CAMBRIDGE IGCSE (0478)

Module C: Data representation

Workbooklet mark scheme · 11 parts · 222 marks

Contents

Part	Topic	Marks
C1	Why computers use binary: mark scheme	20
C2	Converting denary and binary: mark scheme	20
C3	Hexadecimal and why we use it: mark scheme	20
C4	Converting binary and hexadecimal: mark scheme	20
C5	Binary addition and overflow: mark scheme	21
C6	Logical binary shifts: mark scheme	19
C7	Two's complement: mark scheme	22
C8	Text and character sets: ASCII and Unicode: mark scheme	20
C9	Representing images: mark scheme	20
C10	Representing sound: mark scheme	20
C11	Compression, lossy and lossless: mark scheme	20
	Total	**222**

MODULE C DATA REPRESENTATION · C1

Why computers use binary: mark scheme

20 marks · Cambridge 0478 syllabus 1.1.1

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.1.1 / 1.1.2(a)	(base) 2	**1**	Accept "two". Do not accept 10 / 16.
1(b) 0478 1.1.1	bit	**1**	Do not accept byte / nibble.
1(c) 0478 1.1.1	logic gates // switches // transistors	**1**	Do not accept "the CPU" / "the processor" alone.
2(a) 0478 1.1.1	Sound converted: True Digits 1 and 2: False Registers: True Only numbers in binary: False	**4**	1 mark per correct row. More than one tick in a row: 0 for that row. Row 2: binary uses 0 and 1. Row 4: all forms of data, including text, are represented in binary.
3(a) 0478 1.1.1	Any three from: images // photographs sound // audio numbers // (temperature) readings text // characters	**3**	1 mark each, max 3. Accept program instructions. Two answers of the same type (e.g. "photos" and "pictures") score once.
3(b) 0478 1.1.1	register(s)	**1**	Do not accept RAM / memory / hard disk.
4(a) 0478 1.1.1	4 bits: 16 8 bits: 256 16 bits: 65 536	**3**	1 mark per row. Working: $2^4 = 16$; $2^8 = 256$; $2^{16} = 256 \times 256 = 65\,536$. Do not accept 15 / 255 / 65 535 (largest values, not number of patterns).
4(b) 0478 1.1.1 / 1.3.1	nibble	**1**	Do not accept byte / bit.

QUESTION	ANSWER	MARKS	GUIDANCE
5(a) 0478 1.1.1	Any two from: computers are made of / use logic gates // switches // transistors ... which only have two states // on and off two states can be represented by 1 and 0 data is stored in registers (as 1s and 0s)	**2**	max 2. "A computer can only understand binary" alone: 0 (too vague).
6(a) 0478 1.1.1	Any three from: the statement does not give a reason // is too vague data is processed using logic gates // switches // transistors each has only two states // on and off the two states are represented by 1 and 0 // binary so all forms of data (text, images, sound) must be converted to binary // stored in registers as binary	**3**	max 3. Repeating the statement in other words scores 0.

Total: 20 marks

MODULE C DATA REPRESENTATION · C2

Converting denary and binary: mark scheme

20 marks · Cambridge 0478 syllabus 1.1.2

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. 16-bit place values: 32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1, most significant bit on the left. A binary answer with the bits reversed is never credited. Binary answers must have the number of bits asked for. Where "Show your working" is asked, a correct answer with no working still gets full marks.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.1.2(a)	2 // base 2	**1**	Correct answer only.
1(b) 0478 1.1.2(a)	10 // base 10	**1**	Correct answer only.
2(a) 0478 1.1.2(b) (i)	115	**1**	64 + 32 + 16 + 2 + 1 = 115.
2(b) 0478 1.1.2(b) (i)	1024 + 256 + 128 + 16 + 4 + 2 // correct place values identified for each 1 1430	**2**	1 mark working, 1 mark answer. Correct answer with no working = 2.
3(a) 0478 1.1.2(b) (i)	10110101	**1**	128 + 32 + 16 + 4 + 1 = 181.
3(b) 0478 1.1.2(b) (i)	512 + 256 + 128 + 64 + 32 + 8 // 1000 − 512 = 488, − 256 = 232, − 128 = 104, − 64 = 40, − 32 = 8, − 8 = 0 // repeated division by 2 with remainders 0000001111101000	**2**	1 mark working, 1 mark answer. Correct answer with no working = 2. Answer must be 16 bits: 1111101000 = working mark only.
3(c) 0478 1.1.2(b) (i)	32768 + 4096 + 2048 + 1024 + 64 // 40000 − 32768 = 7232, − 4096 = 3136, − 2048 = 1088, − 1024 = 64, − 64 = 0 1001110001000000	**2**	1 mark working, 1 mark answer. Correct answer with no working = 2.
4(a) 0478 1.1.2(b) (i)	Shelf number: 00110100 Parcel count: 129 Distance travelled: 0001001110001000 Item code: 34817	**4**	1 mark per box. 32 + 16 + 4 = 52; 128 + 1 = 129; 4096 + 512 + 256 + 128 + 8 = 5000; 32768 + 2048 + 1 = 34817. Binary must be 8 bits / 16 bits as stated.

QUESTION	ANSWER	MARKS	GUIDANCE
5(a) 0478 1.1.2(b) (i)	Any three from: the bits are in reverse order // written backwards the place values were used from smallest to largest (from the left) // the most significant bit / 128 should be on the left 10110000 is actually 176 // 13 is 8 + 4 + 1 00001101	**3**	max 3. Third mark for the correct answer requires 00001101 exactly; do not accept 1101.
6(a) 0478 1.1.2(b) (i)	65535	**1**	1111111111111111. Do not accept 65536.
6(b) 0478 1.1.2(b) (i)	Any two from: the largest number in 16 bits is 65535 70000 is greater than 65535 // too large for 16 bits it would need more than 16 bits // 17 bits	**2**	max 2. 70000 in binary needs 17 bits.

Total: 20 marks

MODULE C DATA REPRESENTATION · C3

Hexadecimal and why we use it: mark scheme

20 marks · Cambridge 0478 syllabus 1.1.2(b)(ii), 1.1.3

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Ignore case in hex; ignore leading 0s. A correct final answer with no working gets full marks, even where working is asked for; the working mark credits a correct method when the final answer is wrong.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.1.2(a)	16 // base 16	**1**	Correct answer only.
1(b) 0478 1.1.2(a)	3AF	**1**	G and H are not hex digits; values are integers only (no point). More than one tick = 0.
2(a) 0478 1.1.2(b)(ii)	5C	**1**	92 ÷ 16 = 5 remainder 12; 12 = C. Do not accept C5 or 512.
2(b) 0478 1.1.2(b)(ii)	400 = 1 × 256 + 9 × 16 + 0 // 400 ÷ 16 = 25 r 0, 25 ÷ 16 = 1 r 9 190	**2**	1 mark working, 1 mark answer. Do not accept 19 (trailing 0 dropped) or 091.
2(c) 0478 1.1.2(b)(ii)	2750 = 10 × 256 (2560) + 11 × 16 (176) + 14 // 2750 ÷ 16 = 171 r 14, 171 ÷ 16 = 10 r 11 ABE	**2**	1 mark working, 1 mark answer. 2560 + 176 + 14 = 2750. Do not accept EBA (reversed) or 101114.
3(a) 0478 1.1.2(b)(ii)	127	**1**	7 × 16 + 15 = 127.
3(b) 0478 1.1.2(b)(ii)	1 × 256 + 10 × 16 + 0 // 256 + 160 416	**2**	1 mark working, 1 mark answer. Do not accept 26 (1A only) or 1100.
3(c) 0478 1.1.2(b)(ii)	3 × 4096 + 12 × 256 + 2 × 16 + 13 // 12 288 + 3072 + 32 + 13 15 405	**2**	1 mark working (place values 4096, 256, 16, 1 used with the correct digit values), 1 mark answer.
4(a) 0478 1.1.2(b)(ii)	200	**1**	12 × 16 + 8 = 200.

QUESTION	ANSWER	MARKS	GUIDANCE
4(b) 0478 1.1.3	Any two from: MAC addresses IPv6 addresses error codes / error messages memory addresses // memory dumps assembly language / machine code (listings) // debugging URLs (special characters)	**2**	max 2. Colour codes = 0 (given in the question). Do not accept IPv4 addresses (written in denary).
5(a) 0478 1.1.3	4 (bits) // a nibble	**1**	48 ÷ 12 = 4.
5(b) 0478 1.1.2(b) (ii)	65 535	**1**	FFFF = 15 × 4096 + 15 × 256 + 15 × 16 + 15 = 65 535 (16 bits). Do not accept 65 536.
6(a) 0478 1.1.3	Any three from: hex is a shorter representation of binary // uses fewer digits one hex digit represents 4 bits / a nibble // e.g. 2 hex digits for 8 bits (so) easier for humans to read / understand / remember (so) fewer errors when reading, writing or typing it easy / quick to convert between hex and binary the computer still stores and processes the data in binary	**3**	max 3. "Easier" alone is not enough: needs a reason such as shorter. Do not accept: hex uses less storage / memory; the computer uses / understands hex; hex is faster to process.

Total: 20 marks

MODULE C DATA REPRESENTATION · C4

Converting binary and hexadecimal: mark scheme

20 marks · Cambridge 0478 syllabus 1.1.2(b)(iii)

How to use. One mark per creditworthy point up to the maximum shown. **//** separates alternative answers. Words in (brackets) are not needed for the mark. Working is shown for the teacher; a correct final answer with no working gets full marks (marks here are for the answer digits, so none are withheld). Never credit digits in the wrong order, a reversed bit pattern, or a denary value (e.g. 15) written in place of a hex digit.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.1.2(b)(iii)	4 // a nibble	**1**	Correct answer only.
1(b) 0478 1.1.2(b)(iii)	4	**1**	16 ÷ 4 = 4.
2(a) 0478 1.1.2(b)(iii)	C (1100 = 8 + 4 = 12) A (1010 = 8 + 2 = 10)	**2**	Answer CA. 1 mark per correct digit in the correct position. AC = 0. 1210 = 0.
2(b) 0478 1.1.2(b)(iii)	3 (0011 = 2 + 1) E (1110 = 8 + 4 + 2 = 14) 8 (1000) 5 (0101 = 4 + 1)	**4**	Answer 3E85. 1 mark per correct digit in the correct position. 14 written for E does not earn the E mark. Digits in reverse order (58E3) = 0.
3(a) 0478 1.1.2(b)(iii)	1001 (9 = 8 + 1) 1111 (F = 15)	**2**	Answer 1001 1111 // 10011111. 1 mark per correct nibble in the correct position.
3(b) 0478 1.1.2(b)(iii)	0010 (2) 1011 (B = 11 = 8 + 2 + 1) 0000 (0) 1101 (D = 13 = 8 + 4 + 1)	**4**	Answer 0010 1011 0000 1101 // 0010101100001101. 1 mark per correct 4-bit nibble in the correct position: "10" for the first nibble, or a missing 0000 nibble, does not earn that mark. Reversed pattern = 0.
4(a) 0478 1.1.2(b)(iii)	Left out the (second) nibble 0000 // did not write 0 for the group of four zeros	**1**	Accept: missed out a digit for 0000. Do not accept "wrong conversion" alone.
4(b) 0478 1.1.2(b)(iii)	D0A3	**1**	1101 = 13 = D; 0000 = 0; 1010 = 10 = A; 0011 = 3.

QUESTION	ANSWER	MARKS	GUIDANCE
5(a) 0478 1.1.2(b) (iii)	0110 1111 → 6F B4 → 1011 0100 0001 0010 1100 1110 → 12CE 8E07 → 1000 1110 0000 0111	**4**	1 mark per row. 6F: 0110 = 6, 1111 = 15 = F. B4: B = 11 = 1011, 4 = 0100. 12CE: 0001 = 1, 0010 = 2, 1100 = 12 = C, 1110 = 14 = E. 8E07: 8 = 1000, E = 14 = 1110, 0 = 0000, 7 = 0111. Binary must have the number of bits shown (8 // 16). Do not accept 121214 for 12CE.

Total: 20 marks

MODULE C DATA REPRESENTATION · C5

Binary addition and overflow: mark scheme

21 marks · Cambridge 0478 syllabus 1.1.4

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Binary addition: one mark for each correct nibble (max 2) plus one mark for a correct method (carries shown). Converting to denary and back does not earn the method mark. These questions say "You must show all your working", so a correct answer with no carries earns the nibble marks only. In the grids, each carry is shown under the column it is carried **into** (the dashed column is the overflow bit); carries written above the numbers are equally fine.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.1.4(b)	255	**1**	Correct answer only. Do not accept 256.
1(b) 0478 1.1.4(a)	One mark for each of: method: carries shown correctly (into the 2, 4, 8, 16, 32, 64 and 128 columns) 1000 (most significant nibble) 0100 (least significant nibble)	**3**	Answer 10000100. Nibble marks only for the correct nibble in the correct place. Check: 103 + 29 = 132.
2(a) 0478 1.1.4(a) (b)	One mark for each of: method: carries shown correctly (into the 4, 16, 32, 64 and 128 columns) 0001 (most significant nibble) 0101 (least significant nibble) overflow bit shown // final carry of 1 out of the 128 column (shown as a 9th bit)	**4**	8-bit answer 00010101; accept 100010101 (9 bits) for the nibble marks. Check: 186 + 91 = 277; 277 − 256 = 21 = 00010101.

	128	64	32	16	8	4	2	1	
	0	1	1	0	0	1	1	1	
	0	0	0	1	1	1	0	1	+
	1	0	0	0	0	1	0	0	Answer
	1	1	1	1	1	1	1		Carries

	128	64	32	16	8	4	2	1	
	1	0	1	1	1	0	1	0	
	0	1	0	1	1	0	1	1	+
1	0	0	0	1	0	1	0	1	Answer
1	1	1	1	1		1			Carries

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>2(b)
0478 1.1.4(b)</td><td>Any two from:
the result (277) is greater than 255
... which is the largest value an 8-bit register can store
the result needs 9 bits // there is a carry out of the most significant bit / 128 column
there is no 9th bit / space in the register for the carry // the extra bit cannot be stored</td><td>2</td><td>max 2. "The number is too big" alone is too vague.</td></tr>
<tr><td>3(a)
0478 1.1.4(b)</td><td>Greater than 255 in 8-bit register: True
Both MSBs 0 can overflow: False
16-bit register stores 300: True
Correct result still stored: False</td><td>4</td><td>One mark per correct row. More than one tick in a row = 0 for that row. Row 2: the largest sum is 127 + 127 = 254. Row 3: the 16-bit maximum is 65 535.</td></tr>
<tr><td>4(a)
0478 1.1.4(b)</td><td>65 535</td><td>1</td><td>Accept 65535. Do not accept 65 536.</td></tr>
<tr><td>4(b)
0478 1.1.4(b)</td><td>Any two from:
an overflow error occurs when a value outside / greater than the limit should be returned / stored
e.g. a result greater than 255 in an 8-bit register // greater than 65 535 in a 16-bit register
the value needs more bits than are available // there are not enough bits to store it
(so) the stored result is incorrect</td><td>2</td><td>max 2.</td></tr>
<tr><td>5(a)
0478 1.1.4(a)</td><td>One mark for each of:
method: carries shown correctly (into the 2, 4, 8, 16, 32, 64 and 128 columns)
1010 (most significant nibble)
0000 (least significant nibble)
<table><tr><th></th><th>128</th><th>64</th><th>32</th><th>16</th><th>8</th><th>4</th><th>2</th><th>1</th><th></th></tr><tr><td></td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td></td></tr><tr><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>+</td></tr><tr><td></td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Answer</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td>Carries</td></tr></table></td><td>3</td><td>Answer 10100000. Check: 117 + 43 = 160.</td></tr>
<tr><td>5(b)
0478 1.1.2(b)(i)</td><td>160</td><td>1</td><td>Follow through from the candidate's 8-bit answer to 5(a).</td></tr>
</table>

Total: 21 marks

MODULE C DATA REPRESENTATION · C6

Logical binary shifts: mark scheme

19 marks · Cambridge 0478 syllabus 1.1.5

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Every shift was checked by program. A correct final answer with no working earns full marks. Bits shifted from the end of the register are lost and zeros are shifted in at the opposite end.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.1.5	10101000	**1**	Working: 00010101 (21) → 10101000 (168 = 21 × 8); three 0s lost on the left, three 0s in on the right. Do not accept 00000010 (shifted right).
1(b) 0478 1.1.5	00110011	**1**	Working: 11001100 (204) → 00110011 (51 = 204 ÷ 4); two 0s lost on the right. Do not accept 00110000 (shifted left).
2(a) 0478 1.1.5	Multiplied by 4	**1**	2+ ticks = 0.
2(b) 0478 1.1.5	Divided by 16	**1**	2+ ticks = 0.
3(a) 0478 1.1.5	Row 1: 01011010 Row 2: 00000011 Row 3: 10110000 Row 4: 01001011	**4**	1 mark per row. Checks: 45 → 90; 96 → 3 (96 ÷ 32); 11 → 176 (11 × 16); 150 → 75. Each must be 8 bits in the correct order.
4(a) 0478 1.1.5	10100100	**1**	The most significant 1 is shifted off the end and lost; a 0 moves in on the right.
4(b) 0478 1.1.5	164	**1**	128 + 32 + 4 = 164. Follow through from (a).
4(c) 0478 1.1.5	The most significant bit / the 1 in the 128 column is shifted off the end / lost The register only holds 8 bits // 420 needs 9 bits // 420 is larger than 255 (the largest 8-bit value)	**2**	Any 2 from. Accept 420 − 256 = 164 as evidence for the first mark.
5(a) 0478 1.1.5	111	**1**	The three least significant bits (the 4, 2 and 1 columns). Accept "three 1s".

QUESTION	ANSWER	MARKS	GUIDANCE
5(b) 0478 1.1.5	The least significant bits / the 1s at the right end are lost (The register holds whole numbers only, so) the fraction / 0.875 / remainder is lost // the result is rounded down	**2**	Any 2 from.
6(a) 0478 1.1.5	0 / zeros	**1**	Correct answer only.
6(b) 0478 1.1.5	(Logical) right shift 5 places	**2**	$32 = 2^5$. 1 mark direction, 1 mark places.
6(c) 0478 1.1.5	They are lost	**1**	Accept "discarded". Do not accept "moved to the other end" (that is a rotation, not a logical shift).

Total: 19 marks

MODULE C DATA REPRESENTATION · C7

Two's complement: mark scheme

22 marks · Cambridge 0478 syllabus 1.1.6

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. All patterns are 8-bit two's complement (place values −128, 64, 32, 16, 8, 4, 2, 1). A binary answer must have exactly 8 bits in the correct order. A correct final answer with no working gets full marks, including where the question says "Show your working".

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.1.6	−128	**1**	Do not accept 128 (the minus sign is needed).
1(b) 0478 1.1.6	Smallest: −128 Largest: 127 // +127	**2**	1 mark each. Do not accept 128 or 255 as largest, or −127 as smallest.
2(a) 0478 1.1.6	90	**1**	64 + 16 + 8 + 2 = 90.
2(b) 0478 1.1.6	−128 + 32 + 16 + 4 + 1 // −128 + 53 −75	**2**	1 mark working (MSB taken as −128 with the correct positive headings), 1 mark answer. Correct answer with no working = 2. 181 = 0. 75 (no minus sign) = 0.
2(c) 0478 1.1.6	−1	**1**	−128 + 127 = −1. Do not accept 255.
3(a) 0478 1.1.6	01001101	**1**	64 + 8 + 4 + 1 = 77. Must be 8 bits (the leading 0 is needed).
3(b) 0478 1.1.6	Working: 77 = 01001101, flipped to 10110010, then + 1 // −128 + 51 (51 = 32 + 16 + 2 + 1) 10110011	**2**	1 mark working, 1 mark answer. Correct answer alone = 2. 10110010 (1 not added; this is −78) = 1. 11001101 (first bit switched on; this is −51) = 0.
3(c) 0478 1.1.6	Working: 19 = 00010011, flipped to 11101100, then + 1 // −128 + 109 (109 = 64 + 32 + 8 + 4 + 1) 11101101	**2**	1 mark working, 1 mark answer. Correct answer alone = 2. 11101100 (1 not added; this is −20) = 1. 10010011 (first bit switched on) = 0.

QUESTION	ANSWER	MARKS	GUIDANCE
4(a) 0478 1.1.6	Any two from: In two's complement the most significant bit is worth −128 (not +128) // 246 is the value read as binary with no sign The MSB is 1 so the reading is negative The correct reading is −10 (°C)	**2**	max 2. Working: −128 + 64 + 32 + 16 + 4 + 2 = −10.
4(b) 0478 1.1.6	Any two from: The largest value that can be stored is 127 150 is greater than 127 // 150 is out of range The MSB is worth −128, so only 7 bits have positive place values // a 1 in the MSB would make the value negative	**2**	max 2. Do not credit "8 bits store up to 255".
5(a) 0478 1.1.6	Row 1: True Row 2: False Row 3: True	**3**	1 mark per correct row. More than one tick in a row = 0 for that row. Row 2: 10000000 = −128; 0 is 00000000. Row 3: positive integers 0 to 127 have MSB 0 and the same pattern.
6(a) 0478 1.1.6	Any three from: Write the positive value (of the number) in 8-bit binary Flip / invert every bit // change 1s to 0s and 0s to 1s (then) add 1 (to the flipped pattern) (using binary addition) carrying where needed // check that the MSB is 1 Alternative: put a 1 in the −128 column // MSB = 1 Alternative: work out (number + 128) and write it in the remaining 7 bits	**3**	max 3. Steps must be in the correct order: adding 1 before flipping = max 1. Do not credit "put a 1 in front of the positive number" on its own.

Total: 22 marks

MODULE C DATA REPRESENTATION · C8

Text and character sets: ASCII and Unicode: mark scheme

20 marks · Cambridge 0478 syllabus 1.2.1

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Every code, pattern and size was checked by program (Python). Codes are given, so none need remembering. A correct final answer with no working earns full marks, including where working is asked for. Never credit a reversed bit pattern.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.2.1	Computers can only process / store binary (0s and 1s) // computers are made of switches / logic gates that are on or off Each character is given a unique binary code // a character set is used to convert each character to binary	**2**	Any 2 from the points listed, max 2.
1(b) 0478 1.2.1	All the characters (and symbols) a computer can represent, each with its own unique (binary) code	**1**	Do not accept "the alphabet" alone.
2(a) 0478 1.2.1	68	**1**	Working: A 65, B 66, C 67, D 68.
2(b) 0478 1.2.1	01000011	**1**	Working: C = 67 = 64 + 2 + 1. Must be 8 bits, as the question asks. Do not accept 11000010 (reversed).
2(c) 0478 1.2.1	D: 01000100 A: 01000001 B: 01000010	**3**	Working: D = 68 = 64 + 4; A = 65 = 64 + 1; B = 66 = 64 + 2. 1 mark per correct row. Each must be 8 bits in the correct order.
3(a) 0478 1.2.1	01100100 = 100 = d 01101001 = 105 = i 01100111 = 103 = g	**3**	Word: **dig**. Working: 64 + 32 + 4 = 100; 64 + 32 + 8 + 1 = 105; 64 + 32 + 4 + 2 + 1 = 103; counting on from a = 97. 1 mark per correct character in the correct position. Upper case (DIG): max 2.
4(a) 0478 1.2.1	Unicode can represent more characters // a greater range of characters than ASCII Unicode includes other languages / alphabets // emojis; ASCII is mainly English characters Unicode uses more bits per character // ASCII uses 7 or 8 bits, Unicode up to 32	**2**	Any 2 from the points listed, max 2. Each difference must be clear about which is which. Do not accept "Unicode uses fewer bits".

QUESTION	ANSWER	MARKS	GUIDANCE
4(b) 0478 1.2.1	ASCII has a limited number of characters // 256 (or 128) characters ASCII cannot represent the characters of many (non-English) languages ASCII has no codes for emojis Unicode uses more bits per character … so it has a code for many more characters // languages and emojis So all users' messages / emojis display correctly	**3**	Any 3 from the points listed, max 3. Not linked to the languages or emojis in the scenario: max 2.
5(a) 0478 1.2.1	40 × 8 = 320 bits 320 ÷ 8 = 40 bytes	**2**	Correct answer 40 (bytes) with no working: 2. Accept 40 × 1 byte = 40.
5(b) 0478 1.2.1	160 (bytes)	**1**	Working: 40 × 32 = 1280 bits; 1280 ÷ 8 = 160 bytes // 40 × 4 bytes = 160.
5(c) 0478 1.2.1	The file is larger // four times larger	**1**	Do not accept "more characters" alone.

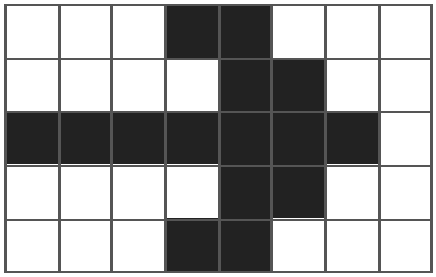
Total: 20 marks

MODULE C DATA REPRESENTATION · C9

Representing images: mark scheme

20 marks · Cambridge 0478 syllabus 1.2.3

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Every bit pattern and power of 2 was checked by program. A correct final answer with no working earns full marks.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.2.3	The smallest (single) element / dot / square of colour in a (bitmap) image	**1**	Do not accept "a bit".
1(b) 0478 1.2.3	The number of pixels in the image // width × height (in pixels)	**1**	Accept number of pixels per unit (e.g. per inch). Do not accept "quality" on its own.
1(c) 0478 1.2.3	The number of bits used to represent each colour // each pixel	**1**	Accept "bits per pixel". Do not accept "number of colours" on its own.
2(a) 0478 1.2.3	All 5 rows correct (see grid below) At least 3 rows correct 	**2**	Rows: 00011000, 00001100, 11111110, 00001100, 00011000 (an arrow pointing right; checked by program). 2 marks all five rows; 1 mark three or four rows. Inverted colours = 0.
2(b) 0478 1.2.3	4 (bits)	**1**	$2^4 = 16$. Do not accept 16.
2(c) 0478 1.2.3	256	**1**	$2^8 = 256$. Do not accept 8 or 16.
3(a) 0478 1.2.3	3 bits: 8 5 bits: 32 128 colours: 7 (bits) 10 bits: 1024	**4**	1 mark per correct cell. $2^3 = 8$, $2^5 = 32$, $2^7 = 128$, $2^{10} = 1024$ (checked by program). Do not accept 6, 10, 64 or 20 (doubling the bits instead of using powers of 2).

QUESTION	ANSWER	MARKS	GUIDANCE
4(a) 0478 1.2.3	7 (bits)	**1**	2^6 = 64 is too few; 2^7 = 128, which is at least 100. Do not accept 6 or 100.
5(a) 0478 1.2.3	True False True	**3**	1 mark per row. Row 2: colour depth changes the bits per pixel, not the number of pixels.
6(a) 0478 1.2.3	More pixels (in the image) More detail // sharper // less pixelated // higher quality Larger file size // more bits to store	**2**	Any 2 from. Do not accept "more colours".
6(b) 0478 1.2.3	Fewer bits per pixel Fewer colours can be shown // 256 colours instead of over 16 million Colours less accurate // shading less smooth // lower quality Smaller file size Resolution // number of pixels is unchanged Less storage space // less bandwidth // faster to load	**3**	Any 3 from. 2^8 = 256; 2^{24} = 16 777 216. Do not accept "the image becomes blockier / pixelated" (that is resolution). (Calculating the file size is covered in B8.)

Total: 20 marks

MODULE C DATA REPRESENTATION · C10

Representing sound: mark scheme

20 marks · Cambridge 0478 syllabus 1.2.2

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Cambridge 0478 uses "sample resolution"; accept "bit depth". Accept "quality" for accuracy.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.2.2	The number of samples taken in a second	**1**	Accept per second. Do not accept the frequency of the sound wave.
1(b) 0478 1.2.2	The number of bits per sample // used to store each sample	**1**	Accept bit depth. Do not accept number of bits in the file.
2(a) 0478 1.2.2	True False True False	**4**	1 mark per row, in this order. Two ticks in a row = 0 for that row.
3(a) 0478 1.2.2	32	**1**	Working: $2^5 = 32$. Do not accept 25 or 10.
3(b) 0478 1.2.2	More bits per sample so more (different) values/levels can be stored // each sample is closer to the actual amplitude/height of the wave	**1**	16 bits gives $2^{16} = 65{,}536$ values against 32.
4(a) 0478 1.2.2	Sample rate increased: more accurate, file size increases Sample resolution decreased: less accurate, file size decreases Sample resolution increased: more accurate, file size increases Sample rate decreased: less accurate, file size decreases	**4**	1 mark per row; both ticks must be correct.

QUESTION	ANSWER	MARKS	GUIDANCE
5(a) 0478 1.2.2	Any five from: sound (wave) is analogue the sound wave is sampled the amplitude/height of the wave is measured at set/regular time intervals each sample is converted to / stored in binary sample rate is the number of samples taken per second a higher sample rate gives a more accurate recording // larger file sample resolution is the number of bits per sample a higher sample resolution gives a more accurate recording // larger file	**5**	max 5. Credit each effect once for sample rate and once for sample resolution. Accept "closer to the original sound" for more accurate. Do not credit "the frequency is measured".
6(a) 0478 1.2.2	Reduce the sample rate // reduce the sample resolution	**1**	Must state the direction. Do not accept compression.
6(b) 0478 1.2.2	The tracks will be less accurate // lower quality // sound less like the original	**1**	Must follow from the change in (a).
6(c) 0478 1.2.2	The length / duration of the track	**1**	Accept number of seconds. File-size calculations (syllabus 1.3.2) are practised in B8.

Total: 20 marks

MODULE C DATA REPRESENTATION · C11

Compression, lossy and lossless: mark scheme

20 marks · Cambridge 0478 syllabus 1.3.3, 1.3.4

How to use. Any N from the points listed, up to the max shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Every RLE answer was checked by program. A correct final answer with no working earns full marks. Credit the type or method the syllabus names, not a file format.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.3.3	Reduces the file size Takes up less storage space Uses less bandwidth Shorter transmission time // quicker to send / download Can be sent where there is a file size limit	**3**	Any 3 from, max 3. Do not accept "more secure".
2(a) 0478 1.3.4	Lossy permanently removes data // the original file cannot be restored Lossless reduces the file size without permanently losing data // the original file can be restored exactly	**2**	1 mark for each type. A reversed description (e.g. "lossless removes data") scores 0 for that mark.
3(a) 0478 1.3.4	6W 3B 4W 3G	**2**	Working: runs of 6 + 3 + 4 + 3 = 16 pixels. 2 marks: all four pairs correct and in the correct order. 1 mark: two or three pairs correct and in the correct order. Accept W6 B3 W4 G3 for 1 mark max (format given in the instructions). Do not credit the runs in a different order.
3(b) 0478 1.3.4	8	**1**	4 counts + 4 colours = 8 values (instead of 16 pixels). Follow through from (a).
3(c) 0478 1.3.4	R R R R W W R W W W W W	**2**	Working: 4 + 2 + 1 + 5 = 12 pixels. 2 marks fully correct; 1 mark for three of the four runs correct and in the correct order.
3(d) 0478 1.3.4	There are no runs (of the same colour) // each run is only 1 pixel long Each pixel would need a count and a colour So 16 values would be stored instead of 8 // the data would get bigger, not smaller	**2**	Any 2 from. Working: 1B 1W 1B 1W 1B 1W 1B 1W = 8 pairs = 16 values.

QUESTION	ANSWER	MARKS	GUIDANCE
4(a) 0478 1.3.4	Reduce the resolution // reduce the number of pixels Reduce the colour depth // reduce the number of bits per pixel / number of colours	**2**	1 mark each. Do not accept RLE (lossless). Do not accept "make the image smaller" alone.
4(b) 0478 1.3.4	Reduce the sample rate // take fewer samples per second Reduce the sample resolution // use fewer bits per sample	**1**	Any 1. Do not accept "reduce the volume" or "shorten the sound".
5(a) 0478 1.3.4	Lossless: no data is permanently lost // the original file is restored exactly Lossy would permanently remove data Characters / words / numbers in the contract would be lost or changed The file could be corrupted / unreadable // the contract would no longer be accurate	**3**	Any 3 from. Do not accept "the text would be blurry".
6(a) 0478 1.3.4	Lossy (1) Much smaller file // web pages load faster // uses less bandwidth // a small loss of quality is unlikely to be noticed (1)	**2**	The reason mark depends on lossy. Lossless with "so no quality is lost" scores 1 mark only. Do not accept a file format in place of the type.

Total: 20 marks