MODULE C DATA REPRESENTATION · C8

Character sets: ASCII and Unicode: mark scheme

20 marks · OCR J277 spec 1.2.4

How to use. 1 mark per bullet up to the maximum shown. **/** separates alternative answers. Words in (brackets) are not needed for the mark. Every code and binary pattern was checked by program (Python ord() and format()). Candidates are not expected to remember codes (J277: memorisation of codes not required). A correct final answer with no working earns full marks, including Q4: OCR credits the correct characters even where "show your working" is asked. Never credit a reversed bit pattern.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 1.2.4	All the characters a computer can represent / recognise (each with its own unique binary code)	**1**	Accept "a defined list of characters and the binary codes that represent them". Do not accept "the alphabet" alone.
1(b) J277 1.2.4	8 (bits) / 1 byte / 7 (bits)	**1**	Accept 7 (standard ASCII) or 8 (as ASCII is written in the J277 exam). Do not accept 16 or 32.
2(a) J277 1.2.4	70	**1**	Working: C 67, D 68, E 69, F 70; codes are in order, so count on 1 per letter.
2(b) J277 1.2.4	01000101	**1**	Working: E = 69 = 64 + 4 + 1. Must be 8 bits in the correct order. Do not accept 01000110 (F) or 10100010 (reversed).
2(c) J277 1.2.4	Upper case and lower case are different characters, so each has its own (unique) code	**1**	Accept "f has a different code" with a reason. The value (102) is not needed.
3(a) J277 1.2.4	B: 01000010 A: 01000001 G: 01000111	**3**	Working: count on from A = 65: B = 66 (64 + 2), G = 71 (64 + 4 + 2 + 1). 1 mark per correct row, each 8 bits in the correct order.
4(a) J277 1.2.4	01100011 = 99 = c 01100001 = 97 = a 01100010 = 98 = b	**3**	Answer: **cab**. Working: 64 + 32 + 2 + 1 = 99; 64 + 32 + 1 = 97; 64 + 32 + 2 = 98; 97 is a, so 98 is b and 99 is c. 1 mark per correct character in the correct position. Upper case (CAB): max 2.

QUESTION	ANSWER	MARKS	GUIDANCE
5(a) J277 1.2.4	256	**1**	2^8 = 256. Do not accept 255.
5(b) J277 1.2.4	16	**1**	2^4 = 16. Do not accept 15 or 8.
5(c) J277 1.2.4	The more bits per character, the more characters can be represented Each extra bit doubles the number of codes / characters = $2^{\text{number of bits}}$	**2**	Max 2. Accept a correct example, e.g. 7 bits = 128, 8 bits = 256. Do not credit "more bits, fewer characters" (a common error, OCR examiners' report June 2025).
6(a) J277 1.2.4	Can represent more / a greater range of characters Can represent other languages / alphabets / emojis / symbols	**1**	Max 1. Do not accept "faster" or "smaller files".
6(b) J277 1.2.4	Uses more bits per character (So) text needs more storage / larger files / more data to send	**1**	Max 1. Do not accept "fewer characters" or "fewer bits".
6(c) J277 1.2.4	ASCII (8 bits) can only represent 256 characters … not enough for the characters of Chinese / Arabic ASCII has no codes for emojis Unicode uses more bits per character (up to 32) … so it has far more codes / covers most languages and emojis So every customer's message displays correctly	**3**	Max 3. A generic answer not linked to the languages or emojis in the scenario: max 2. Accept 16 or 32 bits for Unicode.

Total: 20 marks