MODULE O ROBUST PROGRAMS · O6

Finding and correcting errors in programs: mark scheme

21 marks · Cambridge 0478 syllabus 7.8 (identify errors in given algorithms and suggest ways of correcting these errors), 7.7 (complete a trace table to document a dry-run of an algorithm), 7.6 (suggest and apply suitable test data) and 4.2.3 (how errors are reported by a compiler and an interpreter)

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. Corrections should be in pseudocode; ignore minor slips (e.g. <− or = for ←, missing `THEN`) as long as the logic is clear, unless the slip is the error being corrected. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 4.2.3	One mark per correct row: Lines 1 to 7 run before any error is reported: Interpreter One error report lists both errors: Compiler Line 20 error only reported after line 8 is corrected: Interpreter None of the program can be run until both errors are corrected: Compiler	**4**	A row with both boxes ticked scores 0 for that row. An interpreter translates and executes line by line and stops when it reaches an error; a compiler translates the whole code first and gives an error report for the whole code, and no executable file is produced until the errors are corrected.

2(a)
0478 7.8
	One mark for each correct statement:
	`Count ← Count + 1`
	`FOR Hour ← 1 TO 24` // `FOR Hour ← 0 TO 23`
	`IF Length >= 2 AND Length <= 5`
	`Prefix ← SUBSTRING(Code, 1, 3)`
		4	Row 3: accept brackets round each comparison // `2 <= Length AND Length <= 5`. Do not accept `OR`, ≥/≤, or a second comparison with no variable.
			Row 4: the start position must be 1 and the length must be 3. Do not accept `LEFT` or program code.

QUESTION	ANSWER	MARKS	GUIDANCE
3(a) 0478 7.8	One mark for each error identified **and** corrected: Line 09 // `UNTIL Weight >= 500 AND Weight <= 4000` Line 13 // `ENDWHILE` should be `ENDIF` Line 14 // `NEXT Weight` should be `NEXT Vehicle` Line 15 // `OUTPUT "Vehicles over 3000 kg: ", Heavy` // output Heavy instead of Weight	**4**	max 4. The line number alone, or a correction with no way of telling which line it replaces, scores 0. Line 09: accept `UNTIL NOT (Weight < 500 OR Weight > 4000)` // `Weight > 499 AND Weight < 4001` (weights are whole numbers). Do not accept "change OR to AND" if the comparisons are also changed wrongly. Do not credit changes to lines 04, 10 or 12 (they are correct).
3(b) 0478 7.6, 7.8	One mark for each: **Weight:** any whole number less than 500 or greater than 4000, e.g. 100 // 499 // 4001 // 6000 **Type:** abnormal (for any value outside the range) // boundary (for 499 or 4001 only) **Expected result:** the weight is rejected // an error message is shown and the weight is asked for again	**3**	The type mark must match the weight given: 100 or 6000 is abnormal, not boundary. Do not accept "extreme" (extreme data is the largest/smallest *acceptable* value) or "normal". A weight from 500 to 4000 does not show the error: it scores 0 for all three marks. Do not accept "the weight is accepted" as the expected result (that is what the faulty algorithm does).

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>4(a)
0478 7.7</td><td>One mark per correct column:
Index: 1, 2, 3, 4, 5
Count: 0, 3, 7, 9
OUTPUT: Low rainfall days: 9
<table><tr><th>INDEX</th><th>COUNT</th><th>OUTPUT</th></tr><tr><td></td><td>0</td><td></td></tr><tr><td>1</td><td>3</td><td></td></tr><tr><td>2</td><td></td><td></td></tr><tr><td>3</td><td>7</td><td></td></tr><tr><td>4</td><td></td><td></td></tr><tr><td>5</td><td>9</td><td></td></tr><tr><td></td><td></td><td>Low rainfall days: 9</td></tr></table></td><td>3</td><td>Allow Count values repeated on rows where they do not change.
Allow the starting 0 to be missing if 3, 7, 9 are correct.
OUTPUT: accept 9 alone; ignore quotation marks. Do not accept 3 (the trace must follow the algorithm as written).
Do not penalise a final 6 in the Index column.</td></tr>
<tr><td>4(b)
0478 7.7</td><td>Low rainfall days: 3</td><td>1</td><td>Accept 3 alone.</td></tr>
<tr><td>4(c)
0478 7.8</td><td>Line 05 (identified)
… <code>Count ← Count + 1</code></td><td>2</td><td>The correction mark may be given if the line number is missing but the statement makes clear which line is meant.
Do not credit changes to line 03 (it is correct).</td></tr>
</table>

Total: 21 marks