MODULE M PROGRAMMING · M7

Boolean operators AND, OR and NOT in code: mark scheme

22 marks · OCR J277 spec 2.2.1 (the common Boolean operators AND, OR and NOT; comparison operators; selection)

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative answers. Words in (brackets) are not needed for the mark. Code answers are marked for logic, not exact syntax: accept OCR Exam Reference Language or any high-level language, and allow any logically valid method. Minor syntax slips are not penalised; a repeated slip (e.g. = for comparison) is penalised once then follow through. Capitals in keywords are ignored (`and` = `AND`). Do not accept maths symbols (≥, ≤, ≠) or English ("is more than") in place of operators.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.2.1	True only when both conditions are True: AND Reverses a Boolean value: NOT True when at least one condition is True: OR	**3**	1 mark per correct row. More than one tick in a row: 0 for that row.
2(a) J277 2.2.1	`temperature > 20 AND doorOpen`: False `temperature > 30 OR NOT doorOpen`: True `NOT(temperature < 25 AND alarmSet)`: False	**3**	1 mark per correct row. Accept F/T, 0/1. Working: row 1 True AND False = False; row 2 False OR (NOT False = True) = True; row 3 NOT(True AND True) = NOT True = False.
3(a) J277 2.2.1	Two correct comparisons: `windSpeed >= 0` and `windSpeed <= 250` ... ... joined with `AND` Example: `windSpeed >= 0 AND windSpeed <= 250`	**2**	Both boundaries must be included: `> 0` or `< 250` does not get the first mark. As readings are whole numbers, accept `windSpeed > -1` and `windSpeed < 251`. Each side must be a full comparison: `windSpeed >= 0 AND <= 250` = 1 (AND mark only). Accept `NOT(windSpeed < 0 OR windSpeed > 250)` for 2. Do not accept OR joining the two in-range checks (always True): max 1.

QUESTION	ANSWER	MARKS	GUIDANCE
3(b) J277 2.2.1	Two correct comparisons: `windSpeed < 0` and `windSpeed > 250` ... ... joined with `OR` Example: `windSpeed < 0 OR windSpeed > 250`	**2**	Accept `windSpeed <= -1` and `windSpeed >= 251` (whole numbers). `<= 0` or `>= 250` (rejects a valid boundary) does not get the first mark. AND in place of OR (never True): max 1 for the comparisons. The question asks for OR: a correct `NOT(... AND ...)` with no OR gets max 1.
4(a) J277 2.2.1	Line 03 (identified) ... ... `if hour >= 7 AND hour <= 19 then` (OR changed to AND) Line 04 (identified) ... ... `if keycard == "Y" OR keycard == "y" then` (second comparison given in full)	**4**	The correction mark for each error needs the right line number. Accept `hour > 6 AND hour < 20`. Line 04: accept `keycard.upper == "Y"` or equivalent; accept a line that adds the missing `keycard ==` even if `then` is omitted. Do not accept changing OR to AND on line 04 (a keycard cannot be both "Y" and "y"). Do not accept changes to lines 01, 02 or the outputs.
4(b) J277 2.2.1	`hour <= 19` is True (when hour is 2) / one of the two comparisons is True OR gives True when at least one condition is True / OR only needs one side to be True ... ... so the whole condition on line 03 is True, and the `else` on line 09 is never reached	**2**	Max 2. Accept "every hour is either 7 or more, or 19 or less, so the condition is always True" for 2. "It should be AND" alone (no reason) = 0. Do not credit answers about line 04.

QUESTION	ANSWER	MARKS	GUIDANCE

5(a)
J277 2.2.1
	1 mark each to max 6:
	Inputs the battery level and the booked answer, storing each in a variable
	Attempt at selection (`if ... then ... endif` or equivalent)
	Correct battery check: `battery >= 20` / `battery > 19`
	Correct not-booked check: `booked == "N"` / `booked != "Y"` / `NOT(booked == "Y")`
	Both checks joined with AND (or correctly nested IFs) so "Available" only when both are met
	Outputs `Available` and `Not available` in the correct cases (including every failing case)
	Example:
	`battery = int(input("Enter battery level"))`
	`booked = input("Is the bike booked? (Y/N)")`
	`if battery >= 20 AND booked == "N" then`
	`    print("Available")`
	`else`
	`    print("Not available")`
	`endif`
		6	Do not penalise a missing `int()`/casting. Accept any variable names except reserved words (e.g. `input`). `battery > 20` loses the battery-check mark. Second side with no variable (`battery >= 20 AND == "N"`) loses the not-booked mark. Using OR between the two checks loses the AND mark. Nested IFs: the output mark needs "Not available" in both the low-battery and the booked case. Accept `booked == "N"` or `booked == "n"`. Do not accept flowcharts, structured English or bullet points.

Total: 22 marks