MODULE M PROGRAMMING · M11

Condition-controlled loops (WHILE): mark scheme

21 marks · OCR J277 spec 2.2.1 (the use of the three basic programming constructs: iteration, count- and condition-controlled loops; ERL `while ... endwhile`, condition checked before entering the loop), with 2.1.2 (trace tables; identify common errors and suggest fixes; complete and refine algorithms) and 2.3.1 (input validation) for Q4

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative words; **//** separates alternative answers for the same mark. Words in (brackets) are not needed for the mark. **...** at the start of a bullet means it depends on the bullet before it. In code answers, ignore capitals and minor syntax slips; repeated slips are penalised once, then follow through. Any high-level language is accepted where code is written. Credit any other valid answer.

QUESTION	ANSWER	MARKS	GUIDANCE
1 J277 2.2.1	Condition checked before the code inside runs: **True** Code inside always runs at least once: **False** Suitable when the number of repeats is not known in advance: **True**	**3**	1 mark per correct row. A row with two ticks scores 0. Row 2 is false: a `while` loop checks its condition first, so if the condition is False at the start the code inside runs zero times.

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr>
<td>2(a)
J277 2.1.2
J277 2.2.1</td>
<td>
<code>total</code>: 0, 30, 75, 95
<code>boxes</code>: 0, 1, 2, 3
<code>weight</code>: 30, 45, 20, 15
Output: 3 (once, at the end only)
Example completed table:
<table>
<tr><th>LINE</th><th>TOTAL</th><th>BOXES</th><th>WEIGHT</th><th>OUTPUT</th></tr>
<tr><td>01</td><td>0</td><td></td><td></td><td></td></tr>
<tr><td>02</td><td></td><td>0</td><td></td><td></td></tr>
<tr><td>03</td><td></td><td></td><td>30</td><td></td></tr>
<tr><td>05</td><td>30</td><td></td><td></td><td></td></tr>
<tr><td>06</td><td></td><td>1</td><td></td><td></td></tr>
<tr><td>07</td><td></td><td></td><td>45</td><td></td></tr>
<tr><td>05</td><td>75</td><td></td><td></td><td></td></tr>
<tr><td>06</td><td></td><td>2</td><td></td><td></td></tr>
<tr><td>07</td><td></td><td></td><td>20</td><td></td></tr>
<tr><td>05</td><td>95</td><td></td><td></td><td></td></tr>
<tr><td>06</td><td></td><td>3</td><td></td><td></td></tr>
<tr><td>07</td><td></td><td></td><td>15</td><td></td></tr>
<tr><td>09</td><td></td><td></td><td></td><td>3</td></tr>
</table>
</td>
<td>4</td>
<td>1 mark for each correct set of changes to a variable, in the correct order, and 1 mark for the output.
Line numbers are not marked. Values may be repeated on later rows. Values may be on fewer rows (e.g. one row per pass of the loop) as long as each sequence is correct.
Do not award the <code>total</code> mark if 110 appears (95 + 15 fails the condition, so line 05 does not run again).
Do not award the output mark if anything other than 3 is output, or if 3 is output more than once.</td>
</tr>
<tr>
<td>2(b)
J277 2.2.1</td>
<td>0</td>
<td>1</td>
<td>Correct answer only. (0 + 120 <= 100 is False, so the loop body never runs and <code>boxes</code> is still 0.)
Do not accept "nothing" / "no output" / 1.</td>
</tr>
<tr>
<td>2(c)
J277 2.2.1</td>
<td><code>weight</code> must have a value before the condition (on line 04) is checked // the condition uses <code>weight</code>, and the condition is checked before the loop is entered // to get the first box weight to test in the condition</td>
<td>1</td>
<td>Must link the input to the condition / the check at the start of the loop.
Do not accept "to input a weight" alone (NE), or "to start the loop" alone.
Accept "otherwise there would be an error because <code>weight</code> has no value when line 04 is reached".</td>
</tr>
</table>

QUESTION	ANSWER	MARKS	GUIDANCE
3(a) J277 2.1.2 J277 2.2.1	`saved` is never changed / increased inside the loop // `saved` stays at 0 ... so the condition `saved < TARGET` is always True / never becomes False The loop never ends / is an infinite loop (so line 07 is never reached)	**2**	Max 2. Accept "0 is always less than 500" for the second mark. "It is an infinite loop" with no reason scores 1 mark only. Do not accept "`weeks` never changes" (it does change).
3(b) J277 2.1.2	`saved = saved + 60`	**1**	Accept `saved += 60` (high-level language). Do not accept `saved = 60`, `saved =+ 60`, `saved + 60` on its own, or `TARGET = TARGET - 60` (a constant cannot be changed).
3(c) J277 2.2.1	Weeks needed: 9	**1**	Accept 9 on its own. After 8 weeks `saved` is 480 (still less than 500); after 9 weeks it is 540. Do not accept 8 or 8.33.

4
J277 2.2.1
J277 2.3.1
	Line 02: `temp < 10` // `temp <= 9`
	... `OR temp > 30` // `OR temp >= 31`
	Line 04: `temp = int(input("Enter temperature setting"))`
	Example answer:
	`02 while temp < 10 OR temp > 30`
	`04  temp = int(input("Enter temperature setting"))`
		3	Line 02: accept `NOT(temp >= 10 AND temp <= 30)` for both marks. Accept `or` in lower case.
			Do not award the second mark if AND is used (the condition could never be True), or if the variable is missing from the second half (`temp < 10 OR > 30`): max 1.
			Do not accept maths symbols such as ≤ or ≥.
			Line 04: must assign the input to `temp`; the prompt wording may differ. Casting is not required for the mark. Do not accept `input = ...` or an input that is not stored in `temp`.

QUESTION	ANSWER	MARKS	GUIDANCE

5
J277 2.2.1
	Running total set to 0 before the loop
	First distance input (and stored) before the loop
	`while` loop with a correct condition, e.g. `distance != 0` // `distance > 0`
	... distance added to the total inside the loop
	... next distance input inside the loop
	Total output with a message after / outside the loop
	Example answer (ERL):
	`total = 0`
	`distance = int(input("Enter distance in metres (0 to finish)"))`
	`while distance != 0`
	`    total = total + distance`
	`    distance = int(input("Enter distance in metres (0 to finish)"))`
	`endwhile`
	`print("Total distance rowed: " + str(total) + " metres")`
		5	1 mark per bullet, max 5.
			Allow a loop such as `while True` with the input first inside the loop and an `if distance == 0` ... `break`: award the "input before the loop" and "while condition" marks for this.
			A `do ... until` loop: credit the other features shown, but not the `while` loop mark.
			Uninitialised total: do not award mark 1. Do not award the total mark for `total = distance` or `total =+ distance`.
			Output mark: not awarded if the output is inside the loop. A comma between message and total is allowed (e.g. Python `print("Total:", total)`). Casting is not required for any mark.

Total: 21 marks