MODULE M PROGRAMMING · M14

Nested selection and nested loops: mark scheme

22 marks · OCR J277 spec 2.1.2 (use of nesting for selection and iteration; trace tables; identify errors and suggest fixes), 2.2.1 (selection and iteration), 2.2.3 (2D arrays)

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. Do not accept maths symbols (≥, ≤, ≠) or English ("is less than") in place of operators.

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>1(a)
J277 2.1.2</td><td>Code A: Nested selection
Code B: Nested iteration
Code C: No nesting</td><td>3</td><td>1 mark per correct row. More than one tick in a row: 0 for that row. Code C has two loops one after the other (sequence), not one inside the other.</td></tr>
<tr><td>2(a)
J277 2.1.2, 2.2.1</td><td>1 mark for each correct column:
<code>row</code>: 1, 2
<code>seat</code>: 1, 2, 3, 1, 2, 3
<code>total</code>: 0, 1, 4, 6, 12
Output: 12 (once, at the end)
Example completed table:
<table>
<tr><th>ROW</th><th>SEAT</th><th>TOTAL</th><th>OUTPUT</th></tr>
<tr><td></td><td></td><td>0</td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td></td></tr>
<tr><td></td><td>2</td><td></td><td></td></tr>
<tr><td></td><td>3</td><td>4</td><td></td></tr>
<tr><td>2</td><td>1</td><td>6</td><td></td></tr>
<tr><td></td><td>2</td><td></td><td></td></tr>
<tr><td></td><td>3</td><td>12</td><td></td></tr>
<tr><td></td><td></td><td></td><td>12</td></tr>
</table></td><td>4</td><td>Values must be in the correct order. Values may be repeated on later rows (e.g. <code>row</code> 1 written on every row of the first pass). Allow the starting 0 for <code>total</code> to be missing if 1, 4, 6, 12 are correct. Do not penalise a final <code>seat</code> value of 4 or <code>row</code> value of 3 (loop variable after the loop). The output mark is lost if any other value appears in the Output column. Working: row 1 adds 1×1 and 1×3; row 2 adds 2×1 and 2×3; seat 2 is skipped each time.</td></tr>
</table>

QUESTION	ANSWER	MARKS	GUIDANCE
3(a) J277 2.1.2	12, "M": Use a standard locker 25, "L": Take to counter 20, "L": Use a large locker	**3**	1 mark per correct row. Ignore quotation marks. Accept clear shortened answers ("standard locker", "counter", "large locker"). Row 3: 20 <= 20 is True, so the inner `if` is tested.
3(b) J277 2.1.2	Max 1: The condition on line 03 is False (25 is not 20 or less), so the `else` on line 09 / line 10 runs instead Line 04 is nested inside the True branch of the `if` on line 03 / the inner `if` is only checked when the outer condition is True	**1**	"Because it is over 20" alone = 0 unless it links to the line 03 condition being False / the inner if being skipped. Credit any other valid answer that explains the nesting.
4(a) J277 2.1.2, 2.2.3	Line 03 (identified) ... ... `for shelf = 0 to 5` Line 09 (identified) ... ... `next aisle`	**4**	The correction mark for each error needs the right line number. Line 03: accept `for shelf = 0 to 6 - 1` or any expression equal to 5. Line 09: the outer loop must be closed with its own variable. Do not accept changing line 07 (it correctly closes the inner loop), line 01 or line 02. Do not accept moving line 02 outside the outer loop (the count must restart for each aisle).
4(b) J277 2.1.2	24	**1**	4 aisles × 6 shelves = 24. Accept "4 × 6".

5(a)
J277 2.1.2, 2.2.1
	1 mark each to max 6:
	Outer loop for the levels, running 1 to 3 (3 times)
	Inner loop for the spaces, running 1 to 20 (20 times), inside the outer loop and closed before it
	Outputs the level, a dash and the space number joined together, inside the inner loop
	Attempt at selection inside the inner loop that checks the space number
	Correct condition `space <= 4` / `space < 5` ... adding `(badge)` only for those spaces and the plain code for all others
	Counter set to 0 before the loops, add 1 in the badge branch, and output after both loops end
	Example:
	`count = 0`
	`for level = 1 to 3`
	`  for space = 1 to 20`
	`  code = str(level) + "-" + str(space)`
	`  if space <= 4 then`
	`  print(code + " (badge)")`
	`  count = count + 1`
	`  else`
	`  print(code)`
	`  endif`
	`  next space`
	`next level`
	`print("Badge spaces: " + str(count))`
		6	Allow 0 to 2 and 0 to 19 loops if the output adds 1 so the codes shown are 1 to 3 and 1 to 20. Do not penalise missing `str()` or a comma separator in the output. A counter that is reset inside a loop, or output inside a loop, loses the last mark. A fixed total typed in (e.g. `print(12)`) with no counting does not get the last mark. Do not accept flowcharts, structured English or bullet points.

Total: 22 marks