AQA Computer Science A-Level

4.5.5 Information coding systems

Past Paper Mark Schemes

Additional Specimen AS Paper 2

03	**1**		**Marks are for AO2 (apply)** The character has been received correctly; as there are an odd amounts of 1s;	2

03	**2**		**Mark is for AO2 (apply)** **0 0 1 1 1 0 0 1;**	1

03	**6**	**All marks AO2 (apply)** Not received correctly; As contains an odd number of 1s; **A.** odd number of 0s	2

03	**7**	**All marks AO1 (understanding)** To enable the representation of a greater range of characters; So that more languages // all (modern) languages can be represented (in one character set); Improved portability of documents in UNICODE // each character has a unique representation in UNICODE // in ASCII, representations could vary depending upon the code page used; **MAX 2**	2

June 2016 AS Paper 2

02	5	**Mark is for AO1 (understanding)** 9; **I**. Quotes around answer **I.** subscript 10 after the answer 9	1
02	6	**Marks are for AO1 (understanding)** The number of 1s (in the other 7 bits) has been counted // there are four 1s (in the 7 bits); there are an even number of 1s so the parity bit has been set to 0 (to keep the number of 1s even); **Alternative answer** The 7 data bits have been XORed; The result is a 0 so the parity bit has been set to 0 (so the result of XORing the 8 bits will be 0);	2

02	7	**1 mark is for AO1 (knowledge) and 1 mark for AO1 (understanding)** **AO1 knowledge – 1 mark:** Each bit is sent multiple times; **A.** A specified (odd) number greater than 2, instead of multiple **Marking guidance – to get this mark sent/sender must be clear** **AO1 understanding – 1 mark:** The receiver checks the bits it has received and if they are not all the same it assumes the one it received the most copies of is the correct value for the bit; **R.** receiver knows that the bit is correct **A.** receiver takes as correct (or similar) **Marking guidance – to get this mark received/receiver must be clear** **A. alternative answer using majority voting with a whole byte instead of individual bits** **AO1 knowledge – 1 mark:** The bit pattern (**R.** data) is sent multiple times; **A.** A specified number greater than 2, instead of multiple **Marking guidance – to get this mark sent/sender must be clear** **AO1 understanding – 1 mark:** The receiver checks the bit patterns (**R.** data) it has received and if they are not all the same it assumes the one it received the most copies of is the bit pattern (**R.** data) that was sent; **R.** receiver knows that the bit pattern is correct **A.** receiver takes as correct (or similar) **Marking guidance – to get this mark received/receiver must be clear**	2

June 2017 AS Paper 2

02	5	**Mark is for A02 (apply)** 1;	**1**

02	6	**1 mark for AO1 (knowledge) and 1 mark for AO1 (understanding)** Majority voting can correct as well as identify errors; due to the majority bits being taken as the correct value (and discounting/ignoring the minority bit); // Majority voting can detect multiple (bit) errors; as each triplet/odd set of bits represents one bit of data and can identify an error on that bit (not just an error within a byte); // Majority voting is more efficient at detecting errors (through multiple bits being corrupted); as parity bit system may miss errors if an even number of bits are corrupted; **A.** points made in reverse, identifying weaknesses of parity bits. **NE.** bits are sent multiple times as an explanation **R.** implication that receiver knows that the bit is correct **R.** data for bit/bits/byte **AO1 (knowledge) – 1 mark:** Identifying an advantage of majority voting over using parity bits. **AO1 (understanding) – 1 mark:** Explaining how the point is an advantage for majority voting. **Must award knowledge mark to award corresponding understanding mark.** **Max 2 marks**	**2**

June 2012 Comp 1

02	011 0010; **R.** If not 7 bits	**1**
03	1011 0000 **Mark as follows:** Correct data bits; Correct parity bit for the candidate's data bits; **R.** If not 8 bits	**2**

June 2013 Comp 1

10	Unicode uses more bits for each character // ASCII uses fewer bits for each character // Unicode can represent a wider range of characters // ASCII can represent a smaller range of characters // Unicode uses 16/32 bits, ASCII uses 7 bits (**A.** 8 bits);	**1**

11	**Role of sender:** Sender counts/checks the number of 1s in the bit pattern/value/data; and adds an extra bit to ensure even number of 1s; // Sender adds a 0 parity bit if there are an even number of 1s in the bit pattern/value/data; if odd number of 1s then a 1 parity bit is added; **Role of receiver:** Receiver counts/checks the number of 1s in the bit pattern/value/data received; if there are an odd number of 1s it identifies that an error has occurred; **A.** if even number of 1s it accepts the data received; **A.** if even number of 1s data is assumed to be correct; **A.** if odd number of 1s it requests that the data be resent; **R.** if even number of 1s, data is correct // Receiver regenerates parity bit from data received; compares generated parity bit with received parity bit – if different it identifies that an error has occurred; **A.** if the same it accepts the data received; **A.** if the same data is assumed to be correct; **A.** if different it requests that the data be resent; **R.** if the same, data is correct **A.** an odd number of errors (in the bit pattern received) will be detected; **Marking Guidance** **R.** Implication that sender or receiver are people **MAX 2** if role of sender <u>and</u> receiver not included in answer **R.** if mark point is about bit pattern being even/odd rather than the number of 1s being even/odd	**Max 4**

Specimen AS Paper 2

03	**1**		**Mark is for AO1 (knowledge)** A character code uses a unique number/code to represent each different character;	1

03	**2**		**Marks are for AO1 (understanding)** **1 mark:** b = 1100010; **1 mark:** e = 1100101;	2

03	**3**		**Mark is for AO2 (apply)** **1 mark:** 1000011;	1

Specimen Paper 2

07	**4**	**1 mark for AO1 (knowledge) and 2 marks for AO1 (understanding)** **AO1 (knowledge): 1 mark:** Checksum (is a number/value which) is calculated from // is a hash of the data in the packet (before it is transmitted); **AO1 (understanding): 1 mark:** Checksum recalculated when packet is received; **AO1 (understanding): 1 mark:** If checksum received in packet matches recalculated checksum then data received correctly // If	3

		checksum received in packet differs from recalculated checksum then data has been corrupted;	