globally unique and non-routable only locally unique<br><br>Would/May not be enough unique IP addresses for each device to have a routable address // globally more IP addresses would be required if all devices had routable IP addresses // more bits would be required to store an IP address if all devices had routable IP addresses;<br><br>**MAX 2** | 2 |

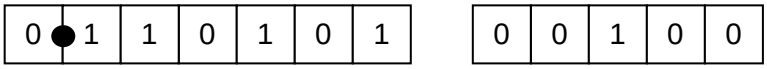
| | | | | |
|---|---|---|---|---|
| **3** | **d** | | AND operation performed using IP address(es) and subnet mask (to produce network IDs/subnet IDs of both desktop computer and FTP server) // Network IDs/subnet IDs/ first three octets/bytes/values (in IP addresses) computed using IP address(es) and subnet mask;<br><br>**To award either of the next two marks, the candidate must have indicated that the subnet mask is used to produce the results that will be compared - even if the method by which the subnet mask is used is incorrect so the first mark has not been awarded.**<br><br>Network/subnet IDs of both computers/machines compared;<br>**A.** Results of previous operation compared<br>**A.** First three octets/bytes/values (in IP addresses) compared<br>**A.** Award this mark by implication if it is stated what will happen if these two are the same or different<br><br>As network/subnet IDs (**A.** first thee octets/bytes/values/results) differ, desktop computer determines that FTP server is not on same network (so must be communicated with via combined device); | 3 |

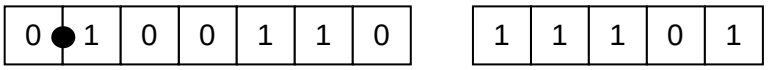
| | | | | |
|---|---|---|---|---|
| **3** | **e** | | Block/allow (traffic on) specific ports // block specified protocols;<br>Block/allow (traffic from) specific IP addresses/domain names;<br>Search packets for specific contents / text (and block/allow based on this);<br>Act as a proxy server // all traffic to Internet must go via firewall // stops computers on the Internet directly accessing devices on the LAN;<br>Stateful inspection // firewall maintains information about current connections and only allows packets relevant to these connections through;<br>Identifies unusual behaviour from a host // example of unusual behaviour eg sending an unusually large amount of data;<br>**N.E.** Packet filtering<br>**N.E.** "Data" instead of "packets"<br>**N.E.** Block specific programs connecting to Internet<br>**A.** Firewall checks packets using rules/criteria for 1 mark if not other marks awarded<br>**MAX 3** | 3 |

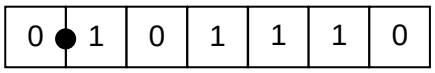
| | | | | |
|---|---|---|---|---|
| **3** | **f** | | *Baseband*<br>Whole bandwidth of medium dedicated to one transmission (at a time) // one channel (at a time) // only one computer can send data (at a time) // sends signals with frequencies from 0Hz to a maximum highest<br>frequency;<br>*Broadband*<br>Bandwidth of medium shared // multiple channels<br>can be carried (simultaneously) // many computers can send data (simultaneously) // frequency bands assigned to different communications; **TO** multiple wires<br>**MAX 1** | 1 |

| | | | | |
|---|---|---|---|---|
| **3** | **g** | | More reliable // less susceptible to interference // more stable connection;<br>Faster transmission speed // higher bit rate // lower latency;<br>**R.** More secure (not relevant in this instance)<br>**N.E.** Just the word "faster" on its own.<br>**MAX 1** | 1 |

| | | | | |
|---|---|---|---|---|
| **4** | **a** | | **1 mark** per correct answer:<br><br>**Value description** / **Correct letter (A-D)**<br>A negative value. / C;<br>The largest positive number of the four values. / B;<br>A value that is not valid in the representation because it is not normalised. / A;<br><br>If a letter is used more than once then mark as correct in the position that it is correct (if any). | 3 |

| | | | | |
|---|---|---|---|---|
| **4** | **b** | |  <br><br>Mantissa Exponent<br><br>**1 method mark** for either:<br>• showing correct value of both mantissa and exponent in denary (Mantissa = 0.828125 // 53/64, Exponent = 4)<br>• showing binary point shifted 4 places to right in binary number<br>• indicating that final answer calculated using answer = mantissa x $2^{exponent}$<br><br>**1 mark** for correct answer<br><br>Answer = 13 1/4 // 13.25 // 53/4<br><br>*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** | |  <br>Mantissa Exponent<br><br>**1 method mark** for either:<br>• showing correct value of both mantissa and exponent in denary (Mantissa = 0.59375 // 19/32, Exponent = -3)<br>• showing binary point shifted 3 places to left in binary number<br>• indicating that final answer calculated using answer = mantissa x $2^{exponent}$<br><br>**1 mark** for correct answer<br><br>Answer = 19/256, 0.07421875 **A.** rounded to at least 2dp<br><br>*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** | **d** | | **2 marks** for working:<br><br>Correct representation of 2944 in fixed point binary: 101110000000; **A.** leading 0s.<br>Showing the correct value of the exponent in denary (12) or binary (1100) // showing the binary point being shifted 12 places;<br>Showing the correct value of the mantissa in binary: 0.101110; **A.** leading and trailing 0s.<br><br>**MAX 2**<br><br>**1 mark** for correct mantissa and exponent together:<br>  <br>Mantissa<br>  <br>Exponent<br><br>*If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.*<br><br>*Working marks can be awarded for work seen in the final answer e.g. correct exponent.* | 3 |

| | | | | |
|---|---|---|---|---|
| **4** | **e** | i | 0.12 // 12.87-12.75 // 3/25<br>**R.** -0.12<br>**A.** Award **BOD** mark if correct method has been shown i.e. 12.87 - 12.75 but candidate has then made an error performing the subtraction operation | 1 |

| | | | | |
|---|---|---|---|---|
| **4** | **e** | ii | 0.9324%<br>**A.** 0.009324 // 0.0093 // 0.12÷12.87<br>**A.** 4/429<br>**A.** Follow-through of incorrect answer to part 4 (e) (i)<br>**A.** Award **BOD** mark if correct method has been shown but candidate has then made an error performing the division operation | 1 |

| | | | | |
|---|---|---|---|---|
| **4** | **e** | iii | Underflow;<br>**A.** Underflow error<br>**R.** Stack underflow | 1 |

| | | | | |
|---|---|---|---|---|
| **5** | **a** | i | Faster to lookup a record / search (in most circumstances); **A.** direct access to record<br>Faster to insert a record;<br>Faster to delete a record;<br>Time complexity is O(1).<br>**N.E.** Faster without an example of what is faster<br>**N.E.** Easier, simpler, more efficient instead of “faster”<br>**MAX 1** | 1 |

| | | | | |
|---|---|---|---|---|
| **5** | **a** | ii | More compact file // no "empty" records in file;<br>Producing a sorted list/processing the records in order is faster;<br>Do not have to design a hash function so easier to design/program // no need to deal with collisions;<br>**N.E.** "Easier" without reason<br>**N.E.** No collisions<br>**MAX 1** | 1 |

| | | | | |
|---|---|---|---|---|
| **5** | **b** | | 751 for both answers;<br>**N.E.** If just one answer is 751 | 1 |

| | | | | |
|---|---|---|---|---|
| **5** | **c** | | ***Effect (1 mark):***<br>The records for both cars would map to / be stored in the same location (in the file) // the record for the second car would overwrite the record for the first;<br>**A.** Memory location as BOD<br><br>***How dealt with (1 mark):***<br>Store one record/car in the next available location in the file // store a pointer (in each file location) that points to a list of records that have all collided at the file location // use a linked list from the location;<br>**A.** Idea that each storage location could store more than one record e.g. five records per location, if explained - must be clear that each location can store a fixed number of records, just storing the second record in the same location is not enough.<br>**A.** Example of what "next available" might be<br>**A.** Key is rehashed<br>**A.** Table for file<br>**A.** Use a hashing function that computes different values for the example registration numbers // use a hashing function that uses more characters from the registration number<br>**N.E** Use a better / different hashing function | 2 |

| | | | | |
|---|---|---|---|---|
| **6** | | | **SUBJECT MARKING POINTS:**<br><br>**Role of the operating system (1 point):**<br><br>To hide the complexities of the hardware from the user // provision of virtual machine;<br><br>**Tasks carried out by most operating systems (MAX 3 points):**<br><br>Processor scheduling // allocation of processors // process scheduling; **A.** processor management<br>Thread management;<br>Context switching;<br>Inter-process communication;<br>Memory management // allocation of memory/RAM;<br>Virtual memory provision // a brief explanation of what virtual memory is;<br>I/O management // allocation of I/O devices // management of/communication with (device) driver software ; **A.** examples of devices<br>File management // organisation of files on storage devices // loading and saving of files; **A.** examples of devices<br>Interrupt handling;<br>Power / battery management;<br>Provision of a user interface // allows user to interact with computer;<br>Provision of an Application Programming Interface / API (so that application software can call operating system routines);<br>**A.** management of system security<br>**A.** managing communications over a network<br>**A.** just names of tasks e.g. "memory management"<br>**A.** “storage management” for one of “memory management” or “file management”<br><br>**Additional functionality of a real time operating system (MAX 3 points):**<br><br>Must be able to deal with many events occurring simultaneously;<br>Must be able to deal with events that occur at unpredictable times;<br>Must produce output / perform processing within a specified / predictable / known time interval // Must produce output quickly enough to affect (the source of) the inputs; **A.** quickly enough for task, in a timely manner **N.E.** quickly, instantly, in a reasonable time<br>Must be fail-safe;<br>Must be able to quickly switch between threads / processes // quickly allocate memory // quickly handle interrupts;<br>Must support non-sequential application programs; | 7 |

**HOW TO AWARD MARKS:**

| Mark Bands and Description | |
|---|---|
| 7 | *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).*<br>*SUB* Candidate has made seven mark-worthy points and successfully covers all of the three topic areas (role, tasks, real time).<br>*QWC1* Text is legible.<br>*QWC2* There are few, if any, errors of spelling, punctuation and grammar. Meaning is clear.<br>*QWC3* The candidate has selected and used a form and style of writing appropriate to the purpose and has expressed ideas clearly and fluently.<br>*QWC4* Sentences (and paragraphs) follow on from one another clearly and coherently.<br>*QWC5* Appropriate specialist vocabulary has been used. |
| 4-6 | *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).*<br>*SUB* Candidate has made at least four mark-worthy points and successfully covers at least two of the three topic areas (role, tasks, real time).<br>*QWC1* Text is legible.<br>*QWC2* There may be occasional errors of spelling, punctuation and grammar. Meaning is clear.<br>*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.<br>*QWC4* The candidate has used well-linked sentences (and paragraphs).<br>*QWC5* Appropriate specialist vocabulary has been used. |
| 1-3 | *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).*<br>*SUB* Candidate has made a small number of relevant points but only successfully covers one or two of the three topic areas (role, tasks, real time).<br>*QWC1* Most of the text is legible.<br>*QWC2* There may be some errors of spelling, punctuation and grammar but it should still be possible to understand most of the response.<br>*QWC3* The candidate has used a form and style of writing which has many deficiencies. Ideas are |

| | | | | |
|---|---|---|---|---|
| | | | not always clearly expressed.<br>*QWC4* Sentences (and paragraphs) may not always be well-connected.<br>*QWC5* Specialist vocabulary has been used inappropriately or not at all.<br>0 Candidate has made no relevant points.<br><br>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.<br><br>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. | |

| | | | | |
|---|---|---|---|---|
| **7** | **a** | | **Description** / **Correct letter (A-D)**<br>A graph that is not connected. – B;<br>A graph that is a tree. – A;<br><br>If a letter has been used more than once then mark it as correct in the row that it is correct in, if any. | 2 |

| | | | | |
|---|---|---|---|---|
| **7** | **b** | | **Example 1**<br><br>| | **1** | **2** | **3** | **4** | **5** | **6** |<br>|---|---|---|---|---|---|---|<br>| **1** | 0 | 1 | 0 | 1 | 0 | 0 |<br>| **2** | 1 | 0 | 1 | 0 | 1 | 0 |<br>| **3** | 0 | 1 | 0 | 0 | 1 | 1 |<br>| **4** | 1 | 0 | 0 | 0 | 1 | 0 |<br>| **5** | 0 | 1 | 1 | 1 | 0 | 0 |<br>| **6** | 0 | 0 | 1 | 0 | 0 | 0 |<br><br>**Example 2**<br><br>| | **1** | **2** | **3** | **4** | **5** | **6** |<br>|---|---|---|---|---|---|---|<br>| **1** | 0 | 1 | 0 | 1 | 0 | 0 |<br>| **2** | | 0 | 1 | 0 | 1 | 0 |<br>| **3** | | | 0 | 0 | 1 | 1 |<br>| **4** | | | | 0 | 1 | 0 |<br>| **5** | | | | | 0 | 0 |<br>| **6** | | | | | | 0 |<br><br>**1 mark** for labelling matrix with indices running from 1 to 6 on both axis and filled only with 0s and 1s, or some other symbol to indicate presence/absence of edge. e.g. T/F (allow a third symbol along diagonal). Absence can be represented by an empty cell.<br>**1 mark** for correct values entered into matrix, as shown in either example above<br><br>In **Example 2**, the shaded portion can be in either half – some indication must be made that half of the matrix is not being used. This could just be leaving it blank, unless the candidate has also represented absence of an edge by leaving cells blank.<br><br>Allow use of a third symbol in the central diagonal to indicate it unused, as it would not make sense to use it in this example.<br><br>Accept column and row labels in any order so long as they correspond to the data i.e. do not have to be in sequence 1 to 6. | 2 |

| | | | | |
|---|---|---|---|---|
| **7** | **c** | | | 6 |

| | | | | | | | Discovered | | | | | | Parent | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **S** | **D** | **V** | **U** | **C** | **Queue** | **1** | **2** | **3** | **4** | **5** | **6** | **1** | **2** | **3** | **4** | **5** | **6** | **Found** | **Output** |
| | | | | | | F | F | F | F | F | F | | | | | | | | |
| **1** | **6** | | | | **1** | **T** | F | F | F | F | F | | | | | | | F | |
| 1 | 6 | **1** | **2** | | **2** | T | **T** | F | F | F | F | | **1** | | | | | F | |
| 1 | 6 | 1 | **4** | | **2 4** | T | T | F | **T** | F | F | | 1 | | **1** | | | F | |
| 1 | 6 | **2** | **1** | | **4** | T | T | F | T | F | F | | 1 | | 1 | | | F | |
| 1 | 6 | 2 | **3** | | **4 3** | T | T | **T** | T | F | F | | 1 | **2** | 1 | | | F | |
| 1 | 6 | 2 | **5** | | **4 3 5** | T | T | T | T | **T** | F | | 1 | 2 | 1 | **2** | | F | |
| 1 | 6 | **4** | **1** | | **3 5** | T | T | T | T | T | F | | 1 | 2 | 1 | 2 | | F | |
| 1 | 6 | 4 | **5** | | 3 5 | T | T | T | T | T | F | | 1 | 2 | 1 | 2 | | F | |
| 1 | 6 | **3** | **2** | | **5** | T | T | T | T | T | F | | 1 | 2 | 1 | 2 | | F | |
| 1 | 6 | 3 | **5** | | 5 | T | T | T | T | T | F | | 1 | 2 | 1 | 2 | | F | |
| 1 | 6 | 3 | **6** | | **5 6** | T | T | T | T | T | **T** | | 1 | 2 | 1 | 2 | **3** | T | |
| 1 | 6 | 3 | 6 | **6** | 5 6 | T | T | T | T | T | T | | 1 | 2 | 1 | 2 | 3 | T | 6 |
| 1 | 6 | 3 | 6 | **3** | 5 6 | T | T | T | T | T | T | | 1 | 2 | 1 | 2 | 3 | T | 3 |
| 1 | 6 | 3 | 6 | **2** | 5 6 | T | T | T | T | T | T | | 1 | 2 | 1 | 2 | 3 | T | 2 |
| 1 | 6 | 3 | 6 | **1** | 5 6 | T | T | T | T | T | T | | 1 | 2 | 1 | 2 | 3 | T | 1 |

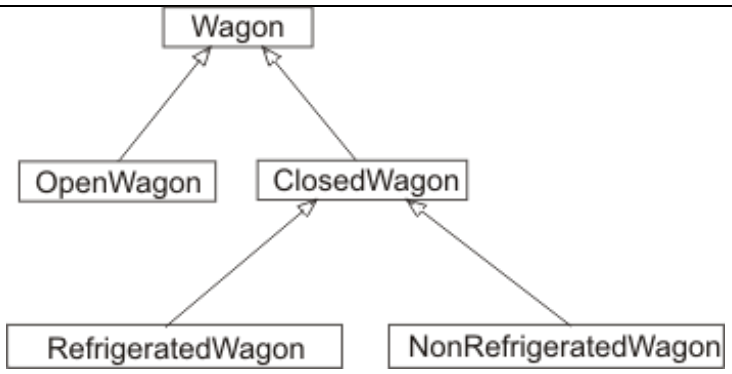
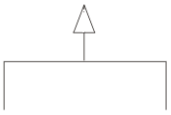
**1 mark** for having the correct value changes in each region highlighted by a rectangle and no incorrect changes in the region. Ignore the contents of any cells that are not changed.
**A.** Alternative indicators that clearly mean True and False.
**A.** It is not necessary to repeat values that are already set (shown lighter in table)
**A.** For the queue column, for cells that should only have one value in them, accept if the student has written out the value twice at both the Front and the Rear, so long as this has been done consistently throughout the table. eg "4" written as "4 4".

| | | | | |
|---|---|---|---|---|
| **7** | **d** | | So that packets/data arrive as quickly as possible/more quickly/with lower latency;<br>So that packets/data are delivered at lower cost/lowest cost;<br>So that routers do not have to process more packets/data due to unnecessary hops being made;<br>**N.E.** it is quicker without clarifying what "it" refers to<br>**MAX 1** | 1 |

| | | | | |
|---|---|---|---|---|
| **8** | **a** | | An abstraction / leaving out non-essential details // A representation of reality; | 1 |

| | | | | |
|---|---|---|---|---|
| **8** | **b** | | **1 mark for how stack works:**<br>Stack/It is a Last-in-First-Out / LIFO / First-in-Last-Out / FILO (data structure);<br><br>**1 mark for correspondence with siding (MAX 1):**<br>The last wagon to enter will be the first to leave;<br>Wagons enter and leave from same end of siding;<br>Wagons cannot leave siding before wagons that have entered after them;<br>Note: Responses must refer to both entering and leaving to gain this mark<br>**N.E.** References to "start", "end", "front", "back" of siding, without further clarification, as not clear which end of siding these terms refer to<br>**N.E.** A siding is LIFO – the student must refer to wagon in their answers, for example the last wagon to enter will be the first to leave. | 2 |

| | | | | |
|---|---|---|---|---|
| **8** | **c** | | `If TopOfStackPointer = 0`<br>`Then`<br>`  Stack Empty Error`<br>`Else`<br>`  CurrentWagon ← StackArray [TopOfStackPointer]`<br>`  Decrement TopOfStackPointer`<br>`EndIf`<br><br>**1 mark** for appropriate `If` structure including condition (does not need both `Then` and `Else`) – Do not award this mark if value is popped off stack outside of `If`.<br>**1 mark** for reporting error in correct place<br>**1 mark*** for decrementing `TopOfStackPointer`<br>**1 mark*** for transferring value from correct position in array into `CurrentWagon` variable<br><br>* = if the `CurrentWagon` assignment is performed after the decrement instruction OR the `If` structure then award **MAX 1** of these two marks UNLESS the item is removed from position `TopOfStackPointer+1` so the code would work.<br>**I.** unnecessary initialisation of any variables<br>**A.** `Stack Is Empty` for `TopOfStackPointer = 0`<br>**A.** Logic of `If` structure reversed i.e. `If stack is not empty / TopOfStackPointer>0` / `<>0` / `!=0` and `Then`, `Else` swapped<br>**A.** Any type of brackets or reasonable notation for the array index<br>**A.** Award the mark for dealing with the error situation even if the condition in the IF statement is not correct, as long as the purpose of the condition is clearly correct<br>**A.** Dealing with error in another sensible way eg by setting `CurrentWagon` to `Null`<br>**A.** Additional lines of code that do not affect behaviour but MAX 3 if these lines of code would stop the algorithm working correctly<br>**DPT** If candidate has used a different name for any variable then do not award first mark but award subsequent marks as if correct name used. | 4 |

| | | | | |
|---|---|---|---|---|
| **8** | **d** | |  <br>**1 mark** for Wagon at top of diagram with OpenWagon and ClosedWagon directly underneath it and linked to it and no other labels linked to it;<br>**1 mark** for ClosedWagon with RefrigeratedWagon and NonRefrigeratedWagon directly underneath it and linked to it, and no other labels linked to it (except Wagon above);<br>**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.***<br><br>**A.** Arrows drawn as:<br>  <br><br>**A.** Filled/empty arrowheads<br>**A.** Diagram rotated by 90 degrees | 3 |

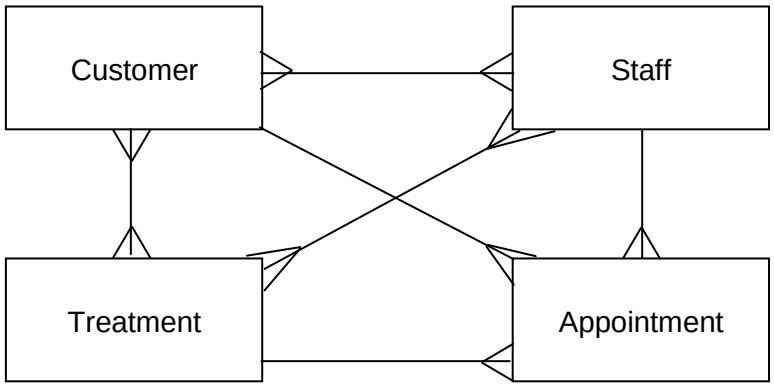
| | | | | |
|---|---|---|---|---|
| **8** | **e** | | `ClosedWagon = Class/Subclass/Extends Wagon` 1<br>`(Public)`<br>`Procedure CreateWagon (Override)` 1<br>`Function GetHeight`<br>`Function GetNumberOfDoors` } 1<br>`Function GetSuitableForFoodStuffs`<br>`Private / Protected`<br>`Height : Real`<br>`NumberOfDoors : Integer` } 1<br>`SuitableForFoodstuffs : Boolean`<br>`End`<br><br>Accept answers that use different notations, so long as meaning is clear.<br><br>**1 mark** for correct header including name of class and parent class;<br>**1 mark** for redefining the `CreateWagon` procedure;<br>**1 mark*** for defining all 3 extra functions needed to read variable values, all identified as being public (keyword `public` is optional if functions are declared before variables);<br>**1 mark**# for defining all 3 extra variables, with appropriate data types and identified as being private;<br><br>**A.** Any sensible numeric types for `Height` and `NumberOfDoors`. `Height` must accept non-integer values and `NumberOfDoors` integer values only.<br>**A.** Answers that indicate separately that each variable is private or each method is public<br>**R.** Do not award mark for declaring new functions if any of the functions have the same name as the variables<br>**I.** Parameters to methods, minor changes to names that do not affect clarity<br>* - Do not award this mark if any extra functions/procedures have been declared, EXCEPT for functions that would set values individually e.g. `SetHeight` or an incorrectly named procedure to add e.g. `CreateClosedWagon` which are acceptable for this mark<br># - Do not award this mark if any extra variables have been declared | 4 |

| | | | | |
|---|---|---|---|---|
| **9** | **a** | | (Using an algorithm) to convert a message into a form that is not understandable (without the key to decrypt it);<br>(Using an algorithm) to convert a message into a form that is only understandable by the intended parties // can only be read with the correct key;<br>(Using an algorithm) to convert a message into cipher text;<br>**N.E.** Scrambling unless further explanation is provided<br>**N.E**. Coding<br>**A.** "Unreadable" for "understandable"<br>**A.** "Data" for "a message"<br>**R.** Responses that do not make clear that encryption is a process<br>**MAX 1** | 1 |

| | | | | |
|---|---|---|---|---|
| **9** | **b** | | **1 mark** for two or three keys correctly named.<br>**2 marks** for all four keys correctly named.<br><br>**Label** / **Key Name**<br>❶ A's Private Key<br>❷ B's Public Key<br>❸ B's Private Key<br>❹ A's Public Key<br><br>**A.** "Sender" for "A" and "Recipient" for "B" (or similar role descriptions)<br>Allow use of same key name more than once and mark correct in the position it is correct (if any). | 2 |

| | | | | |
|---|---|---|---|---|
| **9** | **c** | | Two (message) digests are compared // received and recalculated digests compared;<br>**A.** "They" for the two message digests<br>**A.** "Hash" for "digest"<br>**R.** Two messages are compared | 1 |

| | | | | |
|---|---|---|---|---|
| **9** | **d** | | To authenticate/confirm identity of sender // to confirm that message was sent by A;<br>**A.** Ensures person is who they say they are<br>**N.E.** Identify the sender (must be clear that the signature confirms this identity), know who the sender is<br><br>To detect if message has been tampered with/changed;<br>**N.E.** Prevent the message being tampered with<br><br>**Award marks in part (d) for valid responses to part (d) that are made in part (c).** | 2 |

| | | | | |
|---|---|---|---|---|
| **10** | **a** | | **1 mark** for any one correct relationship drawn<br>**2 marks** for three correct relationships drawn<br>  <br><br>**MAX 1 if any incorrect relationships drawn** | 2 |

| | | | | |
|---|---|---|---|---|
| **10** | **b** | | `TreatmentName VARCHAR(20) PRIMARY KEY`<br>//<br>`TreatmentName VARCHAR(20)`<br>`PRIMARY KEY(TreatmentName)`<br><br>`Price SMALLMONEY`<br>`TimeTaken INT`<br>`NeedsQualification BOOLEAN`<br><br>**1 mark** for `TreatmentName`, with sensible data type and identified as primary key<br><br>**1 mark** for two other fields with sensible data types and lengths (if given) *OR* **2 marks** for all three other fields with sensible data types and lengths<br><br>**A.** Any sensible types. Lengths do not need to be specified.<br><br>**Valid alternative SQL types are:**<br><br>• Alternative types for `TreatmentName`: char, nchar, nvarchar, ntext, longvarchar, varchar2, nvarchar2, text, tinytext, mediumtext, longtext<br><br>• Alternative types for `Price`: money, float, real, decimal, double, numeric, currency **R.** integer only types for Price<br><br>• Alternative types for `TimeTaken`: tinyint, smallint, mediumint, integer, number, byte, time, date/time<br><br>• Alternative types for `NeedsQualification`: yes/no, bit, byte, bool, tinyint, enum("yes","no") – allow sensible alternative values in enum.<br><br>There should be a comma between the creation of each field, but ignore if these are missing, and accept a semi-colon at the end of the whole query.<br><br>**Answers using a syntax that is clearly not SQL should be awarded zero marks. But:**<br><br>• **ignore one punctuation error e.g. unnecessary colons between fieldname and type**<br><br>• **answers in SQL style syntax but using non-SQL data types can be credited but MAX 1 of 2 for other fields if any non-SQL types used.** | 3 |

| | | | | |
|---|---|---|---|---|
| **10** | **c** | | **Alternative 1**<br><br>`SELECT EmailAddress, Forename, Surname`<br>`FROM Customer, Appointment`<br>`WHERE TreatmentName = "Luxury Manicure"`<br>`  AND ApDate >= "01/01/2014"`<br>`  AND ApDate <= "31/12/2014"`<br>`  AND Customer.CustomerID =`<br>`  Appointment.CustomerID`<br><br>**1 mark** for `SELECT` clause with correct three fields<br>**1 mark** for `FROM` clause with correct two tables<br>**1 mark** for `TreatmentName = "Luxury Manicure"`<br>**1 mark** for `ApDate >= "01/01/2014"`<br>**1 mark** for `ApDate <= "31/12/2014"`<br>**1 mark** for `Customer.CustomerID =`<br>`Appointment.CustomerID`<br><br>**Alternative 2**<br><br>`SELECT EmailAddress, Forename, Surname`<br>`FROM Customer INNER JOIN Appointment ON`<br>`  Customer.CustomerID = Appointment.CustomerID`<br>`WHERE TreatmentName = "Luxury Manicure"`<br>`  AND ApDate >= "01/01/2014"`<br>`  AND ApDate <= "31/12/2014"`<br><br>**1 mark** for `SELECT` clause with correct three fields<br>**1 mark** for `FROM` clause with correct two tables<br>**1 mark** for using `INNER JOIN` with condition<br>`Customer.CustomerID = Appointment.CustomerID`<br>**1 mark** for `TreatmentName = "Luxury Manicure"`<br>**1 mark** for `ApDate >= "01/01/2014"`<br>**1 mark** for `ApDate <= "31/12/2014"`<br><br>**In Either Alternative**<br><br>Marks for correct fields/tables in `SELECT` and `FROM` statements should not be awarded if additional fields/tables included, except allow the inclusion of the `TREATMENT` table in the `FROM` statement so long as it has been correctly linked to the `APPOINTMENT` table either in the `FROM` or the `WHERE` clause.<br>Marks can be awarded for the conditions in the `WHERE` statement even if the required tables are not present in the `FROM`.<br>If `WHERE` is missing/incorrect, do not award marks for the first correct condition, but award marks for subsequent correct conditions<br>If `AND` operators are missing and there are multiple conditions, do not award marks for the first correct condition, but award | 6 |

| | | | | |
|---|---|---|---|---|
| | | | marks for subsequent correct conditions<br>Accept table names before fieldnames.<br>Accept use of `Alias`/`AS` command e.g. `FROM Customer AS C` then use of `C` as table name (note some dialects of SQL do not require `AS` e.g. `FROM Customer C`)<br>Accept insertion of spaces into fieldnames.<br>Accept `ApDate > "31/12/2013", ApDate < "01/01/2015"` as alternative date criteria.<br>Accept dates written in any format eg `"12-31-2013"`<br>Valid delimiters for dates are `"`, `'` or `#`<br>Valid delimiters for strings are `"` or `'`<br>Valid symbols between date parts are `/`, `-` or no symbol<br>Ignore unnecessary clause<br>`Appointment.TreatmentName = Treatment.TreatmentName`<br>Ignore unnecessary brackets.<br>Accept the following alternative methods for specifying the year, each of which are worth **2 marks**:<br>`YEAR(ApDate) = 2014`,<br>`DATEPART("yyyy",ApDate) = 2014` or no quotation marks<br>`DATEPART("yy",ApDate) = 2014` or no quotation marks<br>`DATEPART("Year",ApDate) = 2014` or no quotation marks<br>`LIKE(ApDate, "*2014*")` or `"*2014"`<br>`LIKE(ApDate, "%2014%")` or `"%2014"`<br>`LIKE "*/*/2014"`<br>`LIKE "*2014"`<br>`BETWEEN "01/01/2014" AND "31/12/2014"` or allow `"01/01/2015"` as upper limit<br>**DPT** for unnecessary punctuation – allow one semicolon at the very end of the statement, but not at the end of each clause.<br>**DPT** missing delimiters around data values, eg no quotation marks around dates.<br>**DPT** for fieldname before table name.<br><br>**Refer responses using nested SQL queries to team leaders.** | |

| | | | | |
|---|---|---|---|---|
| **11** | **a** | | The problem can be solved // algorithm exists for problem;<br>in polynomial time (or less) // in a reasonable amount of time; | 2 |

| | | | | |
|---|---|---|---|---|
| **11** | **b** | | Use of heuristic;<br>An algorithm that makes a guess/estimate based on experience;<br>**N.E.** just algorithm that uses previous knowledge/experience<br>That provides a close-to-optimal solution/approximation // that only works in some cases; **A.** non-optimal<br>Example of heuristic method e.g. hill-climbing/stochastic/local improvement/greedy algorithms/simulated annealing/trial and error/any reasonable example;<br>Relax some of the constraints on the solution; **A.** solve simpler version of problem **A.** limit size of input<br>**MAX 2** | 2 |

| | | | | |
|---|---|---|---|---|
| **11** | **c** | | <table><tr><th>Problem</th><th>Unsolvable?<br>(Tick one row)</th></tr><tr><td>The problem of sorting a list into order.</td><td></td></tr><tr><td>The Halting problem.</td><td>✓;</td></tr><tr><td>The travelling salesman problem.</td><td></td></tr></table><br>**R.** Responses in which more than one row is ticked<br>**A.** Responses in which a symbol other than a tick is used, so long as it is only placed on one row.<br>**A.** Use of two symbols, with one indicating which problem is unsolvable and the other indicating which two are solvable, so long as the meaning of the symbols is clear. | 1 |

| | | | | |
|---|---|---|---|---|
| **11** | **d** | | <table><tr><th>Order of Time Complexity</th><th>Least Efficient?<br>(Tick one row)</th></tr><tr><td>$O(2^n)$</td><td>✓;</td></tr><tr><td>O(n)</td><td></td></tr><tr><td>$O(n^2)$</td><td></td></tr></table><br>**R.** Responses in which more than one row is ticked<br>**A.** Responses in which a symbol other than a tick is used, so long as it is only placed on one row.<br>**A.** Use of two symbols, with one indicating which algorithm is least efficient and the other indicating which two are more efficient, so long as the meaning of the symbols is clear. | 1 |

| | | | | |
|---|---|---|---|---|
| **12** | **a** | | <table><tr><th><variable></th><th>Valid?<br>(Tick any number of rows)</th></tr><tr><td>a</td><td>✓</td></tr><tr><td>money-paid</td><td></td></tr><tr><td>taxrate2</td><td>✓</td></tr><tr><td>2ndPlayerName</td><td></td></tr></table><br>**1 mark** for ticks in the correct two rows and other rows left blank.<br><br>**A.** Use of alternative symbol for tick<br>**A.** Use of two symbols - one to indicate validity and one to indicate invalidity, so long as the meaning of the symbols is clear e.g. a tick and a cross or a Y and an N. | 1 |

| | | | | |
|---|---|---|---|---|
| **12** | **b** | | Required as an integer can contain any number of digits;<br>**N.E.** More than one digit<br>**A.** "numbers" for "digits" as **BOD**<br>BNF does not support iteration/looping // BNF can only achieve iteration through recursion // would need infinite number of rules otherwise;<br>**N.E.** Rule needs to loop<br>**MAX 1** | 1 |

| | | | | |
|---|---|---|---|---|
| **12** | **c** | | Variable may not have been declared;<br>Variable may be of inappropriate type;<br>Position of statement within program may be invalid;<br>Rightmost integer may be a lower value than the leftmost one;<br>One of the numbers/limits may be outside of the range of valid integers;<br>**A.** Examples of any of the above<br>**MAX 1** | 1 |