

1.4.1

DATA TYPES

TOPIC WISE EXAM QUESTIONS

ANSWERS

A-LEVEL

OCR

3	(a)	(i)	<ul style="list-style-type: none"> • BD 	1	Correct answer only
3	(a)	(ii)	<ul style="list-style-type: none"> • 2AF 	1	Correct answer only
3	(b)	(i)	<ul style="list-style-type: none"> • 1110 1011 	1	Correct answer only
3	(b)	(ii)	<ul style="list-style-type: none"> • 1001 0101 	1	Correct answer only
3	(b)	(iii)	<ul style="list-style-type: none"> • Calculations are more easily performed on two's complement • Two's complement allows for a (negligible) larger range of numbers to be stored // by example • No additional hardware is required in two's complement // Addition and subtraction are carried out using only an adder • Two's complement has only one representation for 0 	1	Accept the reverse of the MP
3	(c)		<ul style="list-style-type: none"> • -15.75 is 10000.01 (in fixed point two's complement) • Binary point moved 4 places left • Mantissa 1000 0010 • Exponent 0100 	4	Mantissa must be 8 bits, exponent must be 4 bits Accept alternative working for MP2 if appropriate. If mantissa AND exponent are correct with any working, 4 marks
3	(d)		<ul style="list-style-type: none"> • Exponent is -2 • Binary point moved 2 places left (0.001) // 0.5×2^{-2} • 0.125 // 1/8 (one eighth) 	3	MP2 is for correct working of whichever method is used. If answer is correct and working is shown, 3 marks
3	(e)		<ul style="list-style-type: none"> • precision / accuracy • range / size / magnitude 	2	

AS - Level

3	(a)		<ul style="list-style-type: none"> • List/mapping of characters (that can be understood by the hardware/software/computer) • Each character is given a unique binary/numeric code that is stored instead of the character. 	AO1.2 (2)	Allow a list of letters /symbols
3	(b)	(i)	<ul style="list-style-type: none"> • 102 (correct answer) • Suitable working out 	AO2.2 (2)	
		(ii)	<ul style="list-style-type: none"> • Hexadecimal values are shorter than binary as 4 bits/nibble can be represented by one hex character. • Hexadecimal values are faster / more reliable to communicate / enter / write down / read. 	AO2.2 (2)	Do not accept Hexadecimal is "easy" communicate/enter etc...
3	(c)		<ul style="list-style-type: none"> • Fixed point is 0110.01 • Mantissa becomes 0.11001 • Exponent of three needed (represented at 011) 	AO2.2 (3)	Final Answer 011001 011 Correct answer with any working gets full marks
3	(d)		<ul style="list-style-type: none"> • Allows for more accuracy/precision from the given number of bits • The representation of each binary value is unique 	AO1.1 (1)	

3	(a)	(i)	<ul style="list-style-type: none"> • 1011 0111 1110 	1	CAO															
		(ii)	<ul style="list-style-type: none"> • -149 	1	CAO															
		(iii)	0011 1001 <ul style="list-style-type: none"> • One mark for correct left hand nibble (CAO) • One mark for correct right hand nibble (CAO) • One mark for working clearly shown 	3	Working could include showing "borrowing" values from other columns or making the second number negative and adding. Answer must be 8 bits to achieve full marks (stated in question). No marks if only working is denary															
	(b)		<ul style="list-style-type: none"> • Exponent is -2 • Move decimal place 2 places <u>left</u> • Fill with 1s giving 1.1101 • Denary answer is $-0.1875 // \frac{-3}{16}$ 	4	Accept alternative method of -0.75×2^{-2} for BP2 and BP3. BP2 credited for -0.75 , BP3 credited for multiplying by 2^{-2} Correct answer with valid working gets full marks															
	(c)		<table border="1"> <thead> <tr> <th>Binary number</th> <th>Normalised</th> <th>Not normalised</th> </tr> </thead> <tbody> <tr> <td>010101 100</td> <td>x</td> <td></td> </tr> <tr> <td>110101 111</td> <td></td> <td>x</td> </tr> <tr> <td>011010 010</td> <td>x</td> <td></td> </tr> <tr> <td>101010 110</td> <td>x</td> <td></td> </tr> </tbody> </table>	Binary number	Normalised	Not normalised	010101 100	x		110101 111		x	011010 010	x		101010 110	x		4	One mark per row. No mark if both/neither box is ticked. Accept other marks that clearly indicate choice (e.g. X)
Binary number	Normalised	Not normalised																		
010101 100	x																			
110101 111		x																		
011010 010	x																			
101010 110	x																			
(e)	(i)		<pre>function countCapitals(text) // initialise counter to 0 capCount = 0 // loop through each character in the string passed in for x = 0 to text.length-1 c = text.subString(x, 1) // check if character is a capital</pre>	3	Accept alternative answers in high-level languages (e.g. <code>capCount ++</code> or <code>capCount +=1 / &&</code> for and for BP1) Accept <code>countCapitals = capCount</code> for BP3 (Returning via assigning to function identifier is used in VB / Pascal)															
			<pre>if asc(c) >= 65 and asc(c) <= 90 // if so, increment counter capCount = capCount + 1 endif next x return capCount endfunction</pre>		Accept and <code>asc(c) < 91</code> instead of <code><=90</code> Allow FT for returning the value they increment as a counter if it isn't <code>capCount</code>															
	(ii)		<ul style="list-style-type: none"> • Both (use binary) to represent characters // are character sets • The first 7/8 bits of Unicode is the same as ASCII (overlaps) 	1																
	(iii)		<ul style="list-style-type: none"> • ASCII has fewer characters (128/256) // Unicode has more characters • ASCII is 7/8 bits whereas Unicode can be larger 16/32 / can have variable sized characters • ASCII limited to Latin / English / European characters whereas Unicode can represent other symbols (e.g. Chinese/Cyrillic/Emojis) 	2	Only mark the 1st answer for each difference For BP2&3, must have both sides to get the mark															

4	(a)		1 mark for any of the following bullet points, e.g: <ul style="list-style-type: none"> Computers use binary logic for on/off or 1/0 Computer systems are based on switches/transistors Binary is high tolerance 	1 AO1.2 (1)	
4	(b)	(i)	<ul style="list-style-type: none"> 0110 0001 	1 AO2.2 (1)	Must get the whole answer right for the mark to be awarded.
4	(b)	(ii)	<ul style="list-style-type: none"> AB 	1 AO2.2 (1)	Must get the whole answer right for the mark to be awarded.
4	(b)	(iii)	<ul style="list-style-type: none"> 1001 1111 	1 AO2.2 (1)	Must get the whole answer right for the mark to be awarded.
4	(b)	(iv)	<ul style="list-style-type: none"> 1001 0001 	1 AO2.2 (1)	Must get the whole answer right for the mark to be awarded.
4	(c)		<ul style="list-style-type: none"> Can be easily used in binary arithmetic or Increased range of numbers available 	2 AO2.2 (2)	Award 0 or 2 marks
4	(d)	(i)	1 mark per bullet up to a maximum of 2 marks, e.g: <ul style="list-style-type: none"> (1) 0110 0100 suitable working out 	2 AO2.2 (2)	Award mark for bullet point one with or without the overflow stated.

2021

2	(a)	i	11100111	1 AO1.2	
		ii	0110 1111	1 AO1.2	
	(b)	i	<ul style="list-style-type: none"> Move decimal place 1 place right = 0.1101 to normalise Pad with 0s, <u>mantissa is 011010</u> Exponent is 4 / Subtract 1 from exponent... New exponent is 3 = <u>0011</u> 	4 AO1.2	
		ii	<ul style="list-style-type: none"> move dp 3 places right 0110.1 6.5 	3 AO1.2	Accept conversion of un-normalised number (will result in same output) Accept method of using 0.1101×2^3 FT required from part (i) if this is done incorrectly.
	(c)		<ul style="list-style-type: none"> Calculating exponent of 6 for first number/Calculating exponent of 4 for second number Moving binary point correct number of places in both mantissa to give 0110000.0 and 01010.0 Addition of 0110000.0 and 01010.0 to give 0111010.0 Movement of binary point six places to left to match original mantissa format and give value of 011101 Calculate exponent in original format to give value of 0110 	5 AO1.2	Alternative: <ul style="list-style-type: none"> Calculation of exponents (6 and 4) // difference between exponents of 2 Move <u>first mantissa 2 places left</u> // <u>second mantissa 2 places right</u> to match exponents ...giving $011.000 + 0.10100 \text{ (exp 4)}$ // $0.11000 + 0.00101 \text{ (exp 6)}$ Correct addition (answer of 011.1010 // 0.111010) Final answer normalised (011101 exp 6) <p>If exponents initially matched to 4, final bullet point should only be given if answer is normalised. If exponents matched to 6, answer will already be normalised. Ensure that answer is in format described (6 bits for mantissa 4 bits for exponent) to gain final bullet point.</p>

3	(a)	i	1000 1001	1 AO1.2	Correct answer only
3		ii	6D	1 AO1.2	Correct answer only
3		iii	AB	1 AO1.2	Correct answer only
3		iv	1010 0110 1 mark per nibble	2 AO1.2	Correct answer only
3	(b)		<ul style="list-style-type: none"> 10110.111 in fixed point Move binary point four places to the left (1.0110111) Gives mantissa of 10110111 therefore exponent is 4 = 0100 Forming final answer of 10110111 0100 	5 AO1.2	
3	(c)	(i)	<ul style="list-style-type: none"> 1001 1010 	2 AO1.2	1 mark per nibble, mark left to right
		(ii)	<ul style="list-style-type: none"> 1000 0000 OR 	2 AO2.2	

AS - Level

5	(a)		236	1 AO2.1 (1)	
	(b)		B3	1 AO2.1 (1)	
	(c)		-101	1 AO2.1 (1)	
	(d)		1 mark per bullet up to a maximum of 2 marks, e.g: <ul style="list-style-type: none"> 0101 1010 Suitable working out 	2 AO2.1 (2)	Allow any suitable method of working out, e.g. the carry method or using two's complement.

8	a	30 goes into 100 3 times/3	1 AO3.3
	b	integer	1 AO1.2
	c	Boolean	1 AO1.2
	d	String	1 AO1.2

10	a	5E	1 AO1.2	
	b	155 (1 mark for first two digits, 1 mark for final digit)	2 AO1.2	
	c	11010111 (1 Mark for the left most 1, 1 mark for the remaining 7 bits)	2 AO1.2	
	d	$\begin{array}{r} 1121 \\ 022022 \\ 04004001- \\ \underline{00101111} \\ \underline{00011010} \end{array}$ 1 mark for correct answer 1 mark for valid method	2 AO1.2	
	e	<ul style="list-style-type: none"> - Exponent is 3 - Mantissa becomes 0100.11 - Which is 4.75. (accept 4¾) (1 per -, max 3)	3 AO1.2	
	f	<ul style="list-style-type: none"> - In fixed point is 1010.11 - Mantissa becomes 1.01011 - Exponent of 3 / 11 - Giving answer of 101011 011 (1 per -, max 4)	4 AO1.2	For MP3 any number of leading 0s is valid (including none)

AS - Level

2	a	Hello – String 35 – Integer True - Boolean	3 (AO2.1)	
	b	00100011	1 (AO1.2)	
	c	<ul style="list-style-type: none"> - If A is 65, H is 72 - 72 in binary is 01001000 	2 (AO2.1)	If step one is incorrect allow FT for second mark
	d	<ul style="list-style-type: none"> - -2.625 in fixed point is 101.011 - Binary point moves two places left giving 1.01011/Mantissa is 101011 - Exponent of 2 is 0010 Answer 101011 0010 (1 per -, max 3)	3 (AO1.2)	Give full marks if correct answer. Allow FT

9	a	10101001 1111111	← Answer, 1 Mark ← Carry bits, 1 Mark	2 (AO1.2)	
	b	1 022 11001111 <u>00111001</u> 10010110	← Borrowed bits, 1 Mark ← Answer, 1 Mark	2 (AO1.2)	Allow 2 marks for any other valid method with working shown. If converted to denary and calculated, no marks.
	c	370F 1 Mark for the first two digits (i.e. 37) 1 Mark for the last two digit (i.e. 0F)		2 (AO1.2)	
	d	-Exponent is 2 -Mantissa becomes 010.01 -Value is 2.25 (1 Mark per -, Max 3)		3 (AO1.2)	
	e	01100 0000 1 Mark for mantissa, 1 mark for exponent. 10000 0100 1 Mark for mantissa, 1 mark for exponent.		4 (AO1.2)	
	f	10111001		1 (AO1.2)	cao
	g	11111111		1 (AO1.2)	cao

AS - Level

5	a	01001000		1 AO1.2 (1)	
	b	133		1 AO1.2 (1)	
	c	68		1 AO1.2 (1)	
	d	- Exponent is 4 - Move binary point 4 places to the right - Answer is: 9.25 (1 per -, max 3)		3 AO1.2 (3)	Accept any other method if working is shown
	e	- Computers use a character set/ASCII/ UNICODE - To map binary values to characters - Each character is represented by a unique value (1 per -, max 2)		2 AO1.1 (2)	

6	a	i	10111100	1 (AO1.2)	
		ii	BC	1 (AO1.2)	
	b	i	10101100	1 (AO1.2)	
		ii	11010100	1 (AO1.2)	
	c		Shift Right (1) Two Places (1)	2 (AO1.2)	Allow one mark for correct number of places but wrong direction.
	d		Binary point: shifted four places gives: 01001.0 (1) Binary point shifted two places gives: 010.010 (1) Subtraction carried out ... 01001.000 - 010.010 (1) ...'Borrowing' shown... (1) ...Answer: 0110.110 (1) Normalised to: Mantissa 011011 (1) Exponent 0011 (1)	6 (AO1.2)	Correct answer with clear binary subtraction/2's complement addition calculation gives full marks.

AS - Level

5	a		6F	1 AO2.1	
	b	i	11101101	1 AO2.1	
		ii	10010011	1 AO2.1	
	c		$\begin{array}{r} 212 \\ 002 \\ 11000011 \\ 01110010 \\ \hline 01010001 \end{array}$ <p>1 Mark for answer 1 Mark for showing working using appropriate binary method.</p>	2 AO2.1	NB some candidates represent carries with 10 as binary 2 rather than 2 Accept answer with missing leading zero.
	d		$1 \frac{5}{8}$ is 1.101 in fixed point (1 Mark) binary point needs moving one place giving 01101 001 One mark for Mantissa 01101	3 AO2.1	

5	a	00110111 (1 mark per nibble)	2	
			AO2.1	
	b	01101110 00000110 (1 mark for mantissa, 1 for exponent)	2	
			AO2.1	
	c	0110111 0110 (1 mark for mantissa, 1 for exponent)	2	
			AO2.1	
	d	<ul style="list-style-type: none"> The variable may need to store decimal numbers. To store very large/small values. 	1	Max. 1 mark
			AO1.2	

EXTRA

b		<ul style="list-style-type: none"> Calculate the first number as: 010.100 Calculate the second number as: 01.1000 Calculate the third number as: 100.010 Add the three together to get (1)000.010 Show carry bits. Discarding of leading one may be shown or implicit. Normalised result is: 010000 1111 	6	If candidate fails to discard the overflowing 1 they can still get marks 1–4.
		(1 per –)		

2	a	<ul style="list-style-type: none"> Shift left (1). 	1	For 1 mark.
	b	00101010 1 1 1	2	For 2 marks – award 1 mark for correct answer and 1 mark for carrying bits.
		Total	3	
3	a	00000000	1	
	b	11111111	1	
		Total	2	
4		<p>Exponent of first number is 3 (1) Making it 0100.1100 (1) Exponent of second number is 2 (1) Making it 010.01010 (1)</p> $ \begin{array}{r} \\ \\ \\ \hline \end{array} $ <p>Subtract numbers (1) Normalised is 01001110 0010 (1)</p>	5	Accept any sensible method (eg converting one number to have same exponent as other and subtracting) with correct answer for full marks.

**If you found this
useful, drop a follow
to help me out!**

THANK YOU!

GCST