A-level

COMPUTER SCIENCE

7517/2

Paper 2

Mark scheme

June 2020

Version: 1.0 Final

Mark schemes are prepared by the Lead Assessment Writer and considered, together with the relevant questions, by a panel of subject teachers. This mark scheme includes any amendments made at the standardisation events which all associates participate in and is the scheme which was used by them in this examination. The standardisation process ensures that the mark scheme covers the students' responses to questions and that every associate understands and applies it in the same correct way. As preparation for standardisation each associate analyses a number of students' scripts. Alternative answers not already covered by the mark scheme are discussed and legislated for. If, after the standardisation process, associates encounter unusual answers which have not been raised they are required to refer these to the Lead 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 2020

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.

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 4.5 (max 5 marks)
AO2 (analyse) – 3 marks
AO3 (programming) – 2 marks

Question 8.1 (max 2 marks)
AO1 (understanding) – 1 mark
AO2 (analyse) – 1 mark

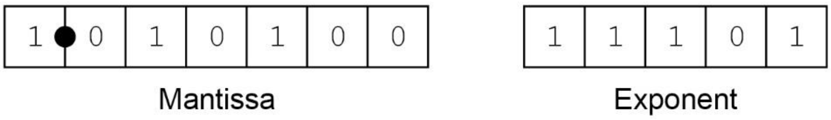
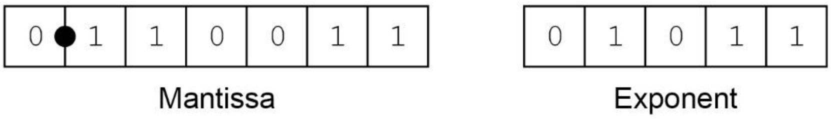
Question 8.2 (max 2 marks)
AO1 (understanding) – 1 mark
AO2 (analyse) – 1 mark

Question 8.3 (max 6 marks)
AO1 (knowledge) – 3 marks
AO2 (analyse) – 3 marks

01	**1**	**2 marks for AO2 (apply)** Award **2 marks** if correct final answer is shown: 1,600,000 (bytes); **A.** 1600kB or 1.6MB for **1 mark** but **NE.** 1600 or 1.6 without units If final answer in not given then award **2 marks** if correct calculation is shown: (60+40) x 16 x 8000 / 8 or 100 x 16 x 8000 / 8 If final answer is not given/incorrect and fully correct working is not shown then award **1 mark** for doing any three of: • multiplying by 8000 • multiplying by 100 • multiplying by 16 • dividing by 8. **A.** Multiplying by 2 as an alternative to multiplying by 16 and dividing by 8	**2**

01	**2**	**2 marks for AO1 (knowledge)** **2 marks**: All three points in list below covered OR **1 mark**: At least one point from list below covered • (Analogue signal **A.** sound as BOD) sampled at fixed/regular time intervals **R.** references to graphs • Amplitude/Voltage of signal/wave (at each sample point) measured • Measurement coded into a fixed number of bits // coded in binary	**2**

02	**1**	**2 marks for AO1 (understanding)** **Advantage of floating point (max 1):** (In a given number of bits) a floating point system can represent numbers with a greater range than a fixed point system; **A.** can represent numbers much closer to zero // can represent much smaller numbers **A.** can represent much larger numbers **Advantage of fixed point (max 1):** (In a given number of bits) a fixed point system can represent (some) numbers more precisely than a floating point system; **A.** “accurately” for “precision” as **BOD** Calculations can be performed more quickly; **NE.** time efficient; **A.** simpler evaluation Represents all numbers to a constant (**A.** fixed, guaranteed) level of precision/accuracy; **NE.** easier to understand	**2**

02	**2**	**2 marks for AO2 (apply)**  1 ● 0 1 0 1 0 0 — Mantissa 1 1 1 0 1 — Exponent Award **2 marks** for correct answer: Answer: -0.0859375 // -11/128 **A.** expressed to at least four decimal places eg -0.0859 If answer is incorrect then award **1 method mark** for either: • showing correct value of both mantissa and exponent in decimal (Mantissa = -0.6875 // -11/16 Exponent = -3) • showing binary point shifted 3 places to left in binary number • indicating that final answer calculated using answer = mantissa x $2^{exponent}$ and using at least one of the correct mantissa or exponent in this calculation.	**2**
02	**3**	**3 marks for AO2 (apply)** Award **3 marks** for correct answer: Answer  0 ● 1 1 0 0 1 1 — Mantissa 0 1 0 1 1 — Exponent If answer is incorrect then award up to **2 method marks** for: • correct (unsigned) representation of 1632 in binary: 11001100000; **A.** leading 0s **A.** omission of trailing zeros if clear what place value of each column with a 1 in it has • showing the correct value of the exponent in decimal (11) or binary (1011) // showing the binary point being shifted 11 places left; (**Note:** mark for correct exponent can be awarded if seen in final answer). • showing the correct value of the mantissa in binary (0.110011).	**3**

02	**4**	**3 marks for AO2 (apply)** **If either the highest or lowest value is correct then award 1 mark.** **Award 3 marks if both are correct:** • highest value: 32,256 **A.** 0.984375 x 2^{15} // 63/64 x 2^{15} • lowest value: -32,768 **A.** - 2^{15} // -1 x 2^{15} **If 3 marks have not been awarded for the two correct values then working marks can be awarded, but a maximum of 2 marks can be awarded for the question overall.** Working marks are available as follows: **1 mark** for expressing the highest value in binary: • 111111000000000 **A.** leading 0s OR • mantissa 0.111111 AND exponent 01111 **1 mark** for expressing the lowest value in binary: • 1000000000000000 **A.** leading 1s OR • mantissa 1.000000 AND exponent 01111 **1 mark** for doing the calculation of multiplying a value by the correct exponent in decimal (2^{15} or 32,768), regardless of whether the value is an appropriate one or the result of the calculation is correct	**3**

02	**5**	**2 marks for AO2 (analysis)** There are not enough bits in the mantissa (to represent 28.25 exactly) // 7 bits is not enough to represent 28.25 exactly // the binary representation of 28.25 has more than 7 significant digits // the binary representation of 28.25 has more significant digits than there are bits in the mantissa // binary representation of 28.25 needs 8 bits (in mantissa) / has 8 significant digits // insufficient (bits) of precision available; **R.** 28.25/some numbers can never be represented exactly in binary It could be rounded to the nearest representable value // it may be truncated // represent as 28 // represent as 28.5; **R.** an error would be generated	**2**

03 **1**

2 marks for AO1 (understanding)

2

Component Name	Component Number (1–5)
Address Bus	4
Data Bus	5
Main Memory	1
Processor	2
USB I/O Controller	3

1 mark: At least three components correctly numbered
2 marks: All five components correctly numbered

03 **2**

Mark is for AO2 (apply)

1

4,194,304 (kibibytes);
A. 1024 x 1024 x 4
A. 4 x 2^{20}
A. 2^{22}

03 **3**

2 marks for AO1 (understanding)

2

Avoid/reduce bottleneck of single data/address bus(es) // avoid/reduce delays waiting for memory fetches; **A.** <u>Instruction and data</u> can be accessed simultaneously;

Avoids possibility of data being executed as code (which is one method that can be exploited by hackers) // Being able to use exclusively ROM for instruction memory prevents the program being modified/hacked; **A.** program cannot be (accidentally) overwritten (by data)

Instruction and data memory can have different word lengths;

Different technologies can be used to implement instruction and data memory;

Different quantities of instruction and data memory means that address lengths can differ between the two // memory address structures can differ;

Max 2

NE. So programs/tasks will run faster
NE. More efficient
NE. Quicker access, without further explanation
NE. Instructions and data stored in different memories

<table>
<tr>
<td><b>03</b></td>
<td><b>4</b></td>
<td><b>2 marks for AO1 (knowledge)</b>
<table>
<tr><th>Number</th><th>Register Name</th></tr>
<tr><td>❶</td><td>Memory Address Register <b>NE.</b> MAR</td></tr>
<tr><td>❷</td><td>Program Counter <b>NE.</b> PC</td></tr>
<tr><td>❸</td><td>Current Instruction Register <b>NE.</b> CIR, IR
<b>A.</b> Instruction Register</td></tr>
</table>
<b>1 mark:</b> Two registers correctly named OR
<b>2 marks:</b> All three registers correctly named

If student has used initialisms instead of full register names (or a mixture of both) then award <b>1 mark</b> if all three registers are given the correct abbreviated name.</td>
<td><b>2</b></td>
</tr>
</table>

03	**5**	**2 marks for AO1 (knowledge)** Allows the currently executing process/task/program to be suspended; **A.** "stopped" as BOD **R.** Suspend/stop the fetch-execute cycle / processor **R.** "instruction" for "process" So that a device/source that needs the (immediate) attention of the processor can be serviced/dealt with // so that an urgent error condition can be serviced/dealt with; **A.** Examples of error conditions that would be likely to generate an interrupt **NE.** To deal with an error, unless stated or clear from example that must be dealt with immediately **NE.** So that a task of higher priority can be carried out	**2**

03	**6**	**2 marks for AO1 (understanding)** So that the currently running process/task/program can be returned to; **NE.** So that the content will not be lost/overwritten **NE.** So that the F-E cycle can continue afterwards As the (code that deals with the) interrupt will change/overwrite/clear register values; **NE.** The contents of the registers will be lost	**2**

03	**7**	**Mark is for AO1 (knowledge)** Software (is the programs that) execute(s) on the hardware // hardware is the electrical/physical components that allow the software to execute; **A.** Software controls the operation of the hardware as BOD	**1**

04	**1**	**Mark is for AO1 (knowledge)** D; **R.** if more than one lozenge shaded	**1**

04	**2**	**Mark is for AO1 (understanding)** B; **R.** if more than one lozenge shaded	**1**

04	**3**	**2 marks are for AO1 (understanding)** More compact; **A.** facilitates faster transmission, smaller file size, uses less memory Quicker (**A.** easier) to parse; Structure understood directly in some languages (eg Javascript); (Native) support for arrays; Easier <u>for humans</u> to read/write/understand; **Max 2**	**2**

04	**4**	**Mark is for AO2 (analysis)** That the buyer will only view the same property once on a particular day; **R.** each visit made by only one buyer	**1**

04	5	**3 marks for AO2 (analyse) and 2 marks for AO3 (programming)**	5

Mark Scheme

AO2 (analyse) – 3 marks:

1 mark for correctly analysing the data model and identifying the tables that data needs to be extracted from (`Property, Buyer`) and the fields that need to be extracted (`PropertyID, Street, Bedrooms, AskingPrice`), and including these and no other tables or fields in the query.

1 mark for correctly identifying two conditions relating to how the data in the required tables should be combined to produce the desired results OR **2 marks** for identifying all four required conditions. The four conditions are:

- `BuyerID = 23`
- `Buyer.DesiredArea = Property.Area`
- `Buyer.MinBedrooms <= Property.Bedrooms`
- `Buyer.MaxPrice >= Property.AskingPrice`

Note: The AO2 marks for analysing the data model should be awarded regardless of whether correct SQL syntax is used or not as they are for data modelling, not syntactically correct SQL programming

AO3 (programming) – 2 marks:

1 mark for fully correct SQL in two of the three/four clauses (SELECT, FROM, WHERE, ORDER BY)

OR

2 marks for fully correct SQL in all three/four clauses (SELECT, FROM, WHERE, ORDER BY)

Note: For an SQL clause to be counted as “fully correct”, the syntax of the clause must be correct and the relevant AO2 decisions must also have been taken for the clause, eg the `SELECT` clause must have the correct fields in it only

Example Solutions

Example 1

```
SELECT PropertyID, Street, Bedrooms, AskingPrice
FROM Buyer, Property
WHERE BuyerID = 23
  AND DesiredArea = Area
  AND MinBedrooms <= Bedrooms
  AND MaxPrice >= AskingPrice
ORDER BY AskingPrice DESC
```

Example 2

```
SELECT PropertyID, Street, Bedrooms, AskingPrice
FROM Buyer INNER JOIN Property
  ON DesiredArea = Area
WHERE BuyerID = 23
  AND MinBedrooms <= Bedrooms
  AND MaxPrice >= AskingPrice
ORDER BY AskingPrice DESC
```

Example 3

```
SELECT PropertyID, Street, Bedrooms, AskingPrice
FROM Buyer INNER JOIN Property
  ON DesiredArea = Area
  AND MinBedrooms <= Bedrooms
  AND MaxPrice >= AskingPrice
WHERE BuyerID = 23
ORDER BY AskingPrice DESC
```

Example 4 – A Nested Solution

```
SELECT PropertyID, Street, Bedrooms, AskingPrice
FROM (SELECT DesiredArea, MinBedrooms, MaxPrice
      FROM Buyer
      WHERE BuyerID = 23) AS Requirements
     INNER JOIN Property
WHERE DesiredArea = Area
  AND MinBedrooms <= Bedrooms
  AND MaxPrice >= AskingPrice
ORDER BY AskingPrice DESC
```

Refer nested solutions to team leaders for marking

Overall Max 4 if solution does not work fully

<u>Additional Guidance</u>

AO2 marks:

Mark(s) can be awarded for the correct logical conditions even if the required tables are not identified as being used by the query

AO3 marks:

A. table names before fieldnames separated by a full stop.
A. use of `Alias`/`AS` command eg `FROM Buyer AS B` then use of `B` as the table name but note that command `AS` is not required eg `FROM Buyer B`.
A. `INNER JOIN` written as one word ie `INNERJOIN`.

		A. `ORDER BY` written as one word ie `ORDERBY`. **A.** insertion of spaces into fieldnames. **I.** unnecessary brackets so long as they would not stop the query working **I.** quotation marks of any type around the `23` **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.	

05	**1**	**Mark is for AO1 (understanding)** Reduces the need for expert knowledge when configuring a host; **A.** No requirement to manually assign IP addresses / other values **A.** Automatic assignment of IP addresses Reduces the time required to configure hosts; Facilitates efficient use of a limited pool of IP addresses; **A.** Example of how this is facilitated eg reuse Avoids errors - with a relevant example such as duplicating IP addresses or programming incorrect subnet mask; **NE.** "avoiding errors" without an example **Max 1**	**1**

05	**2**	**2 marks for AO1 (understanding)** The computers have private/non-routable IP addresses // 192.168.2.3 is a private/non-routable IP address; **NE.** The computers can have the same IP addresses as they are on different networks NAT/Network Address Translation will be performed (so that the computers can communicate on the Internet) // as data passes onto Internet, private IP address replaced with public IP address of router/gateway;	**2**

05	**3**	**4 marks for AO1 (understanding)** Block/allow (traffic on) specific ports // block specified protocols; Block/allow (traffic from) specific IP addresses; **A.** Domain names as BOD **NE.** Block access to certain websites **R.** MAC addresses Block/allow certain types of packet; **A.** Examples eg pings/echo requests **NE.** Block specific programs connecting to Internet Firewall maintains information about current connections and only allows packets relevant to these connections through; **NE.** Just the name “stateful inspection” Act as a proxy server // all traffic to Internet must go via firewall // stops computers on the Internet directly accessing devices on the LAN; Identify unusual behaviour from a host // example of unusual behaviour eg sending an unusually large amount of data; Rules are written to specify conditions under which to block/allow; If none of the first three marks scheme points awarded then a mark can be awarded for: Examine the contents of the packet header and allow/block based on rules; **NE.** Just the name “packet filtering” **Max 4**	**4**

05 **4**

12 marks for 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 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. A good level of understanding has been shown of at least one area or a reasonable understanding has been shown of at least 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. There is insufficient evidence of a good understanding of any of the three areas.	1–3

<u>Guidance – Indicative Response</u>

Determining if on LAN:

- AND operation of subnet mask with Computer A's IP address
- AND operation of subnet mask with Computer B's IP address
- result (of each AND operation) is the network/subnet ID
- network/subnet IDs compared
- as they are different, then packet must be sent via router/gateway/Internet // Computer B is not on the same subnet
- if they were the same, then packet can be sent directly to Computer B // Computer B is on the same subnet.

If no other points made, then a very basic understanding could be shown by recognising that the subnet mask is used with the IP addresses to determine if the two computers are on the same network/subnet.

Good level of understanding = most of the key elements listed above are covered.

Routing across Internet:

- hierarchical organisation of routers
- example of hierarchical organisation of routers eg 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 eg congestion, technical problems • (possible) repackaging of packet to use different protocol (eg 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" • IP address of Computer A will be replaced with IP address of Router A3 // NAT / Network Address Translation will occur at router(s). *Good level of understanding = most of the key elements listed above are covered.* **Checksum:** • checksum produced when packet transmitted // by computer A • (hash) value / checksum calculated from packet contents • MOD operation (often) used to limit magnitude of checksum // fit value to specific number of bits • this value / checksum transmitted with packet // appended to packet • computer B recalculates checksum // performs same calculation on data • received and calculate checksum compared • if these match packet contents/data are accurate // if these differ the data has been changed // if these differ there is an error in the data. *Good level of understanding = most of the key elements listed above are covered.*	

06	**1**	**Mark is for AO1 (knowledge)** Table C; **R.** if more than one lozenge shaded	**1**

06	**2**	**4 marks for AO2 (apply)**	**4**

INPUTS				INTERMEDIATE POINTS					OUTPUT
X3	**X2**	**X1**	**X0**	**A**	**B**	**C**	**D**	**E**	**Q**
0	*0*	*0*	*0*	1	0	1	0	0	0
0	*0*	*0*	*1*	*1*	*0*	*1*	*0*	*0*	*0*
0	*0*	*1*	*0*	1	1	1	0	1	1
0	*0*	*1*	*1*	*1*	*1*	*0*	*0*	*1*	*1*
0	*1*	*0*	*0*	0	0	1	1	1	1
0	*1*	*0*	*1*	*0*	*0*	*1*	*1*	*1*	*1*
0	*1*	*1*	*0*	*0*	*0*	*1*	*1*	*1*	*1*
0	*1*	*1*	*1*	0	0	0	0	0	0
1	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*1*
1	*0*	*0*	*1*	1	0	1	0	0	1

Marks are awarded for the correct values in the unshaded cells only.

1 mark: Correct values in columns A or C
1 mark: Correct values in columns B or D
1 mark: Correct values in column E
1 mark: Correct values in column Q

Max 3 if any incorrect values in table

06	**3**	**Mark is for AO2 (analyse)** b; **A.** the middle bar	**1**

06	**4**	**4 marks for AO2 (apply)** **Marking guidance for examiners:** • award marks for working out until an incorrect step has been made • ignore missing steps from the example solutions, as long as the jumps between steps are logically correct • 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: $A \cdot B$ **Max 3** for working. Award up to three 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. **Example Solution (1)** $\overline{\overline{\mathrm{A} \cdot (\mathrm{A} + 1)} \cdot \overline{\mathrm{B}}} \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B} + 0}}$ $\overline{\overline{\mathrm{A} \cdot 1} \cdot \overline{\mathrm{B}}} \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B} + 0}}$ By $X + 1 = 1$ $\overline{\overline{\mathrm{A}} \cdot \overline{\mathrm{B}}} \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B} + 0}}$ By $X \cdot 1 = X$ $\overline{\overline{\mathrm{A}} \cdot \overline{\mathrm{B}}} \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B}}}$ By $X + 0 = X$ $(\mathrm{A} + \mathrm{B}) \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B}}}$ Application of De Morgan $(\mathrm{A} + \mathrm{B}) \cdot (\mathrm{A} \cdot \mathrm{B})$ Application of De Morgan $\mathrm{A} \cdot \mathrm{A} \cdot \mathrm{B} + \mathrm{B} \cdot \mathrm{A} \cdot \mathrm{B}$ Expansion of brackets $\mathrm{A} \cdot \mathrm{B} + \mathrm{B} \cdot \mathrm{A} \cdot \mathrm{B}$ By $X \cdot X = X$ $\mathrm{A} \cdot \mathrm{B} + \mathrm{B} \cdot \mathrm{A}$ By $X \cdot X = X$ $\mathrm{A} \cdot \mathrm{B}$ By $X + X = X$ **Example Solution (2)** $\overline{\overline{\mathrm{A} \cdot (\mathrm{A} + 1)} \cdot \overline{\mathrm{B}}} \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B} + 0}}$ $\overline{\overline{\mathrm{A} \cdot 1} \cdot \overline{\mathrm{B}}} \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B} + 0}}$ By $X + 1 = 1$ $\overline{\overline{\mathrm{A}} \cdot \overline{\mathrm{B}}} \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B} + 0}}$ By $X \cdot 1 = X$ $\overline{\overline{\mathrm{A}} \cdot \overline{\mathrm{B}}} \cdot \overline{\overline{\mathrm{A}} + \overline{\mathrm{B}}}$ By $X + 0 = X$ $\overline{\overline{\mathrm{A}} \cdot \overline{\mathrm{B}} + \overline{\mathrm{A}} + \overline{\mathrm{B}}}$ Application of De Morgan $\overline{\overline{\mathrm{A}} + \overline{\mathrm{B}}}$ By $X + X \cdot Y = X$ $\mathrm{A} \cdot \mathrm{B}$ Application of De Morgan	**4**

		Example Solution (3) $\overline{\overline{A\cdot(A+1)}\cdot\overline{B}}\cdot\overline{\overline{A}+\overline{B+0}}$ $\overline{A\cdot(A+1)}\cdot\overline{B}+\overline{A}+\overline{B+0}$ Application of De Morgan $\overline{A\cdot A}\cdot\overline{B}+\overline{A}+\overline{B+0}$ By $X+1=1$ $\overline{A}\cdot\overline{B}+\overline{A}+\overline{B+0}$ By $X\cdot X=X$ $\overline{A}+\overline{B+0}$ By $X+X\cdot Y=X$ $\overline{A}+\overline{B}$ By $X+0=X$ $A\cdot B$ Application of De Morgan	

07	**1**	**Mark is for AO2 (apply)** A; **R.** more than one lozenge shaded	**1**

07	**2**	**Mark is for AO1 (understanding)** E; **R.** more than one lozenge shaded	**1**

08	**1**	**1 mark for AO2 (analyse) and 1 mark for AO1 (understanding)** **AO2 (analyse) 1 mark:** A lot of individual products will need to be scanned simultaneously / when a lorry/delivery arrives/leaves; **AO1 (understanding) maximum 1 mark from this list:** • the RFID tags could be read without removing products from their pallet; • RFID tags can be read from a (greater) distance (than barcodes); • no need for a person to scan tags // no need to manually scan tags; • RFID tags can be read at a faster rate; • RFID tags less easily damaged (than barcodes) // barcodes can become easily damaged and made unreadable; • don’t have to spend time locating barcodes on items; **R.** RFID tags can store more data (not relevant)	**2**

08	**2**	**1 mark for AO2 (analyse) and 1 mark for AO1 (understanding)** **AO1 (understanding) 1 mark:** Barcodes are cheaper than RFID tags // less electronic waste (assuming tags not reused) // barcodes can only be read when scanner pointed directly at them // barcodes not susceptible to radio interference // barcodes usually include human-readable encoding of same data; **AO2 (analyse) 1 mark:** Higher cost of RFID tags would be added on to prices // higher cost would have to be paid by manufacturer/supermarket/customer // barcodes can be scanned by existing equipment at checkouts // less risk of nearby product being accidentally scanned // backup system exists as barcodes can be keyed in;	**2**

08	**3**	**3 marks for AO1 (knowledge) – 3 marks:** • RFID reader/scanner (at warehouse entrance) transmits/sends signal; • Signal activates/energises/induces current in RFID transponder/tag; • RFID transponder/tag transmits/sends data by radio(wave); **3 marks for AO2 (analyse) – Max 3 marks:** • RFID signals processed into a format suitable for querying the database; • (SELECT) query used to check if there is already a record for the product/ProductID in the database // return of empty data set could be used to identify if the product is not in the table; **NE.** used to lookup record • UPDATE statement used to increase the QuantityInStock/stock level (by the number of items delivered) <u>if the product is already in the database</u>; • INSERT statement used to create new record for product <u>if it is not already in the database</u>; user will need to enter some details manually as these are not contained in the RFID tag.	**6**

09	**1**	**Mark is for AO1 (understanding)** 1024 // 2^{10}; **A.** 1 KiB	**1**

09	**2**	**Mark is for AO1 (knowledge)** The operand is the value/datum that the instruction should use;	**1**

09	3	**3 marks for AO3 (programming)** Values in memory locations 101 and 102 loaded into two different registers; Contents of the two registers are exclusive ORed; **A.** Memory addresses used as operands directly if no other marks awarded for this question part ie `EOR 103, 101, 102` **A.** Exclusive or achieved in another way eg use of two ANDs, two NOTs and an OR Value of register storing the result of exclusive or operation is stored into memory location 103; **A.** result of an incorrect combination of the values in locations 101 and 102 stored in location 103 **DPT.** Use of invalid register name eg Rd **DPT.** Use of incorrect addressing mode **DPT.** Inclusion of invalid symbols in commands **Example Solution**	3

```
LDR R1, 101
LDR R2, 102
EOR R3, R1, R2
STR R3, 103
```

09	4	**3 marks for AO2 (analyse)** **What the problem is:** Some letters will be shifted back before the letter A // before the start of the alphabet // before ASCII code 65 // some letters will end up as non-alphabetic characters; **R.** Some values will not be valid ASCII codes **Solution:** These need to be shifted back // wrapped around to the end of the alphabet // use an If statement to check if the code is below 65; Shifting achieved by adding 26 to any code below 65 // by using MOD 26 in the calculation;	3

10		**6 marks for AO1 (understanding)** **Area 1: How it could work:** • members could specify their interests / views and stories could be matched to these; • consider basic facts about member eg age, gender, location; • consider what stories have been read by friends of the member; • analyse the type of stories that the user has read before; • analyse the information that a member shares about themselves to identify characteristics/interests etc; • track how popular a story is to display the most popular ones; • look at member's search history; • look at member's reaction to other similar stories eg likes; • show stories viewed by others with a similar profile to this user; • display articles that have been more popular // had more hits // received more positive feedback; • compare keywords in articles with keywords in articles previously viewed by the member; • how can the algorithm avoid displaying click-bait?; **Area 2: Legal** • who owns the copyright in the story?; • is it legal for the company to reproduce a news story that someone else has written?; • is the company legally responsible for the content/accuracy of stories?; • do contracts need to be signed between the company and the organisations/ individuals that stories will be displayed from?; • do laws in some countries prevent some types of stories being displayed? // need to ensure laws in different countries are followed; • need to ensure that stores are age-appropriate; • need to notify members about how their information is being used to select stories; **Area 3: Ethical / Moral** • by choosing what news stories to display, will the service influence the views of members?; • how should the company deal with governments/organisations who might want to influence/control which stories are displayed?; • should the company accept payments to promote stories?; • how should the company deal with complaints / issues raised by members (in a timely fashion)?; • will the reproduction of news stories adversely (or positively) affect the number of people who go to read the original stories from their authors?; • how can / should reliability of stories be checked / shown (fake news)?; • how can / should the company assess bias / prevent spread of propaganda; • does the company have a duty to try to provide balance?; • should a method be provided so members can request their data is not analysed for this purpose? // importance of consent; • should the company let them know that the news they are seeing is being tailored to them / not everyone sees the same news?;	**6**

		Area 2 or 3: Legal OR Ethical / Moral • should the company have people who read/check each story?; Is it practical to do this?; • how should the company select which organisations/individuals it will display stories from?; **Max 4 if all points are from one area**	

11	**1**	**Mark is for AO2 (apply)** **1 mark:** Both head and tail are correctly identified: • Head: `"Blackpool"` • Tail: `["Paris", "New Brighton", "Toronto"]` **I.** If quotation marks are omitted **A.** Omissions of brackets from the tail or addition of brackets to the head, this time only	**1**

11	**2**	**3 marks for AO2 (analysis)** The function is recursive; It splits the list up into the head and the tail; It calls itself with the tail of the list that it was called with (as an argument); Each call adds the value that is the head of the list to the total/sum of the values in the tail of the list; The recursion terminates when the list is empty (by returning 0); **Max 3**	**3**

11	**3**	**3 marks for AO1 (knowledge)** A function that takes a function as an argument; and/or returns a function as a result; **Max 2** **A.** “Parameter”, “Input” for “Argument” **NE.** A function that uses another function **R.** Explanations that are specifically of the `map` or `fold` functions and do not explain higher-order	**2**

11	**4**	**Mark is AO2 (apply)** 12;	**1**