MODULE L STANDARD ALGORITHMS · L3

Bubble sort: mark scheme

20 marks · Cambridge 0478 syllabus 7.4 (standard methods: bubble sort), 7.7 (trace tables), 7.8 (errors in algorithms), 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. Credit any other valid answer that shows the same understanding. Code answers must be in pseudocode; ignore minor syntax slips (e.g. <− or = for ←).

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 7.4	Any four from: Compares each pair of adjacent values // compares the first value with the second (then the second with the third, ...) If the first value is larger than the second // they are in the wrong order, they are swapped (A temporary variable is used to) swap the values Moves along the array until the end is reached // this is one pass After each pass the largest (remaining) value is at the end // in its correct position Passes are repeated ... ... until a pass makes no swaps // until the array is sorted // for (number of values − 1) passes	**4**	max 4. Credit a correct pseudocode answer that is clearly explained. Do not accept "it sorts the values" or "it puts them in order" (outcome only). Do not accept "swaps the values if they are in the right order".
2(a) 0478 7.4	After pass 1: 21 34 15 28 19 40 After pass 2: 21 15 28 19 34 40 After pass 3: 15 21 19 28 34 40	**3**	One mark per correct row. Follow through for pass 2 and pass 3 from the candidate's own previous row if it is one correct bubble sort pass of that row. The array is not sorted after pass 3 (19 and 21 are still out of order).
2(b) 0478 7.4	4	**1**	Swaps in pass 1: 34/21, 40/15, 40/28, 40/19.

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr>
<td>3(a)
0478 7.7</td>
<td>One mark for each correct column: Swapped, Index, Temp
One mark for columns Stock[1] and Stock[2] both correct
One mark for columns Stock[3] and Stock[4] both correct
<table>
<tr><th>Swapped</th><th>Index</th><th>Temp</th><th>Stock[1]</th><th>Stock[2]</th><th>Stock[3]</th><th>Stock[4]</th></tr>
<tr><td></td><td></td><td></td><td>4</td><td>9</td><td>2</td><td>7</td></tr>
<tr><td>FALSE</td><td>1</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>TRUE</td><td>2</td><td>9</td><td></td><td>2</td><td>9</td><td></td></tr>
<tr><td></td><td>3</td><td>9</td><td></td><td></td><td>7</td><td>9</td></tr>
<tr><td>FALSE</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>TRUE</td><td>1</td><td>4</td><td>2</td><td>4</td><td></td><td></td></tr>
<tr><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>3</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>FALSE</td><td>1</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>3</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
Column sequences: Swapped FALSE, TRUE, FALSE, TRUE, FALSE // Index 1, 2, 3, 1, 2, 3, 1, 2, 3 // Temp 9, 9, 4 // Stock[1] 4, 2 // Stock[2] 9, 2, 4 // Stock[3] 2, 9, 7 // Stock[4] 7, 9</td>
<td>5</td>
<td>Mark each column as a sequence of values; rows may be laid out differently. Accept TRUE written again when the second swap of pass 1 is made. Accept Temp 9, 4 (second 9 not rewritten). Ignore a 4 in the Index column at the end of a pass. No further rows after the third pass: the loop stops because no swap was made. Do not credit Temp values 2 or 7.</td>
</tr>
<tr>
<td>3(b)
0478 7.9</td>
<td><code>IF Stock[Index] < Stock[Index + 1]</code></td>
<td>1</td>
<td>Accept <code>IF Stock[Index + 1] > Stock[Index]</code>. Do not accept <= (equal values would be swapped on every pass, so the loop would not stop if two trays held the same number). Do not accept "change > to <" with no line written.</td>
</tr>
<tr>
<td>4(a)
0478 7.8</td>
<td>One mark for each error identified and corrected:
Line 05: <code>FOR Position ← 1 TO 4</code> // <code>FOR Position ← 1 TO 5 - Pass</code>
Line 06: <code>IF Lengths[Position] > Lengths[Position + 1]</code>
Line 09: <code>Lengths[Position] ← Lengths[Position + 1]</code></td>
<td>3</td>
<td>Line number and correction are both needed for each mark. Line 05: <code>TO 5</code> reads <code>Lengths[6]</code>, outside the array. Line 06: < gives descending order. Line 09: stores <code>Hold</code> back in the same place so the value in <code>Lengths[Position + 1]</code> is lost. Accept <code>IF Lengths[Position + 1] < Lengths[Position]</code> for line 06. Do not credit changes to lines 04, 08 or 10 (these are correct).</td>
</tr>
</table>

QUESTION	ANSWER	MARKS	GUIDANCE

5(a)
0478 7.9
	One mark per bullet point:
	A new temporary variable stores the name: `TempName ← Plant[Index]`
	`Plant[Index] ← Plant[Index + 1]` **and** `Plant[Index + 1] ← TempName` (in this order, after the name is stored)
	Placed inside the IF ... THEN // between lines 08 and 13 // next to the swap of `Stock` (so names are swapped only when the numbers are swapped)
	`TempName ← Plant[Index]`
	`Plant[Index] ← Plant[Index + 1]`
	`Plant[Index + 1] ← TempName`
		3	Accept any identifier for the new variable. Reusing `Temp`: no first mark (Temp is declared INTEGER at line 02 and a plant name is a STRING), but marks 2 and 3 can still be given.
			Statements placed outside the IF (e.g. after line 13 or before line 07): no third mark. Ignore any DECLARE statement.

Total: 20 marks