0 1 . 1 What is encryption?

[1 mark]

0 1 . 2 A sensitive message could be encrypted using either the Vernam cipher or the Caesar cipher.

Explain why the Vernam cipher is a better choice.

[2 marks]

The bit pattern 1010011 1001111 1001110 represents the string 'SON' in 7-bit ASCII.

The bit pattern 1000001 represents the character 'A' in 7-bit ASCII and other characters follow on from this in sequence. For example, the bit pattern 1001000 represents the character 'H'.

0 1 . 3 What bit pattern results from encrypting the string 'SON' using a Vernam cipher with the key 'HOG'?

You **must** show your working.

[3 marks]

0 2 . 1 The bit pattern `00111000` is the character code for the numeric character '`8`'

The bit pattern `00001000` represents the decimal number 8

Explain how a computer could convert the character code for '`8`' to the bit pattern for its corresponding decimal value.

[1 mark]

0 2 . 2 ASCII and Unicode are two common information coding systems.

Explain why Unicode was introduced as an alternative to ASCII.

[2 marks]

0 3 . 1 When transmitting data, the wireless network uses the following systems:

- Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with Request to Send/Clear to Send (RTS/CTS)
- Majority Voting.

Explain an advantage that majority voting has over using parity bits when transmitting data.

[1 mark]

___

___

___

0 4 . 1 A data transmission system transmits one byte of data, using the majority voting system for error correction.

Figure 1 shows the bit pattern that was received.

Figure 1

1	1	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	1	1	1	1	0	1	1

Shade **one** lozenge to indicate the byte of data that the receiver will assume was sent.

[1 mark]

A 10010011 ⬭

B 10011011 ⬭

C 10010111 ⬭

D 10011110 ⬭

0 5 . 1 A check digit can be used to detect errors when data are entered or transmitted.

Explain what a check digit is and outline how the check digit is generated.

[2 marks]

0 6 . 1 A data transmission system uses even parity. Data are transmitted in bytes, with each byte containing seven data bits and one parity bit.

Explain how the receiver will perform error detection on a received byte.

[2 marks]

0 7 . 1 ASCII is one character coding system.

Explain the term 'character code'.

[1 mark]

0 7 . 2 Explain why Unicode was introduced as an alternative to ASCII.

[2 marks]

Figure 3 shows a 7-bit ASCII character code. The character code is to be sent across a network using a parity system.

Figure 3

0	0	1	0	1	1	1

0 7 . 3 Describe how the parity bit would be generated for the character code in **Figure 3** using even parity.

[2 marks]

0 7 . 4 Write the parity bit below to complete the byte that will be sent using even parity.

[1 mark]

	0	0	1	0	1	1	1

0 7 . 5 The bit pattern 1000001 represents the character ‘A’ in 7-bit ASCII. Other characters follow on from this in sequence. For example, the bit pattern 1000100 represents the character ‘D’.

The bit pattern 1000100 1000001 1000010 represents ‘DAB’ in 7-bit ASCII.

What bit pattern results from encrypting the string ‘DAB’ using a Vernam cipher with the key ‘EGG’?

You **must** show your working.

[3 marks]

0 8 A data communication system uses asynchronous serial communication.

0 8 . 1 The ASCII code for the digit `'0'` is 48 in decimal. In ASCII, other digits follow on from this value in sequence.

The digit `'4'` is to be transmitted in ASCII using asynchronous serial transmission and **even parity**, with the parity bit stored in the most significant bit of the byte of data containing the ASCII code.

Complete **Figure 6** below to show a valid bit pattern for transmitting the digit `'4'`

[3 marks]

Figure 6

Stop Bit	Parity Bit								Start Bit

0 8 . 2 It is proposed that the communication system is modified so that:

- a majority voting system is used instead of the parity bit
- Unicode is used to encode the characters to be transmitted instead of ASCII.

Discuss the **improvements** that will occur in the communication system as a result of these changes and any **disadvantages** that will result from them.

[4 marks]

0 9 **Figure 2** shows some of the fields contained in a packet, transmitted on a computer network.

Figure 2

Destination Address	Source Address	Payload (data)	Checksum

0 9 . 1 Explain what the checksum is used for **and** outline how the checksum's value will be determined.

[2 marks]

Explain what the checksum is used for ____________________________

____

____

____

Outline how the checksum's value will be determined ____________________

____

____

____

1 0 For question parts **10.1** and **10.2** you should assume that memory locations and registers store **8-bit** values. These question parts use the AQA assembly language instruction set in **Table 3** on **page 23**.

Assembly language instructions can be used to perform masking, which allows the values of individual bits or groups of bits within a number to be isolated or set independently of the values of the other bits in the number.

For example, to isolate the values of the rightmost four bits of an 8-bit number, the number could be ANDed with the binary value `00001111`.

The assembly language instruction `AND R0, R1, #15` performs a bitwise logical AND operation between the value in register `R1` and the number `15` (equivalent to `00001111` in binary), storing the result in register `R0`.

1 0 . 1 In binary, show the result of applying the instruction `AND R0, R1, #15` when register `R1` contains the decimal value 70 which is 46 in hexadecimal.

[1 mark]

R1	0	1	0	0	0	1	1	0
15	0	0	0	0	1	1	1	1
R0								

1 0 . 2 In binary, show the result of applying the instruction `ORR R0, R1, #48` when register `R1` contains the decimal value 6 which is 6 in hexadecimal.

[1 mark]

R1	0	0	0	0	0	1	1	0
48	0	0	1	1	0	0	0	0
R0								

1 0 . 3 A computer program is required to display the value of the contents of a memory location that stores an 8-bit value. The value should be displayed on the screen of the computer in hexadecimal.

Part of the process required to do this is to convert the value stored in the memory location into the correct ASCII codes for each of the two digits that represent that value in hexadecimal.

For example, if the memory location contained:

1	0	0	1	1	1	1	0

which is 9E in hexadecimal, then the ASCII codes of the characters that need to be displayed are:

0	0	1	1	1	0	0	1

0	1	0	0	0	1	0	1

The first of these is the ASCII code of the character 9, the second is the ASCII code of the character E.

Write an assembly language program using the AQA assembly language instruction set that will load a value from memory location 100 and store the ASCII code of the first (lefthand) digit of the hexadecimal representation of this value in memory location 101 and the ASCII code of the second (righthand) digit of the hexadecimal representation of this value in memory location 102.

Your program should use masking and/or shifting to complete this task.

The ASCII codes of the hexadecimal digits are shown in **Table 2** and the AQA assembly language instruction set is in **Table 3** on **page 23**.

Table 2

	ASCII Code	
Digit	**Decimal**	**Binary**
0	48	0110000
1	49	0110001
2	50	0110010
3	51	0110011
4	52	0110100
5	53	0110101
6	54	0110110
7	55	0110111

	ASCII Code	
Digit	**Decimal**	**Binary**
8	56	0111000
9	57	0111001
A	65	1000001
B	66	1000010
C	67	1000011
D	68	1000100
E	69	1000101
F	70	1000110

[10 marks]

This table is included so that you can answer question parts 10.1, 10.2 and 10.3. Table 3 – Standard AQA assembly language instruction set

`LDR Rd, <memory ref>`	Load the value stored in the memory location specified by `<memory ref>` into register d
`STR Rd, <memory ref>`	Store the value that is in register d into the memory location specified by `<memory ref>`
`ADD Rd, Rn, <operand2>`	Add the value specified in `<operand2>` to the value in register n and store the result in register d
`SUB Rd, Rn, <operand2>`	Subtract the value specified by `<operand2>` from the value in register n and store the result in register d
`MOV Rd, <operand2>`	Copy the value specified by `<operand2>` into register d
`CMP Rn, <operand2>`	Compare the value stored in register n with the value specified by `<operand2>`
`B <label>`	Always branch to the instruction at position `<label>` in the program.

`B<condition> <label>`	Branch to the instruction at position `<label>` if the last comparison met the criterion specified by `<condition>`. Possible values for `<condition>` and their meanings are:
	`EQ: equal to` `NE: not equal to`
	`GT: greater than` `LT: less than`

`AND Rd, Rn, <operand2>`	Perform a bitwise logical AND operation between the value in register n and the value specified by `<operand2>` and store the result in register d
`ORR Rd, Rn, <operand2>`	Perform a bitwise logical OR operation between the value in register n and the value specified by `<operand2>` and store the result in register d
`EOR Rd, Rn, <operand2>`	Perform a bitwise logical XOR (exclusive or) operation between the value in register n and the value specified by `<operand2>` and store the result in register d
`MVN Rd, <operand2>`	Perform a bitwise logical NOT operation on the value specified by `<operand2>` and store the result in register d
`LSL Rd, Rn, <operand2>`	Logically shift left the value stored in register n by the number of bits specified by `<operand2>` and store the result in register d
`LSR Rd, Rn, <operand2>`	Logically shift right the value stored in register n by the number of bits specified by `<operand2>` and store the result in register d
`HALT`	Stops the execution of the program.

Labels: A label is placed in the code by writing an identifier followed by a colon (:). To refer to a label, the identifier of the label is placed after the branch instruction.

Interpretation of `<operand2>`

`<operand2>` can be interpreted in two different ways, depending on whether the first character is a # or an R:

- `#` – use the decimal value specified after the `#`, eg `#25` means use the decimal value 25
- `Rm` – use the value stored in register `m`, eg `R6` means use the value stored in register 6

The available general-purpose registers that the programmer can use are numbered 0–12

1 1 . 1 A data communication system uses **asynchronous** data transmission with even parity to send character codes that are encoded using 7-bit ASCII.

Figure 9 shows five binary patterns.

Figure 9

Pattern 1:	1	0	1	1	1	0	1	1	0	1
Pattern 2:	1	1	0	1	1	1	1	0	0	0
Pattern 3:	0	1	0	0	0	0	1	1	1	0
Pattern 4:	1	0	1	1	1	1	0	0	0	0
Pattern 5:	1	1	0	0	0	0	0	1	0	0

How many of the binary patterns in **Figure 9** could represent valid transmissions of a single character in this data communication system?

[1 mark]

---

1 1 . 2 An alternative data communication system uses **synchronous** data transmission.

Describe what synchronous data transmission is.

[1 mark]

---

1 1 . 3 Describe **one** limitation of the use of parity bits for managing errors.

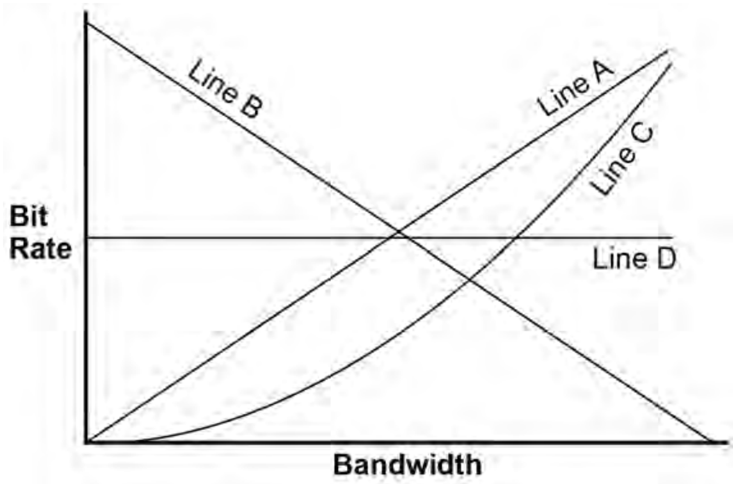
[1 mark]

---

1 1 . 4 Shade **one** lozenge to indicate which of the lines on the graph in **Figure 10** shows the correct relationship between the bandwidth and the bit rate of a communications medium.

[1 mark]

Figure 10



A Line A ⬭

B Line B ⬭

C Line C ⬭

D Line D ⬭