MODULE K ALGORITHM DESIGN · K5

Reading and writing pseudocode: mark scheme

20 marks · Cambridge 0478 syllabus 7.2(c) (design methods: pseudocode), 7.3 (purpose of a given algorithm), 7.8 (identify and correct errors), 7.9 (write algorithms in pseudocode; precision), 8.1.1 (declare variables and constants), 8.1.4 (selection, iteration, counting, totalling, operators) and section 4 (pseudocode)

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. **...** at the start of a point means it builds on the point before. Code answers must be in pseudocode; ignore minor differences in capitals, spacing and indentation. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 7.2(c)	Row 2 ticked only (keywords and structured, code-like statements without the rules of one programming language)	**1**	More than one tick = 0. Row 1 is a structure diagram, row 3 a flowchart, row 4 a trace table.

2(a) 0478 7.9, 8.1.4(f)	One mark for each correct condition:
	`Distance > 100`
	`Answer <> 'N'`
	`Hours >= 1 AND Hours <= 12` // `(Hours >= 1) AND (Hours <= 12)` // `Hours > 0 AND Hours < 13`
		3	Allow `IF` in front of the condition. Row 2: allow `"N"`; allow `NOT(Answer = 'N')`; N must be in quotes; do not accept `!=` (not a pseudocode operator). Row 3: do not accept `Hours >= 1 AND <= 12` (second comparison has no identifier), `OR` instead of AND, `1 <= Hours <= 12`, or => / ≥ / ≤.

3(a) 0478 8.1.1, section 4	One mark for each correct statement:
	`CONSTANT MaxSeats ← 52`
	`DECLARE SeatsLeft : INTEGER`
		2	Allow = or <– for ← in the constant. Do not accept `REAL` or `STRING` for the variable. Do not accept a variable declaration for MaxSeats.

4(a) 0478 7.3	Any two from: To input the weights of 5 loaves To total / add up the weights (of the loaves) To count how many loaves weigh 400 (grams) or more // are accepted To output the total weight and the number of loaves accepted	**2**	max 2. Do not accept "more than 400" / "over 400" for the counting point (400 is accepted). Do not accept "checks the loaves" alone.
4(b)(i) 0478 8.1.4(c)	07	**1**	Allow 7.

QUESTION	ANSWER	MARKS	GUIDANCE
4(b)(ii) 0478 8.1.4(d)	13	**1**	Correct answer only. Line 10 is totalling, not counting.

4(c)
0478 7.3
	One mark for each correct output:
	`Total weight: 2013`
	`Loaves accepted: 3`
		2	Ignore quotation marks. Allow 2013 and 3 without the message text if each is clearly the correct output.

5(a) 0478 7.8	One mark for each error identified and correctly corrected: Line 05: `FOR Day ← 1 TO 7` Line 08: `IF Visitors > Highest` Line 10: `Highest ← Visitors` Line 13: `OUTPUT "Highest number of visitors: ", Highest`	**4**	A correction scores only with its correct line number. Line 05: allow `FOR Day ← 0 TO 6` (also 7 days). Line 08: allow >=. Allow a clear description, e.g. "change < to >", "store Visitors not Day in Highest", "output Highest not Visitors".

6(a)
0478 7.9, 8.1.4(c)(d)
	Any four from:
	Count initialised to 0 before the loop
	Condition-controlled loop that stops when −1 is input // `REPEAT ... UNTIL Time = -1` // `WHILE Time <> -1 DO ... ENDWHILE`
	Input of a journey time inside the loop
	`IF Time > 30 … THEN Count ← Count + 1 … ENDIF`
	Output of the count with a message, after the loop
	`DECLARE Time : INTEGER`
	`DECLARE Count : INTEGER`
	`Count ← 0`
	`OUTPUT "Enter journey time in minutes, -1 to stop"`
	`INPUT Time`
	`WHILE Time <> -1 DO`
	`  IF Time > 30`
	`  THEN`
	`  Count ← Count + 1`
	`  ENDIF`
	`  OUTPUT "Enter journey time in minutes, -1 to stop"`
	`  INPUT Time`
	`ENDWHILE`
	`OUTPUT "Journeys over 30 minutes: ", Count`
		4	max 4. A WHILE loop gets the loop mark if Time has a value before the condition is first tested: either an input before the loop and again at the end of the loop body, or Time set to a value other than −1 before the loop and the input at the start of the loop body. A REPEAT ... UNTIL loop is also fine (−1 is not more than 30, so it is never counted). Do not award the selection mark for `>= 30`. Do not award the output mark if the count is output inside the loop. A FOR loop does not get the loop mark. Program code (e.g. Python) scores 0.

Total: 20 marks