MODULE K ALGORITHM DESIGN · K8

Trace tables and dry-runs: mark scheme

21 marks · Cambridge 0478 syllabus 7.7: complete a trace table to document a dry-run of an algorithm (variables, outputs and user prompts at each step); 7.3: state the purpose of an algorithm; 7.8: identify errors and suggest corrections

How to use. One mark per bullet point / per correct column, up to the max shown. **//** separates alternative answers. Words in (brackets) are not needed for the mark. **Trace tables:** a column is correct when its values appear in the right order; repeated values on later rows are not penalised, and rows may be laid out differently from the example. Outputs are copied exactly but without quotation marks; ignore small spacing slips and a missing colon. A column that carries on after the stopping condition has been met is not correct. Credit any other valid answer that shows the same understanding.

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>1(a)
0478 7.7</td><td>Working through the algorithm by hand, one step at a time, recording what happens (second row)</td><td>1</td><td>Two or more ticks = 0.</td></tr>
<tr><td>1(b)
0478 7.7</td><td>Any one from:
• outputs // the values output
• user prompts // messages asking the user for input</td><td>1</td><td>Do not accept “inputs” alone, “line numbers”, or “errors”.</td></tr>
<tr><td>2(a)
0478 7.7</td><td>One mark for each correct column:
Total: 0, 18, 33, 53
Count: 0, 1, 2, 3
Seats: 18, 25, 15, 20 (12 is not used)
OUTPUT: Enter seats, Enter seats, Too many, Enter seats, Enter seats, Bookings accepted: 3
<table>
<tr><th>Total</th><th>Count</th><th>Seats</th><th>OUTPUT</th></tr>
<tr><td>0</td><td>0</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>Enter seats</td></tr>
<tr><td>18</td><td>1</td><td>18</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Enter seats</td></tr>
<tr><td></td><td></td><td>25</td><td>Too many</td></tr>
<tr><td></td><td></td><td></td><td>Enter seats</td></tr>
<tr><td>33</td><td>2</td><td>15</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Enter seats</td></tr>
<tr><td>53</td><td>3</td><td>20</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Bookings accepted: 3</td></tr>
</table></td><td>4</td><td>The prompt and the input may share a row. 20 is not greater than 20, so it is accepted: Total 53 and Count 3. 25 must not be added to Total. Seats column containing 12 = 0 for Seats. OUTPUT must include all four prompts, Too many once, and the final message; Bookings accepted: 3 alone = 0 for OUTPUT.</td></tr>
</table>

<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:
Index: 1, 2, 3, 4, 5, 6
Low: 38, 31, 29 (only)
Above: 0, 1, 2 (only)
OUTPUT: Lowest: 29 then Above limit: 2
<table><tr><th>Index</th><th>Low</th><th>Above</th><th>OUTPUT</th></tr><tr><td></td><td>38</td><td>0</td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td>1</td><td></td></tr><tr><td>3</td><td>31</td><td></td><td></td></tr><tr><td>4</td><td></td><td>2</td><td></td></tr><tr><td>5</td><td>29</td><td></td><td></td></tr><tr><td>6</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td>Lowest: 29</td></tr><tr><td></td><td></td><td></td><td>Above limit: 2</td></tr></table></td><td>4</td><td>40 is not greater than 40, so Above stays at 2 when Index is 6: Above reaching 3 = 0 for Above. Low must not take 45, 52 or 40. Accept both outputs on one row in the right order. Accept repeated values of Low / Above on rows where they do not change.</td></tr>
<tr><td>3(b)
0478 7.3</td><td>Any two from:
• finds // outputs the lowest // minimum (river-level) reading
• counts how many readings are above 40 // greater than 40 (cm)
• outputs the number of readings above 40</td><td>2</td><td>Max 1 if only one of the two results (lowest reading, count above 40) is given. Do not accept “40 or more” / “at least 40” for the count. Do not accept “sorts the readings” or a line-by-line description with no result.</td></tr>
<tr><td>4(a)
0478 7.7</td><td>One mark for each correct column:
Number: 50, 16, 5, 1
Count: 0, 1, 2, 3
OUTPUT: Enter a number (first) ... 3 (last)
<table><tr><th>Number</th><th>Count</th><th>OUTPUT</th></tr><tr><td></td><td></td><td>Enter a number</td></tr><tr><td>50</td><td>0</td><td></td></tr><tr><td>16</td><td>1</td><td></td></tr><tr><td>5</td><td>2</td><td></td></tr><tr><td>1</td><td>3</td><td></td></tr><tr><td></td><td></td><td>3</td></tr></table></td><td>3</td><td>DIV gives the whole-number part: 16.67 / 16.6 / 5.33 = 0 for Number. OUTPUT without the prompt = 0 for OUTPUT. Count must stop at 3.</td></tr>
<tr><td>4(b)
0478 7.7</td><td>0</td><td>1</td><td>The condition 1 > 1 is false at the first check, so the loop body never runs. Accept “Enter a number, 0”. Do not accept 1.</td></tr>
</table>

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr>
<td>5(a)
0478 7.7</td>
<td>Count: 1, 2, 3, 4 and Score: 12, 9, 15, 6
Total: 0, 12, 0, 9, 0, 15, 0, 6
Average: 1.5 and OUTPUT: Enter score (four times) ...
Average is 1.5
<table>
<tr><th>Count</th><th>Total</th><th>Score</th><th>Average</th><th>OUTPUT</th></tr>
<tr><td>1</td><td>0</td><td></td><td></td><td>Enter score</td></tr>
<tr><td></td><td>12</td><td>12</td><td></td><td></td></tr>
<tr><td>2</td><td>0</td><td></td><td></td><td>Enter score</td></tr>
<tr><td></td><td>9</td><td>9</td><td></td><td></td></tr>
<tr><td>3</td><td>0</td><td></td><td></td><td>Enter score</td></tr>
<tr><td></td><td>15</td><td>15</td><td></td><td></td></tr>
<tr><td>4</td><td>0</td><td></td><td></td><td>Enter score</td></tr>
<tr><td></td><td>6</td><td>6</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>1.5</td><td>Average is 1.5</td></tr>
</table>
</td>
<td>3</td>
<td>One mark per bullet. The trace must follow the algorithm as written: a Total column of 0, 12, 21, 36, 42 (the corrected algorithm) = 0 for Total, and Average 10.5 = 0 for the third mark. FT: Average and final OUTPUT correct for the candidate's final Total ÷ 4.</td>
</tr>
<tr>
<td>5(b)
0478 7.8</td>
<td>Line 02
Move <code>Total ← 0</code> to before the FOR loop // before line 01 // outside the loop</td>
<td>2</td>
<td>One mark for the line, one for the correction. Accept "Total is reset to 0 every time round the loop; it should be set to 0 once, before the loop starts". Do not accept "delete line 02" alone (Total would not be initialised). Do not accept changes to line 05 or line 07.</td>
</tr>
</table>

Total: 21 marks