A-LEVEL
COMPUTER SCIENCE
7517/2

Paper 2

Mark scheme
June 2019

Version: 1.0 Final

196A75172/MS

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 Examiner.

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

Level of response marking instructions

Level of response mark schemes are broken down into levels, each of which has a descriptor. The descriptor for the level shows the average performance for the level. There are marks in each level.

Before you apply the mark scheme to a student's answer read through the answer and annotate it (as instructed) to show the qualities that are being looked for. You can then apply the mark scheme.

Step 1 Determine a level

Start at the lowest level of the mark scheme and use it as a ladder to see whether the answer meets the descriptor for that level. The descriptor for the level indicates the different qualities that might be seen in the student's answer for that level. If it meets the lowest level then go to the next one and decide if it meets this level, and so on, until you have a match between the level descriptor and the answer. With practice and familiarity you will find that for better answers you will be able to quickly skip through the lower levels of the mark scheme.

When assigning a level you should look at the overall quality of the answer and not look to pick holes in small and specific parts of the answer where the student has not performed quite as well as the rest. If the answer covers different aspects of different levels of the mark scheme you should use a best fit approach for defining the level and then use the variability of the response to help decide the mark within the level, ie if the response is predominantly level 3 with a small amount of level 4 material it would be placed in level 3 but be awarded a mark near the top of the level because of the level 4 content.

Step 2 Determine a mark

Once you have assigned a level you need to decide on the mark. The descriptors on how to allocate marks can help with this. The exemplar materials used during standardisation will help. There will be an answer in the standardising materials which will correspond with each level of the mark scheme. This answer will have been awarded a mark by the Lead Examiner. You can compare the student's answer with the example to determine if it is the same standard, better or worse than the example. You can then use this to allocate a mark for the answer based on the Lead Examiner's mark on the example.

You may well need to read back through the answer as you apply the mark scheme to clarify points and assure yourself that the level and the mark are appropriate.

Indicative content in the mark scheme is provided as a guide for examiners. It is not intended to be exhaustive and you must credit other valid points. Students do not have to cover all of the points mentioned in the Indicative content to reach the highest level of the mark scheme.

An answer which contains nothing of relevance to the question must be awarded no marks.

A-level Computer Science

Paper 2

June 2019

To Examiners:

- **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 credit worthy.

  Insert a hyphen when a candidate has not attempted a question, so that eventually the Principal Examiner will be able to distinguish between the two (not attempted / nothing credit worthy) in any statistics.

- 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 responses that are not covered by the mark scheme, but which they deem 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 - 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.
TO - talked out

Examiners are required to assign each of the candidates' responses to the most appropriate level according to **its overall quality**, then allocate a single mark within the level. When deciding upon a mark in a level examiners should bear in mind the relative weightings of the assessment objectives.

eg

In the following questions, the marks available are as follows:

Question 2.1 (max 4 marks)
AO1 (knowledge) – 2 marks
AO2 (analyse) – 2 marks

Question 7.3 (max 3 marks)
AO1 (understanding) – 2 marks
AO2 (analyse) – 1 mark

Question 8.3 (max 2 marks)
AO1 (understanding) – 1 mark
AO2 (analyse) – 1 mark

Question 10 (max 8 marks)
AO2 (analyse) – 2 marks
AO3 (programming) – 6 marks

Question 11.4 (max 3 marks)
AO1 (understanding) – 2 marks
AO2 (analyse) – 1 mark

Q		Marks	
01	1	**Mark is AO1 (knowledge)** **1 mark:** Lozenge for "Utilities" shaded. **R.** if more than one lozenge shaded	**1**
01	2	**All marks AO1 (knowledge)** Allocate processors/cores to processes // schedule processes // decide which process to carry out when; Allocate memory/RAM to processes // moving data into and out of RAM / to a paging file for virtual memory // ensuring processes can only write to memory that they have been allocated; Allocate I/O devices to processes // manages communication between processes and I/O devices // automatic installation of drivers for new I/O devices; **A.** examples of devices (but no more than one mark) **NE.** manages I/O devices Allocate space on a storage device to files // organising files into directories // determines where on a device to save a file // recognising storage devices when they are connected; **A.** defragmentation of disks **NE.** saving a file Installation of new software // automatic/managing updating of software; **A.** "programs" or "tasks" for "processes" **R.** handling interrupts **R.** hides complexity **Note:** Students must describe the type of resource management – phrases such as "processor management", "allocating memory" etc are not enough. **Max 2**	**2**

02	**1**	**2 marks AO2 (analyse) and 2 marks AO1 (knowledge)**
		
		Award **1 mark** for each correctly named protocol, up to a maximum of **2 marks** and **1 mark** for each correct explanation of what a protocol will be used for, if linked to the correct protocol.
		
		Protocol (AO2) / **Use (AO1)**
		SMTP // Simple Mail Transfer Protocol — To send/transmit/receive emails (to/from another email server/client).
		POP(3) // Post Office Protocol (3) // IMAP // Internet Message Access Protocol — (So that clients can) retrieve/manage emails on the server. **TO.** sending emails **A.** receiving emails as BOD but **TO.** receiving emails if answer suggests that this is done as the email is sent.
		SSH // Secure Shell // Telnet // RDP // Remote Desktop Protocol — So that technicians can execute commands on the server // to give access to command line // provides a secure/encrypted connection for remote management (only award for secure protocols) **NE.** to login remotely
		HTTP / HTTPS — So users can access email via the web / a web browser // so that technicians can access web-based control panels.
		
		A. if correct initialism used but then the full term is incorrect eg "SMTP – Special Mail Transfer Protocol" as **BOD**
		A. other protocols that achieve the same purposes as those listed above
		R. non-application layer protocols eg TCP, IP
			4

02	**2**	**Mark is AO1 (understanding)** (The transport layer will) use the port number to (determine which server / software should deal with the request) // by adding a port number to the request/data/packet; **A.** examples of specific port numbers and which server / software they would be directed to.	**1**

02	**3**	**Mark is AO1 (knowledge)** Adds source IP / destination IP address(es) (to datagrams/packets); **R.** MAC address **NE.** Adds IP address **NE.** Uses destination IP address Performs routing // selects the next host / hop to transmit a datagram/packet to; **A.** determines where to send data to using destination IP address **NE.** determines where to send data to Creates checksum for datagram/packet header // performs error detection on the datagram/packet header; **NE.** error detection on its own Encapsulating/splitting data into datagrams // reassembling data from datagrams; **R.** packets for this mark point only **Max 1**	**1**
03	**1**	**Mark is AO1 (understanding)** The original data can be fully recovered if lossless compression has been used // lossless data compression can be reversed; **NE.** no data is lost **NE.** no loss of quality The original data cannot be recovered if lossy compression has been used // lossy compression cannot be reversed // the data is degraded by lossy compression; **A.** redundant / less important data removed **NE.** data is lost **NE.** quality is reduced **Max 1**	**1**
03	**2**	**All marks AO1 (understanding)** A dictionary is built that maps sequences of characters/substrings/words/strings in the text onto tokens/values/numbers; **A.** sequences of characters/substrings/words/strings are stored at known positions in a list/table/array **TO.** sequences of characters/substrings/words/strings and their frequencies/positions (in the paragraph) are stored The (sequences of) characters/substrings/words/strings in the text are then replaced by the corresponding tokens/values/numbers/indices in the dictionary; **A.** shown by example **If no other marks awarded, award one mark if stated that sequences of characters/substrings/words/strings are assigned tokens/values/numbers, regardless of whether it is clear if this means in the dictionary or paragraph of text.**	**2**

03	**3**	**All marks AO1 (understanding)** For small pieces of text there is little repetition (and so the compressed text will be similar in size to the original); **A.** The dictionary itself will require storage space // will need to be transmitted; **Max 1**	**1**
04		**Mark is AO2 (analysis)** In column C Result should be 1 / is wrong // column C should be 1 carry 1 /11 // the carry has not been included when adding up the values in column C; **A.** column C should be 1 not 0 **NE.** column C is wrong **NE.** column C should be 1	**1**

05 **All marks AO1 (understanding)** **12**

Level	Description	Mark Range
4	A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response. The response covers all three areas indicated in the guidance below and in at least two of these areas there is sufficient detail to show that the student has a good level of understanding. To reach the top of this mark range, a good level of understanding must be shown of all three areas.	10-12
3	A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response which shows a good level of understanding of at least two areas indicated in the guidance below.	7-9
2	A limited attempt has been made to follow a line of reasoning and the response has a mostly logical structure. Either a good level of understanding of one area from the guidance has been shown or a limited understanding of two areas.	4-6
1	A few relevant points have been made but there is no evidence that a line of reasoning has been followed. The points may only relate to one or two of the areas from the guidance or may be made in a superficial way with little substantiation.	1-3

Guidance – Indicative Response

1. Why translation is necessary

Processor can only execute machine code instructions **A.** computer for processor

High-level instructions cannot be executed directly // high-level instructions are not machine code **A.** must be converted to machine code to be executed

NE. "Understand" for "execute".

Good level of understanding = at least one point made

2. Differences between compilation and interpreting

Compiler analyses program as a whole

Interpreter analyses program on a line-by-line basis

Compiler produces object code/executable file/machine code/bytecode

Interpreter calls subroutines within its own code to carry out commands

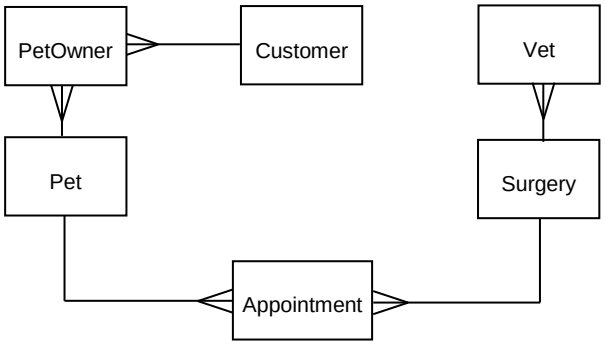
Compiler will not translate any of the program if it encounters an error

Interpreter translates/executes program until first error is encountered

If (unchanged) program executed twice/multiple times, compiler will only need to translate it once

Interpreter translates a program each time it is executed

		Interpreter executes each line immediately after translating it If student has written about compiler outputting bytecode then: bytecode will later be interpreted // executed by a virtual machine // just-in-time-compiled Once translated, compiled code does not need the compiler to be present to run An interpreter must always be present for a program that is interpreted to run Once compiled, code will only run on one type of processor / virtual machine Interpreter could translate the same instruction multiple times (eg if it is in a loop) *Good level of understanding = at least four points made* **3. How machine code instructions fetched and executed** <u>F-E Stage 1 Fetch:</u> Contents of Program Counter / PC transferred to Memory Address Register / MAR **R.** if implied the instruction is stored in the PC Address bus used to transfer this address to main memory Transfer of main memory content uses the data bus Contents of addressed memory location loaded into the Memory Buffer Register / MBR Increment (contents of) Program Counter / PC **A**. at any part of fetch process after transferring PC to MAR Increment Program Counter / PC and fetch instruction simultaneously Contents of MBR copied to CIR <u>F-E Stage 2 Decode:</u> Decode instruction held by the (Current) Instruction Register / (C)IR The control unit decodes the instruction Instruction split into opcode and operand <u>F-E Stage 3 Execute:</u> If necessary, data is fetched If necessary, data is stored in memory The opcode identifies the type of operation/instruction to be performed (by the processor) Result (may be) stored in register/accumulator The operation (identified by the opcode) is performed by the processor. **A.** ALU Status register updated If jump / branch required Program Counter/PC is updated <u>General:</u> Instructions will be for program (if compiled) or interpreter (if interpreted) *Good level of understanding = at least five points made and at least two of the three stages of the F-E cycle are covered.*	

06	**1**	**All marks AO2 (analysis)**  **1 mark:** Entity Appointment added and correct relationships and degree drawn to Pet and Surgery entities. **1 mark:** Entity Customer added and correct relationship and degree drawn to PetOwner. **Note:** If PetOwner relation not created then allow this mark if a many-to-many relationship is drawn between Pet and Customer, even though this is not fully normalised. **1 mark:** Entity PetOwner added and correct relationship and degree drawn to Pet. **A.** entity names do not have to match diagrams exactly but must convey same purpose. **Ignore the inclusion of any additional entities and the drawing of any other relationships, whether they are correct or not.**	**3**

06	**2**	**All marks AO2 (analysis)** Customer(<u>CustomerID</u>, Forename, Surname, TelephoneNumber) Appointment(<u>PetID, Date, Time,</u> SurgeryName) PetOwner(<u>CustomerID, PetID</u>) **1 mark:** Customer relation created and contains the correct attributes and CustomerID identified as the entity identifier. **I.** any additional reasonable attributes, including PetID if the PetOwner relation has not been created, **but** reject PetID included if the PetOwner entity has been created. If PetID is included then it is acceptable for it to be in or not in the entity identifier. **1 mark:** Appointment relation created and contains the correct attributes. **I.** any additional reasonable attributes including CustomerID, VetID **A.** date and Time given as one combined field. **1 mark:** Composite entity identifier of PetID, Date and Time identified for Appointment relation. **A.** creation of a new attribute to be the entity identifier eg AppointmentID. **A.** date and Time given as one combined field. **1 mark:** PetOwner relation created and contains the correct attributes and no others. Additionally, either: • both attributes identified as a composite entity identifier, or, • **A.** a new entity identifier eg OwnershipID created. **R.** Just one of PetID or CustomerID given as entity identifier. **For all mark points** **A.** alternative names for relations and attributes created by candidate, so long as meaning is clear. **R.** use of incorrect attribute names for attributes already named in question paper. **A.** spaces in relation and attribute names. **I.** if any unnecessary relations are created. **I.** any representation for foreign keys. **Accept responses written in SQL – ignore syntactical errors and data type errors in such responses.**	**4**
06	**3**	**All marks AO2 (analysis)** Torquay / the town name is missing quotations marks // needs quotation marks around it; There is no linking condition/join between the two tables using the SurgeryName // a join needs to be added to the `FROM` clause using the SurgeryName // a linking condition needs to be added to the `WHERE` clause using the SurgeryName // the condition `Surgery.SurgeryName = Vet.SurgeryName` is missing / needs adding; **NE.** the tables have not been linked.	**2**

06	**4**	**All marks AO1 (understanding)** Two users (read and) edit a record/the same data simultaneously; **NE.** access/read unless later made clear the record/data is changed **NE.** edit the database simultaneously One user writes the record/data back/saves then the other user writes the record/data back/saves; One user's update is lost // only one user's update is the kept; **NE.** data is lost **A.** examples that map to the above points. **If no other marks awarded, award one mark for the use of the term "lost update problem".** **Refer examples relating to data being read whilst another transaction that is later rolled-back is in progress to team leaders.**	**3**
06	**5**	**All marks AO1 (knowledge)** **Mark against "Record locks" or "Timestamp ordering" mark scheme, depending upon which method the student has selected.** **Record locks:** When a transaction on a record starts / when a user starts to edit a record an (exclusive) lock is set on the record; **R.** database/data/file/table for record Other transactions/users cannot edit (**A.** access) the record/data until the lock is released/while the lock is in place/until the first edit is completed; **Timestamp ordering:** Timestamps are generated for each transaction // timestamps indicate the order that transactions occurred in; **A.** timestamps generated for edits/queries as BOD Database records timestamp of last read / last write transaction for each record / data item; **A.** just one of read/write Database server applies rules to determine if processing a transaction will result in loss of data integrity/inconsistency (and if so aborts the transaction); **A.** Examples of rules for this mark point: • If a transaction tries to write to a record/data item then the transaction should be aborted if the read/write timestamp on the record/data item is greater that the time at which the transaction started. • If a transaction tries to read a record/data item then the transaction should be aborted if the write timestamp on the record/data item is greater that the time at which the transaction started. **Max 2**	**2**

07	**1**	**Mark is AO2 (analysis)** (The set of) integer (numbers) // ℤ // Z // the same set as the domain;	**1**
07	**2**	**Mark is AO2 (apply)** 4;	**1**
07	**3**	**2 marks AO1 (understanding) and 1 mark AO2 (analysis)** **2 AO1 (understanding) marks for describing how partial application works:** The function/add is applied to one of its arguments/4/6; **A.** one of the arguments is fixed/bound **A.** the new function takes one less argument // the new function takes one argument The output of this function application is a new function; **A.** a new function is created **1 AO2 (analysis) mark for working out what the output of the partial application would be in this instance:** The new function always adds the argument that the original function was applied to/4/6 to one argument; **A.** award this mark if it is clear what the new function does from its name eg "add4". **Note:** If a specific value (4 or 6) is used this must be the same value (4 or 6) for which the earlier mark point was awarded, if it was. Accept answers given by example, mapping the relevant part of the example to the mark scheme points. **Example of an answer by example** add6(x) **2 AO1 marks** for: • output of the partial application is a new function (eg add6) • the new function only has one argument(eg x) where add6(x) = 6 + x **1 AO2 mark** for explaining what new function does. **Max 2** for an answer by example if there is no description. *Answers can be a mix of marks for description and example.*	**3**

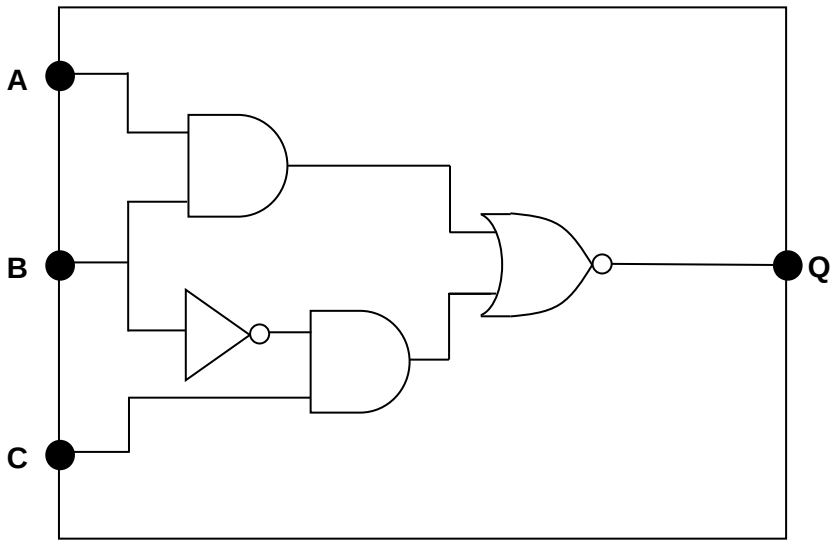
08	**1**	**Mark is AO1 (knowledge)** **OR Gate** **NAND Gate** **1 mark:** All values in both **Output** columns are correctly completed.	**1**

OR Gate

Inputs		Output
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

NAND Gate

Inputs		Output
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

08	**2**	**All marks AO2 (apply)**  **1 mark:** Circuit inputs A and B connected as the inputs to an AND gate. **1 mark:** Circuit input B connected as the input to a NOT gate. **A.** NOT gate drawn as triangle without circle as BOD **1 mark:** Output of a NOT gate and C connected as the inputs to an AND gate. **A.** B and C going into AND gate if second mark point not awarded. **1 mark:** Outputs of two AND gates connected to a NOR gate as inputs which has its output connected to Q. **A.** correct use of OR and NOT gate instead of NOR gate. **Max 3** if circuit logic not fully correct **<u>Logically Equivalent Expressions</u>** If a response includes a statement of a logically equivalent expression for example $\overline{A\,.\,B}\,.\,\overline{C\,.\,\overline{B}}$ or $(\overline{A}+\overline{B}).(\overline{C}+B)$ or $\overline{\overline{\overline{A}+\overline{B}}+\overline{\overline{C}+B}}$ then: • If the student appears to have drawn a circuit for the expression given to them on the question paper mark against mark points above. • If the student appears to have drawn a circuit for their logically equivalent expression refer the response to a team leader for marking. If a response includes a statement of an expression that is **not** logically equivalent then mark against mark points above.	**4**
08	**3**	**All marks AO1 (understanding)** If input A is 0 then NOT A will be 1 and if A is 1 then NOT A will be 0 // one of the inputs to the AND operator will always be 0 // the inputs can only be 0,1 or 1,0; **NE.** if only expressed one way around eg if A is 0 then NOT A is 1 **NE.** NOT A is always the opposite of A unless clarified that possible values are 0/1 **NE.** if only presented as a truth table **A.** on/off, true/false for 1/0 An <u>AND</u> gate only outputs 1 if both inputs are 1 // an <u>AND</u> gate always outputs 0 if one of its inputs is 0 // when inputs to <u>AND</u> are 1 and 0 then output is 0;	**2**

08	**4**	**All marks AO2 (apply)** **Marking guidance for examiners** • Award marks for working out until an incorrect step has been made. • If, in any one step, a candidate is simplifying different parts of an expression simultaneously award all relevant marks for this multiple stage but don't award any further marks for working in any parts simplified incorrectly. For example, if the expression P.P.(P+Q) + P.P.1 was changed to P.(P+Q)+P.0, the candidate would get one mark for simplifying the first part to P.(P+Q) and could get further marks for correctly simplifying this part of the expression further but should not be awarded marks for simplifying the incorrectly changed part P.0 (ie to 0) **1 mark** for final answer: A + B **Max 3** for working. Award up to two marks for applying each one of the three techniques (one mark per application): • a successful application of De Morgan's Law (and any associated cancellation of NOTs) that produces a simpler expression. • applying an identity other than cancelling NOTs that produces a simpler expression. • successfully expanding brackets. **Note:** A simpler expression is one that is logically equivalent to the original expression but uses fewer logical operators. **Max 3** if correct final answer but any incorrect working **Example Solution (1)** $\overline{\overline{\overline{B} . A} . \overline{B}} + A . B$ $\overline{(B + \overline{A}) . \overline{B}} + A . B$ — By De Morgan's law $\overline{B . \overline{B} + \overline{A} . \overline{B}} + A . B$ — Expansion of brackets $\overline{0 + \overline{A} . \overline{B}} + A . B$ — By identity $X . \overline{X} = 0$ $\overline{\overline{A} . \overline{B}} + A . B$ — By identity $X + 0 = X$ $A + B + A . B$ — By De Morgan's law $A + B$ — By redundancy theorem $X + X \cdot Y = X$	**4**

Example Solution (2)

$\overline{\overline{\overline{B} \cdot A} \cdot \overline{B}} + A \cdot B$

$\overline{(B + \overline{A}) \cdot \overline{B}} + A \cdot B$	By De Morgan's law
$\overline{B + \overline{A}} + B + A \cdot B$	By De Morgan's law
$\overline{B + \overline{A}} + B$	By redundancy theorem $X + X \cdot Y = X$
$\overline{B} \cdot A + B$	By De Morgan's law
$(B + \overline{B}) \cdot (A + B)$	Put into brackets
$(1 \cdot (A + B)$	By identity $X + \overline{X} = 1$
$A + B$	By identity $X \cdot 1 = X$

Example Solution (3)

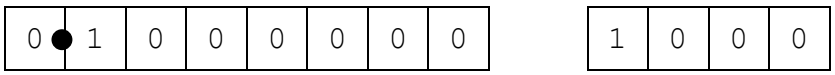
$\overline{\overline{\overline{B} \cdot A} \cdot \overline{B}} + A \cdot B$

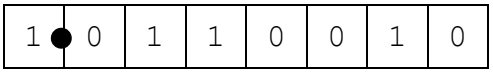
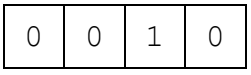
$\overline{B} \cdot A + B + A \cdot B$	By De Morgan's law
$A. (\overline{B} + B) + B$	Identify common factor A
$A \cdot 1 + B$	By identity $X + \overline{X} = 1$
$A + B$	By identity $X \cdot 1 = X$

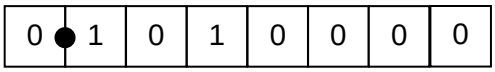
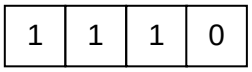
09	**1**	**All marks AO1 (understanding)** **Asynchronous:** Receiver and transmitter are not synchronised by a common clock // receiver's clock synchronised to transmitter's each time a start bit is received // receiver and transmitter clocks synchronised for duration of a transmission; **Synchronous:** Receiver and transmitter (continuously) synchronised by a common clock // timing information transmitted within/alongside the data // receiver and transmitter clocks are (continuously) synchronised; **NE.** Receiver and transmitter are synchronised **Max 1**	**1**
09	**2**	**All marks AO2 (apply)** Stop Bit, Parity Bit, Start Bit **0 1 0 1 1 0 1 0 0 1** **1 mark:** Start bit and stop bit each have the value 1 and 0 and must be different to each other (it does not matter which is 1 or 0). **1 mark:** Correct bit pattern for character '4': 0110100 **1 mark:** Parity bit is 1 **A.** parity bit of 0 if this would be correct based upon an incorrect ASCII code used and reject parity bit of 1 if would not be correct for ASCII code used.	**3**

09	**3**	**All marks AO1 (understanding)** **Improvements (Max 3):** #Errors can (sometimes) be corrected as well as detected; **A.** the location of an error can be identified #Multi-bit errors can be detected; **A.** errors that change an even number of bits can be detected *If neither of the points marked # is awarded then award one mark if the general point that transmissions should be more reliable is made.* A greater range of characters can now be transmitted; **A.** any response that implies this eg support for multiple languages // languages with large sets of characters, inclusion of specialised symbols in character set Elimination of problems caused by different versions of ASCII character sets / extended ASCII / use of code pages // eliminates problem of some ASCII codes representing different characters in different countries // Unicode values can be interpreted more consistently than ASCII codes; **Disadvantages (Max 3):** *Each character will require more bits // 8 bits // 16 bits // 32 bits // between 8 and 32 bits; *Each bit will be sent multiple times // three or more times // there will be redundancy in the data transmissions; **A.** code, character instead of bit *If neither of the points marked * is awarded then award one mark if the general point that more bits are required is made.* The (effective) rate at which information / (useful) data can be transmitted will be reduced; **A.** transmissions will take longer **R.** references to storage space	**4**

10	**1**	**2 marks AO2 (analysis) and 6 marks AO3 (programming)**
		
		Example Solution 1
		
		`  LDR R1, 102`
		`  LDR R2, 103`
		`loop:`
		`  CMP R1, R2`
		`  BEQ finish`
		`  BGT agreaterthanb`
		`  SUB R2, R2, R1`
		`  B loop`
		`agreaterthanb:`
		`  SUB R1, R1, R2`
		`  B loop`
		`finish:`
		`  STR R1, 104`
		
		Example Solution 2
		
		`  LDR R0, 102`
		`  LDR R1, 103`
		`startloop:`
		`  CMP R0, R1`
		`  BEQ end`
		`  CMP R0, R1`
		`  BGT greater`
		`  SUB R1, R1, R0`
		`  B startloop`
		`greater:`
		`  SUB R0, R0, R1`
		`  B startloop`
		`end:`
		`  STR R1, 104`
		
		Note: Any register numbers can be used and any understandable method to identify a label.
		
		DPT use of invalid register names eg R27, Rn
		
		6 marks AO3 (programming syntax must be correct):
		
		Values in memory locations 102 and 103 loaded into two different registers;
		
		Comparison made between the values in the two registers;
		
		If the values in the two registers are the same then the code will exit (after performing any other necessary instructions); **A.** end of program reached if not HALT instruction.
			8

		If A is greater than B then the value in the register representing B is subtracted from the value in the register representing A and result stored in register representing A; **Note:** Award this mark even if further incorrect changes would also be made to values in registers. If A is less than (or equal to B) / then the value in the register representing A is subtracted from the value in the register representing B and result stored in register representing B; **Note:** Award this mark even if further incorrect changes would also be made to values in registers. Before the algorithm exits, in all circumstance, the value in the register representing A (or the register representing B) is stored into memory location 104 (regardless of whether or not this is the gcd); **A.** if this is done on every iteration of a loop instead of just once. **2 marks AO2 (concept must be understood, syntax need not be correct):** The need for a loop has been identified and instructions are used to make the program loop back to before the comparison(s) after each subtraction has taken place; The response provided follows the correct method to calculate the gcd of A and B, regardless of whether the syntax is correct or not, although an attempt must have been made to use the AQA instruction set; **Max 7 if solution not fully working**	
11	**1**	**All marks AO2 (apply)**  Mantissa Exponent **1 mark:** Correct mantissa **1 mark:** Correct exponent	**2**

11	**2**	**All marks AO2 (apply)**  Mantissa  Exponent **1 method mark** for either: • showing correct value of both mantissa and exponent in decimal (mantissa = -0.609375 // -39/64 Exponent = 2) • showing binary point shifted 2 places to right in binary number • indicating that final answer calculated using answer = mantissa x $2^{exponent}$ **1 mark** for correct answer Answer -2.4375 // -39/16 // -2 7/16 **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**

11	**3**	**All marks AO2 (apply)** **2 marks** for working: Correct (unsigned) fixed point representation of 0.15625 in binary: 0.00101; **A.** leading 0s and trailing 0s **NE.** this value only shown in final answer mantissa box Showing the correct value of the exponent in decimal (-2) or binary (1110) // showing the binary point being shifted 2 places right; **A.** if only shown in final answer exponent box Bit pattern 101 is present somewhere in the final mantissa value and all of the other bits of the mantissa are 0; **Max 2** **1 mark** for correct mantissa and exponent together:  Mantissa  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.**	**3**

11	**4**	**1 mark AO2 (analysis) and 2 marks AO1 (understanding)** **Problem (1 mark AO2 (analysis)):** Overflow will occur // it will not be possible to store the result/the exponent in the available number of bits; **Solution (2 marks AO1 (understanding)):** Mark against *Method 1* or *Method 2*. *Method 1* More bits should be added; **TO.** more bits should be added to the mantissa <u>from the exponent</u> **NE.** larger/bigger, must have reference to bits To the exponent; **TO.** both mantissa and exponent *Method 2* Bits could be reallocated from the mantissa to the exponent;;	**3**
12	**1**	**Mark is AO2 (apply)** 4294967296 // 2^{32} (bytes)	**1**
12	**2**	**Mark is AO1 (knowledge)** Increase the number of bits // amount of data that can be transferred <u>at one time</u>; **A.** in one cycle **NE.** increase rate of data transfer	**1**

13		**All marks AO1 (understanding)** **Advantages of vector graphics (max 3 marks):** Individual objects/components/parts of the image can be manipulated/edited independently; **A.** example of objects **NE.** images are easy to edit The image/individual objects/components/parts of the image can be enlarged/scaled without loss of quality; **A.** "zoomed in" for enlarged **A.** example of objects **NE.** easy to scale If an object/component is deleted, the software knows what is behind it // no "hole" is left in the image; Vector graphics are resolution independent; Images saved as vector graphics (typically) take up less storage space // can be transmitted more quickly than an (equivalent) bitmap; **Advantages of bitmaps (max 3 marks):** Can represent images with complex textures // lots of variation in colour/tone (which could not be computed); **NE.** high colour depth, complex image Can represent images that are not composed of regular shapes; Images captured from nature // digital photos // scanned images are naturally represented as bitmaps (because of the method used to capture them); *Award marks for advantages of bitmaps which are stated as such or as disadvantages of vector graphics and vice-versa **but** only award one mark for the same point made both ways eg stating vectors can be enlarged without loss of quality and bitmaps pixelate when enlarged is just one mark.* **Suitable examples (max 2 marks):** • **Vector (1 mark)**: chart, logo, map, plan, clipart – any example that could be made from regular shapes. • **Bitmap (1 mark)**: photograph, scanned image, sprite icon – any example that could not be represented as a vector graphic because it is not composed of regular shapes or is taken from nature. **Max 5** for question If no valid examples given	**6**

<table>
<tr><td>14</td><td>1</td><td>All marks AO2 (apply)

1 mark: Correct conversion of ciphertext and key to binary.

1 mark: The XOR operation is applied to the binary representations of the ciphertext and key to produce the binary representation of the plaintext.
A. award this mark if one or both of the binary representations of the ciphertext and key are incorrect but the plaintext binary pattern has been produced by XORing these bit patterns

1 mark: Correct conversion of plaintext from binary to letters.
A. award this mark if the binary plaintext is incorrect but the conversion of this to letters is correct for the incorrect bit pattern. If only some bit patterns map to letters (eg 11011 does not) then accept that those which do not are not converted, but reject incorrect conversion.
I. Case eg “Dog” is fine.

<table>
<tr><td>Ciphertext in binary:</td><td>10011</td><td>01111</td><td>01101</td></tr>
<tr><td>Key in binary:</td><td>00001</td><td>01100</td><td>00110</td></tr>
<tr><td>Plaintext in binary:</td><td>10010</td><td>00011</td><td>01011</td></tr>
<tr><td>Plaintext as letters:</td><td>D</td><td>O</td><td>G</td></tr>
</table>

If answer is correct (DOG) and some working has been shown, award three marks, even if working would not have gained credit on its own.</td><td>3</td></tr>
<tr><td>14</td><td>2</td><td>Mark is AO1 (knowledge)

The cipher cannot be cracked (by any known method A. technology) in a reasonable/practical/polynomial/useful amount of time; NE. long time

A. given enough ciphertext and time the cipher could be cracked (but this is not reasonable)
R. responses that suggest the cipher could never be cracked
NE. responses about plaintext being deciphered / decrypted /decoded or the cipher solved rather than cracked, unless it is clear that this is being done without the key</td><td>1</td></tr>
<tr><td>14</td><td>3</td><td>Mark is AO1 (knowledge)

How to pass the key (from the sender) to the receiver;
A. key must be sent along with the data
NE. key must be exchanged, without a reference to the receiver

Without it being intercepted / securely;
A. the key can be intercepted if/when it is <u>transmitted</u></td><td>2</td></tr>
</table>

14	**4**	**All marks AO1 (understanding)** B's private key is used to decrypt the message (and signature); **R.** more than one key referenced The message is rehashed // a new message digest/hash is calculated from the message; A's public key is used to decrypt the digital signature (to produce the received message digest); If received message digest and recalculated message digest match / if both hashes match then the sender can be authenticated / B knows that A sent the message; **A.** if recalculated hash matches digital signature then B knows A sent message, if third mark point not awarded. **NE.** if hashes match then B knows message has not been tampered with **A.** data for message **A.** checksum, hash, digest as synonyms **A.** encrypted hash/encrypted digest for signature	**4**