

1	1 mark for 2 or 3 correctly connected verification method boxes, 2 marks for all 4 correct Verification Method Use <pre> graph LR subgraph Verification_Method [Verification Method] PBC[Parity byte check] CS[Checksum] VC[Visual check] PBlC[Parity block check] end subgraph Use [Use] DT[Data transfer] DE[Data entry] end PBC --> DT CS --> DT VC --> DT VC --> DE PBlC --> DE </pre>
---	--

6(a)	Number: Any number where the first two digits sum to 5 / 15 and the last three digits are 623 Explanation: The remainder is the same / 3 // The sum of the (first four) digits is the same / 13
------	--

6(b)	Example solution – conversion of Number to a string <pre> FUNCTION Generate(Number : INTEGER) RETURNS INTEGER DECLARE Total, Index, CheckDigit : INTEGER DECLARE NumString : STRING Total ← 0 NumString ← NUM_TO_STR(Number) FOR Index ← 1 TO LENGTH(NumString) Total ← Total + STR_TO_NUM(MID(NumString, Index, 1)) NEXT Index CheckDigit ← Total MOD 10 Number ← Number * 10 + CheckDigit // NumString ← NumString & NUM_TO_STR(CheckDigit) RETURN Number // STR_TO_NUM(NumString) ENDFUNCTION </pre> Mark as follows: - Function heading, parameters return type and ending - Initialise Total - Convert Number to a string and loop for length of converted Number attempt to form sum of digits in a loop - completely correct sum of digits in a loop - Calculate CheckDigit - Add CheckDigit to original number multiplied by 10 and return number // - Convert CheckDigit to a character and appended to NumString and then convert to a number and return
------	--

6(b)	Example Solution – no conversion to string <pre> FUNCTION Generate(Number : INTEGER) RETURNS INTEGER DECLARE Total, CheckDigit, Remainder, Num: INTEGER Num ← Number Total ← 0 WHILE Num > 0 Remainder ← Num MOD 10 Num ← Num DIV 10 Total ← Total + Remainder ENDWHILE CheckDigit ← Total MOD 10 Number ← Number * 10 + CheckDigit RETURN Number ENDFUNCTION </pre> Mark as follows: 1 Function heading, parameters return type and ending 2 Initialise Total 3 Loop while Num > 0 4 attempt to use MOD and DIV to sum successive digits in Number 5 completely correct sum of digits 6 Calculate CheckDigit 7 Add CheckDigit to original number and return number
------	---

5(a)	1 mark each: • Identification of server in the bank scenario • Description e.g. Receives requests, processes the requests • Identification of client in bank scenario • Description e.g. Sends request to the server, waits and outputs the response
5(b)	1 mark for each correctly completed term: • odd or even • 7-bits • odd • block • byte Computer A and Computer B agree on whether to use odd or even parity. Computer A divides the data into groups of 7-bits. The number of 1s in each group is counted. If the agreed parity is odd and the group has an even number of 1s a parity bit of 1 is appended, otherwise a parity bit of 0 is appended. In a parity block check the bytes are grouped together, for example in a grid. The number of 1s in each column (bit position) is counted. A bit is assigned to each column to make the column match the parity. These parity bits are transmitted with the data as a parity byte.
5(c)(i)	1 mark each to max 3: • Compares all incoming and outgoing transmissions • ... against set criteria/whitelist/blacklist • Blocks all transmissions that do not meet rules • Blocks data entering from specific ports • Blocks unauthorised/unknown internal software transmitting data
5(c)(ii)	1 mark each to max 4: e.g. • Captures an image of the face • Uses image recognition • Trained to identify the features of a face in an image • ... using a large number of images • Analyse images for facial features • Uses the probability of a match

2(a)(i)	1 mark for definition: - Data about the data in the database // data about the structure of the database // metadata for a database 1 mark for a suitable example Examples: - table names - data types - field names								
2(a)(ii)	1 mark for definition - Methods of making sure the data is consistent 1 mark for example Examples: - Enforcing referential integrity - If data in one table is deleted/edited all tables are updated // cascading update/delete - Validation/verification rules								
2(b)(i)	1 mark for each field name and table <table border="1"> <thead> <tr> <th>Foreign key</th><th>Database table</th></tr> </thead> <tbody> <tr> <td>BirdID</td><td>BIRD_SEEN</td></tr> <tr> <td>PersonID</td><td>BIRD_SEEN</td></tr> </tbody> </table>	Foreign key	Database table	BirdID	BIRD_SEEN	PersonID	BIRD_SEEN		
Foreign key	Database table								
BirdID	BIRD_SEEN								
PersonID	BIRD_SEEN								
2(b)(ii)	1 mark for all 3 correct lines <table border="1"> <thead> <tr> <th>Normal Form</th><th>Definition</th></tr> </thead> <tbody> <tr> <td>First Normal Form (1NF)</td><td>All fields are fully dependent on the primary key.</td></tr> <tr> <td>Second Normal Form (2NF)</td><td>There are no repeating groups of attributes.</td></tr> <tr> <td>Third Normal Form (3NF)</td><td>There are no partial dependencies.</td></tr> </tbody> </table>	Normal Form	Definition	First Normal Form (1NF)	All fields are fully dependent on the primary key.	Second Normal Form (2NF)	There are no repeating groups of attributes.	Third Normal Form (3NF)	There are no partial dependencies.
Normal Form	Definition								
First Normal Form (1NF)	All fields are fully dependent on the primary key.								
Second Normal Form (2NF)	There are no repeating groups of attributes.								
Third Normal Form (3NF)	There are no partial dependencies.								

2(b)(iii)	1 mark each - CREATE TABLE start and end bracket - Bird ID as CHAR/VARCHAR - Name and size as VARCHAR/CHAR - Bird ID as primary key Example answer: <pre>CREATE TABLE BIRD_TYPE (BirdID CHAR(4) NOT NULL, Name VARCHAR(9), Size VARCHAR(6), PRIMARY KEY (BirdID));</pre>
2(b)(iv)	1 mark for each correctly completed space <pre>SELECT BIRD_TYPE.Size, COUNT(BIRD_TYPE.BirdID) AS NumberOfBirds FROM BIRD_TYPE, BIRD_SEEN WHERE BIRD_SEEN.PersonID = "J_123" AND BIRD_TYPE.BirdID = BIRD_SEEN.BirdID GROUP BY BIRD_TYPE.Size;</pre>