Mark schemes

**Q1.**

(a)     **Mark is for AO2 (apply)**

39;

**A.** #39

**1**

(b)     **Mark is for AO1 (understanding)**

More compact when displayed;

Easier (for people) to understand/remember; **A.** read

Lower likelihood of an error when typing in data;

Saves (the programmer) time writing/typing in data;

**NE** takes up less space

**R.** if answer states that hexadecimal uses less memory/storage

**Max 1**

**1**

(c)     **Marks are for AO2 (apply)**

3 9/16

//

3.5625

**Mark as follows:**

**1 mark** for correct integer part (3)

**1 mark** for correct fractional part (9/16 or .5625)

**Alternative answer**

57/16;;

**2**

(d)     **Mark is for AO2 (apply)**

57;

**1**

(e)     **Mark is for AO1 (understanding)**

9;

**I.** Quotes around answer

**I.** subscript 10 after the answer 9

**1**

(f)     **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**

(g)     **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**

**[10]**

**Q2.**

**Marks are for AO1 (understanding)**

Parity bits can only detect errors not correct them // Majority voting can correct (most) errors that occur during transmission;

Majority voting can detect multiple (bit) errors;

Majority voting is more efficient at detecting errors;

Majority voting can (sometimes) detect an even number of errors;

**Max 1**

**[1]**

**Q3.**

(a)     1024 / 210;
**A** 100000000002 *(10 0’s)*

**1**

(b)     (i)      1111111111111111; (*16 1’s)*
**A** FFFF;
**A** 65,535 / 216-1;

**1**

(ii)     0000 0000 0010 0101; *accept if leading zeros not given*

**1**

(c)     (i)      0011 0011 1011 0111;;;  *accept 37 transposed*: 1011 0111 0011 0011;;;

*1 mark for parity bits - one mark for each correct character code*

f.t. for parity bits: if even number of 1’s in each byte ;

**3**

(ii)     Parity bit is set when character first generated;

(Parity bit is adjusted to make) number of 1’s /on-bits even;

Parity bit is regenerated / the number of 1’s is checked by receiver;

If parity bit does not match / if there are an odd number of 1’s an error has occurred;

**2**

**[8]**

**Q4.**

**Marks are for AO1 (understanding)**

The number of 1s is summed / counted; if the total is even, the parity bit is set to 0, otherwise it is set to 1 // if the total is odd, the parity bit is set to 1, otherwise it is set to 0 // the parity bit is set to ensure the total number of 1s is even;

The bits are XOR’d with each other; and the result is the parity bit;

**MAX 2**

**[2]**

**Q5.**

**Marks are for AO2 (apply)**

**1 mark** for identifying 1001000 1001111 1000111 as the binary representation of ‘HOG’

**1 mark** for final result being 21 bits long;

**R.** if result is the same as HOG (1001000 1001111 1000111) or SON (1010011 1001111 1001110)

**1 mark** for correct application of XOR;

  0011011 0000000 0001001

**A.** follow through mistakes

**[3]**

**Q6.**

**Marks are for AO2 (application)**

Showing that ‘EGG’ is represented by 1000101 1000111 1000111;

Providing a 21-bit answer that is not ‘DAB’ or ‘EGG’;

Correct answer (reached by applying XOR): 0000001 0000110 0000101;

**A.** Correct result of XORing 1000100 1000001 1000010 with an incorrect representation of ‘EGG’.

**[3]**

**Q7.**

(a)  **Marks for AO2 (apply)**

Identification of length (180 s/ 3 \* 60), sample resolution (16 bit) and sample rate (44,000 Hz) in working ; **A**. 44 (kHz) for sample rate but do not allow follow through.

Performing the correct calculation (3 \* 60 \* 16 \* 44,000 // 180 \* 16 \* 44,000) **or** showing correct intermediary value (126,720,000 bits / 1,584,000 Bytes) ;

**I**. Conversion

Final answer 15.84(MB) ;

**A**. to fewer significant places as long as 15.84 can be seen in working.

**Max 3**

(b)  **Marks are for AO1 (understanding)**

The ADC takes samples of the (analogue/continuous electrical) signal (at regular intervals); **R**. voltage for signal, soundwave, analogue data, sound, waveform for signal.

Samples are quantised // the amplitude/height of each sample is approximated to an integer value // the amplitude/height of samples are measured;

**A**. voltage for amplitude

**A**. digital, number, value for integer value

**A**. explanation of how the signal is quantised

Each sample is assigned a binary value/encoded as a binary value;

**R**. Digital value for binary value

**A**. Stored, converted so long as sample is stated

**3**

(c)  **1 mark for AO1 (knowledge) and 1 mark for AO1 (understanding)**

**Mark as follows:**

**AO1 (knowledge) – 1 mark:**

No/only redundant data is lost during the compression process (if using a lossless format);

Data is lost when storing using a lossy format;

**Max 1**

**AO1 (understanding)) – 1 mark:**

The song can be reproduced identically to the (recorded) original with no loss of quality (if using a lossless format);

If stored in a lossy format the quality may limit later editing possibilities;

**Max 1**

**A**. Recording will be of higher quality / quality of recording will be maintained.

**NE**. music will be of higher quality.

**[8]**