MODULE M PROGRAMMING · M12

Post-condition loops, and choosing the right loop: mark scheme

22 marks · Cambridge 0478 syllabus 8.1.4(c) (iteration: count-controlled, pre-condition and post-condition loops), 8.1.4(d) (totalling and counting), 7.5(a) (validation: range check), 7.7 (trace tables), 7.9 (write and amend algorithms)

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. Algorithm answers must be in pseudocode; marks are for the features listed, not for one exact algorithm. Ignore minor slips in case and spacing. Accept `<-` or `=` for assignment. Do not accept English ("is more than") or maths symbols (≥, ≤, ≠) in place of operators. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 8.1.4(c)	One mark for each correct row: • always executed at least once, condition tested after: post-condition • executed a set number of times: count-controlled • condition tested before, may not be executed at all: pre-condition	**3**	Accept the keywords: `REPEAT ... UNTIL` // `FOR ... NEXT` // `WHILE ... ENDWHILE` for the matching row. "Condition-controlled" alone for row 1 or row 3 = 0 (does not say which).
2(a) 0478 7.7, 8.1.4(c), 8.1.4(d)	One mark for each correct column: • `Weight`: 12, 18, 9, 15 • `Total`: 0, 12, 30, 39, 54 • `Count`: 0, 1, 2, 3, 4 • OUTPUT: `Items in crate: 4` (once, at the end)	**4**	The value 20 is never input: a `Weight` column that includes 20 does not get the `Weight` mark, and any values after 54 / 4 lose the mark for that column. The output may be on the same row as the last values or on the next row. Ignore quotation marks round the output. Accept `Items in crate:4` (spacing).
2(b)(i) 0478 7.7, 8.1.4(c)	`Items in crate: 1`	**1**	Ignore quotation marks. Do not accept `Items in crate: 2` (5 is not input).
2(b)(ii) 0478 8.1.4(c)	Any one from: • after the first value, `Total` is 60, which is more than 50, so the condition (`Total > 50`) is TRUE when it is tested (at the end of the first pass) and the loop stops • the condition is tested after the statements, and the first value alone makes it TRUE, so the statements are not executed again	**1**	Max 1. The answer must link the first value / a total of 60 to the condition being TRUE (or met). "Because it is a REPEAT loop" alone = 0. Do not accept "the condition is FALSE so the loop stops" (reversed).

QUESTION	ANSWER	MARKS	GUIDANCE
3(a) 0478 8.1.4(c)	One mark for each correct row: • traffic lights: count-controlled // FOR • staff entrance: post-condition // REPEAT ... UNTIL • car park barrier: pre-condition // WHILE	**3**	"Condition-controlled" alone for row 2 or row 3 = 0 (does not say which). Row 2: do not accept pre-condition (it must ask at least once). Row 3: do not accept post-condition.
3(b) 0478 8.1.4(c)	Any two from: • the code must be input / asked for at least once // the statements in a post-condition loop always execute at least once • the code cannot be checked until it has been input // the condition is tested after the input, so there is a value to test • the loop repeats (asks again) until the condition is TRUE / until a valid code is entered • the number of attempts needed is not known in advance (so a count-controlled loop is not suitable)	**2**	Max 2. "It checks the code" alone = 0. Credit any other valid answer.

4(a)
0478 7.9, 8.1.4(c)
	One mark per bullet point:
	• `Floor ← 10` before the loop and `REPEAT ... UNTIL` structure
	• `OUTPUT "Floor ", Floor` then `Floor ← Floor - 3`, both inside the loop (in that order)
	• correct condition after UNTIL: `Floor <= 0` // `Floor < 1` // `NOT (Floor > 0)` // `Floor < 0`
	Example:
	`Floor ← 10`
	`REPEAT`
	`  OUTPUT "Floor ", Floor`
	`  Floor ← Floor - 3`
	`UNTIL Floor <= 0`
		3	The output must be Floor 10, 7, 4, 1 only. `UNTIL Floor > 0` (condition not reversed: stops after one pass) and `UNTIL Floor = 0` (never TRUE: 10, 7, 4, 1, −2 ...) do not get the third mark. `UNTIL Floor < 0` is accepted: with these values `Floor` goes 10, 7, 4, 1, −2, so it gives the same output. Do not award the first mark for a WHILE or FOR loop.

QUESTION	ANSWER	MARKS	GUIDANCE

5(a)
0478 7.9, 7.5(a), 8.1.4(c)
	One mark per bullet point:
	• `REPEAT` ... `UNTIL` loop (with a condition after UNTIL)
	• prompt (OUTPUT) and `INPUT` of the number of pots, both inside the loop
	• error message output only when the number is less than 1 or more than 40 (e.g. IF ... THEN ... ENDIF inside the loop)
	• correct UNTIL condition so the loop ends only for 1 to 40 inclusive: `(Pots >= 1) AND (Pots <= 40)`
	• message that includes the number of pots output once, after the loop
	Example:
	`REPEAT`
	`  OUTPUT "Enter the number of pots (1 to 40)"`
	`  INPUT Pots`
	`  IF (Pots < 1) OR (Pots > 40)`
	`  THEN`
	`  OUTPUT "Error: the number must be from 1 to 40"`
	`  ENDIF`
	`UNTIL (Pots >= 1) AND (Pots <= 40)`
	`OUTPUT "Order accepted for ", Pots, " pots"`
		5	Accept `Pots > 0` and `Pots < 41`. Each side of a compound condition must be a full comparison (`Pots >= 1 AND <= 40` loses the condition mark). OR joining the two valid checks loses the condition mark. A WHILE loop (pre-condition) does not get the first mark, but the other marks can be given. A single IF with no loop (only one re-entry): max 2. Accept `NOT` forms that are logically equivalent. Program code instead of pseudocode: 0.

Total: 22 marks