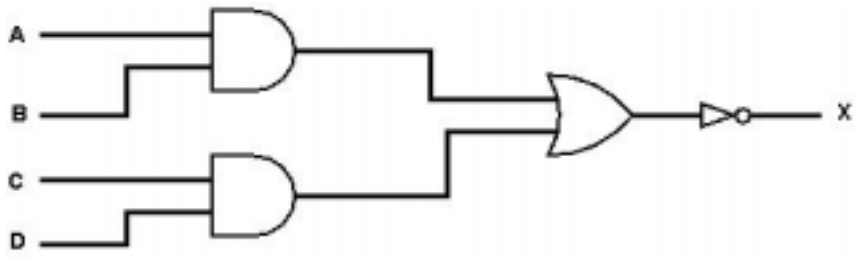
Question	Answer	Marks
1(a)	**1 mark** for each correct answer: *NAND* • The output is 0 when both inputs are 1, otherwise the output is 1 *NOR* • The output is 1 when both inputs are 0, otherwise the output is 0 *XOR* • The output is 1 when one of the inputs is 1 and the other input is 0, otherwise the output is 0 *OR* • The output is 0 when both inputs are 0, otherwise the output is 1	**4**
1(b)	**1 mark** for both AND gates with correct inputs **1 mark** for correct OR and NOT gates with correct inputs and no superfluous gates: 	**2**

Question	Answer	Marks
2(a)	**1 mark** for each correctly completed statement: • **(**LCD) displays/screens/lenses • gyroscope/accelerometer • direction/speed • digital cameras A headset can have one or two **(LCD) displays/screens/lenses** that output the image to the user. The headset has speakers that output surround sound to give a realistic experience. The user's head movements are detected using a sensor. This sensor is a **gyroscope/accelerometer**. The data is transmitted to a microprocessor that analyses the data to identify the **direction/speed** of movement. Some headsets use **digital cameras** that record the user's eye movements for analysis.	**4**
2(b)	**1 mark** each to **max 3**: • The buffer is used as a **temporary** store for data going to the headset • Data is **transferred** into the buffer by the computer • Data is **retrieved** from the buffer by the headset • When the buffer is empty/full an interrupt is sent to the computer requesting more data/stopping further data being sent • When the headset has enough data/needs more data, an interrupt is sent by the headset to the computer to stop sending data from buffer	**3**
2(c)	**1 mark** each to **max 3**: • EEPROM allows **frequent/multiple** read/write/erase operations • … so the headset can take advantage of new features • … without fully erasing the contents of the firmware in the headset first // can erase a particular byte or the whole EEPROM • … without removing the chip(s)/firmware from the headset • ... the contents of the firmware in the headset can be changed by the user without technical expertise • Cheaper to manufacture so headset will be cheaper to purchase	**3**
2(d)(i)	**1 mark** each: • Image is made of pixels and each pixel has one colour • Each colour has a **unique** binary code • **Code for the colour** of each pixel is stored in **sequence**	**3**
2(d)(ii)	**1 mark** each to **max 2**: • List of objects in the drawing • A list that stores the command/description/equation required to draw each object • Properties of each object e.g. the fill colour, line weight/colour	**2**

Question	Answer	Marks
2(d)(iii)	**1 mark** each to **max 2**: • Dedicated connection to the headset // not sharing bandwidth • Already fast connection that can transmit the data without slowing • Video may already be a small file size and does not need further reduction • Video is not saved so storage is not an issue in the headset	**2**

Question	Answer	Marks
3(a)(i)	**1 mark** for security measure **1 mark** each to **max 2** for how the chosen measure works: • Firewall • Checks **incoming** connections • ... against criteria • Blocks data from entering specific ports • Blocks data that does not meet whitelist that meets blacklist • Proxy server • Prevents devices accessing the web server directly • Intercepts any requests • Forwards the request using its own IP address • Screens returning data before sending it to the user	**3**
3(a)(ii)	**1 mark** for security measure **1 mark** each to **max 2** for description of the chosen measure: • Encryption • Encodes/scrambles data • ... so if it is intercepted it cannot be **understood** • Algorithm/key is required to decode the data	**3**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(b)</td><td>1 mark for characteristic
1 mark for description of application to examination software:

<table>
<tr><th>Thin-client characteristic</th><th>Description of use in this software</th></tr>
<tr><td>Data is not stored on the client computer</td><td>Exam papers are stored on the server and not on the examiner’s computer // exam papers are not permanently stored on the examiners’ computers</td></tr>
<tr><td>Client computer is reliant on access to server</td><td>Examiners cannot mark if their device cannot access the server / the server ‘goes down’</td></tr>
<tr><td>Client computer heavily reliant on network/internet connection</td><td>The marking software will not operate without network/internet access</td></tr>
<tr><td>Client computer requires few local resources/memory</td><td>Examiners can use devices with low resources and the marking software will still function</td></tr>
<tr><td>Client computer performs minimal functions/processes</td><td>The marking software transmits requests, the server responds and sends the response to the user</td></tr>
</table></td><td>4</td></tr>
<tr><td>3(c)(i)</td><td>1 mark each to max 2:

• Receives packets from internet
• Analyses the destination IP address of each packet
• Forwards the packet towards its destination
• ... using the routing table
• Maintains/updates the routing table
• Finds the most efficient route to the destination</td><td>2</td></tr>
<tr><td>3(c)(ii)</td><td>1 mark each to max 2:

• The PSTN consists of many different types of communication lines
• … therefore the digital data may need to be converted into a different form/analogue signal
• Data is transmitted in both directions at the same time // duplex data transmission
• Using a PSTN the communication passes through different switching centres/ISPs</td><td>2</td></tr>
</table>

Question	Answer	Marks
4(a)	**1 mark** for: 1-to-many	**1**

4(b)	**1 mark** each:
	
	• Creating table EXAM with opening and closing brackets
	• All fields with appropriate data types and commas at end of lines
	• ExamID as primary key
	
	Example:
	`CREATE TABLE EXAM(`
	`ExamID varchar NOT NULL,`
	`Subject varchar,`
	`Level int,`
	`TotalMarks int,`
	`PRIMARY KEY(ExamID)`
	`);`
		3

4(c)	**1 mark** each:
	
	• Altering table EXAM_QUESTION
	• Linking ExamID to ExamID in EXAM
	
	Example.
	`ALTER TABLE EXAM_QUESTION`
	`ADD FOREIGN KEY (ExamID) REFERENCES EXAM(ExamID);`
		2

4(d)	**1 mark** each to **max 5**: • STUDENT table identified with suitable Primary Key • A linking table between STUDENT and EXAM with suitable Primary Key **and** appropriate name • … that includes the Primary Key of the STUDENT table as a Foreign Key to join with STUDENT • ... and includes the Primary Key of the EXAM table as a Foreign Key to join with EXAM • A linking table between STUDENT and EXAM_QUESTION with suitable Primary Key **and** appropriate name • … that includes the Primary Key of Table 2 as a Foreign Key to join with Table 2 • … that stores the ExamQuestionID and the mark for that question	**5**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>1 mark for each correct answer:

<table>
<tr><th>Program Number</th><th>Code</th><th>ACC Content</th></tr>
<tr><td>1</td><td>LDI 15
SUB #1</td><td>11</td></tr>
<tr><td>2</td><td>LDD 14
ADD 11</td><td>13</td></tr>
<tr><td>3</td><td>LDM #11
ADD #3
SUB 16</td><td>2</td></tr>
<tr><td>4</td><td>LDR #2
LDX 14
ADD #2</td><td>14</td></tr>
</table></td><td>4</td></tr>
<tr><td>5(b)</td><td>1 mark for each correct answer:

<table>
<tr><th>Program Number</th><th>Code</th><th>ACC Content</th></tr>
<tr><td>1</td><td>XOR 29</td><td>0100 1001</td></tr>
<tr><td>2</td><td>AND #29</td><td>0000 0100</td></tr>
<tr><td>3</td><td>OR B11111111</td><td>1111 1111</td></tr>
</table></td><td>3</td></tr>
</table>

Question	Answer	Marks
6	**1 mark** each to **max 4** **Max 3** marks for each management task: *Memory management:* **Max 3 marks** • Stores data from all currently running programs concurrently in RAM • Stops the data from overwriting each other in RAM/primary storage • Decides which processes should be in main memory • Makes efficient use of memory *Process management:* **Max 3 marks** • Allows one process to be paused whilst another process can be actioned • Decides which process is to be run next • Switches between processes to allow them to share the use of the processor • Identification/description of scheduling	**4**

Question	Answer	Marks
7(a)	**1 mark** for: 3300 kibibytes	**1**
7(b)	**1 mark** each: • Converting 100 to binary **0110 0100 and** 10 to binary **0000 1010** • Subtraction method - converting 10 to – 10 and adding // direct subtraction … • … correct answer **0101 1010** Method 1: Converting to -10 and adding: Binary for +10 is 0000 1010 Binary for –10 is 1111 0110 Binary for 100 is 0110 0100 100 + (-10): 0 1 1 0 0 1 0 0 +1 1 1 1 0 1 1 0 (1) 0 1 0 1 1 0 1 0 Carries: 1 1 0 0 1 0 0 0 Method 2: Direct Subtraction Borrows: 0 0 0 1 1 0 1 0 0 1 1 0 0 1 0 0 – 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0	**3**
7(c)	**1 mark** for working: 1100 0000 1111 // 2048 + 1024 + 8 + 4 + 2 + 1 // (12 * 16^2) + 15 // (12 * 16 * 16) + 15 // 3072 + 15 **1 mark** for correct answer: 3087	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>8(a)</td><td>1 mark each to max 2:

• Creates an executable file
• … so the code can be tested multiple times without having to recompile
• … so repeated testing takes less time</td><td>2</td></tr>
<tr><td>8(b)</td><td>1 mark for identification of each feature and 1 mark for matching description:

e.g.
For coding:
<table><tr><th>IDE feature</th><th>Description</th></tr><tr><td>Context-sensitive prompts</td><td>Gives suggestions for code as the user types instead of having to write/remember the code</td></tr><tr><td>Auto-correct</td><td>Corrects spelling mistakes so that user has fewer errors to correct</td></tr></table>
For presentation:
<table><tr><th>IDE feature</th><th>Description</th></tr><tr><td>Pretty-printing</td><td>Colour code keywords so the user can identify any errors</td></tr><tr><td>Expand/collapse (code) blocks</td><td>The user can hide code that they are not currently working on</td></tr></table>
For debugging:
<table><tr><th>IDE feature</th><th>Description</th></tr><tr><td>Single stepping</td><td>Run the code one line at a time // shows the effect of each line of code</td></tr><tr><td>Breakpoints</td><td>Stop the code running at a set point to check the flow/variable contents</td></tr></table></td><td>6</td></tr>
<tr><td>8(c)</td><td>1 mark each to max 2:

• Saves programming/testing time as code does not have to be written/re-written from scratch // code does not have to be tested
• Code is already tested so it is more robust/likely to work
• The programmer does not need to maintain the library // library routines</td><td>2</td></tr>
</table>