MODULE L STANDARD ALGORITHMS · L7

Totalling, counting, maximum, minimum and average: mark scheme

21 marks · Cambridge 0478 syllabus 7.4 (standard methods: totalling, counting, maximum, minimum, average), 7.7 (trace tables), 7.8 (errors), 7.9 (write algorithms), 8.1.4(c)/(d) (iteration; totalling and counting), 8.1.7 (ROUND), 8.2 (arrays)

How to use. One mark per bullet point up to the max shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Algorithms must be in pseudocode: program code is not credited. Ignore minor syntax slips (e.g. <- or = for ←, a missing THEN) if the meaning is clear. Credit any other valid method that meets the requirement.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 7.4, 8.1.4(d)	One mark for each correct line: `TotalWeight ← TotalWeight + Weight` → totalling `IF Weight > Heaviest ...` → finding the maximum `Parcels ← Parcels + 1` → counting `MeanWeight ← TotalWeight / Parcels` → finding the average	**4**	More than one line from a statement: 0 for that statement.
2(a) 0478 7.7, 7.4	One mark for each of: `Day` 1, 2, 3, 4, 5 and `Reading` 24, 31, 18, 30, 27 `Total` 0, 24, 55, 73, 103, 130 `Lowest` 100, 24, 18 `Count` 0, 1, 2 `Average` 26 and OUTPUT: Lowest: 18 // Hot days: 2 // Average: 26	**5**	Mark by column (or pair of columns); ignore the row a value is on if the order is correct. Initial values 0, 100, 0 must be shown for the Total, Lowest and Count marks. Count must reach 2 (30 is counted because of >=). Accept 26.0. OUTPUT: accept the values 18, 2, 26 without the message text; do not accept quotation marks around outputs.

QUESTION	ANSWER	MARKS	GUIDANCE

2(a)
completed table

DAY	READING	TOTAL	LOWEST	COUNT	AVERAGE	OUTPUT
		0	100	0		
1	24	24	24			
2	31	55		1		
3	18	73	18			
4	30	103		2		
5	27	130				
					26	Lowest: 18
						Hot days: 2
						Average: 26

2(b)
0478 7.4, 7.3

Any two from:
100 is higher than any (likely) temperature reading // every reading will be lower than 100
... so the first reading (24) replaces it // so `Lowest` is updated to a real reading
If `Lowest` started at 0, no reading would be lower // `Lowest` would never change
... so the algorithm would output 0, which is not the lowest reading // the wrong answer

2

max 2. Credit "Lowest could instead start as the first reading input" as a valid point. Do not accept "100 is a big number" alone.

3(a)
0478 7.8, 7.4, 8.2

One mark for each error identified **and** corrected:
Line 02: `Count ← 0`
Line 03: `FOR Trip ← 1 TO 20`
Line 04: `IF Distance[Trip] > Longest`
Line 10: `Count ← Count + 1`

4

Line number and correction both needed for each mark. Line 04: accept >=. Line 04 also accept `IF Longest < Distance[Trip]`. Line 01 is not an error (all distances are greater than 0).

QUESTION	ANSWER	MARKS	GUIDANCE

4(a)
0478 7.9, 7.4, 8.1.4(c), 8.1.4(d), 8.1.7
	One mark per bullet point, max 6:
	Initialise `Total` and `Count` to 0 (before the loop)
	Condition-controlled loop (REPEAT ... UNTIL // WHILE ... DO ... ENDWHILE) ...
	... that stops when 0 is input
	Input of the price inside the loop
	Price added to `Total` inside the loop
	`Count` increased by 1 for each price, not counting the 0
	Average calculated as `Total / Count` after the loop
	Output of the total and of the average using `ROUND(..., 2)`, with messages
	`Total ← 0`
	`Count ← 0`
	`REPEAT`
	`  OUTPUT "Enter the price, or 0 to finish"`
	`  INPUT Price`
	`  IF Price <> 0`
	`  THEN`
	`  Total ← Total + Price`
	`  Count ← Count + 1`
	`  ENDIF`
	`UNTIL Price = 0`
	`Average ← Total / Count`
	`OUTPUT "Total of prices: ", Total`
	`OUTPUT "Average price: ", ROUND(Average, 2)`
		6	Accept a WHILE loop with an input before the loop and again at the end of the loop. Accept counting the 0 and then subtracting 1 after the loop. Adding the 0 to Total is not an error.
			Average worked out inside the loop: do not award that bullet unless it is also correct after the last price.
			Not required: checking that Count is not 0. A FOR loop does not gain the loop bullets.

Total: 21 marks