MODULE M PROGRAMMING · M10

# Count-controlled loops (FOR): mark scheme

22 marks · Cambridge 0478 syllabus 8.1.4(c) (iteration: count-controlled loops), with 8.1.4(d) (totalling and counting) for Q2 and Q5, 8.2.1 (arrays) for Q4, 7.7 (trace tables) for Q2 and 7.8 (identifying errors and suggesting corrections) for Q4

**How to use.** One mark per bullet point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Ignore the case of keywords and minor spacing. Accept <- or = for ← in handwritten answers. Pseudocode answers are expected; credit any other valid answer that shows the same logic.

| QUESTION | ANSWER | MARKS | GUIDANCE |
|---|---|---|---|
| **1**<br>0478 8.1.4(c) | One mark for each correct row:<br>`FOR Count ← 1 TO 8`: 8<br>`FOR Count ← 0 TO 15 STEP 5`: 4<br>`FOR Count ← 20 TO 2 STEP -3`: 7<br>`FOR Count ← 6 TO 6`: 1<br>`FOR Count ← 7 TO 3`: 0 // none | **5** | Values taken by `Count`: row 2: 0, 5, 10, 15; row 3: 20, 17, 14, 11, 8, 5, 2.<br>Row 4: the start value equals the end value, so the statements run once.<br>Row 5: the start value is greater than the end value and there is no STEP, so the statements are not executed.<br>Do not accept a list of values instead of the number of times. |

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr>
<td>2<br>0478 8.1.4(c)<br>0478 8.1.4(d)<br>0478 7.7</td>
<td>One mark for each correct column:<br>Day: 1, 2, 3, 4<br>Sold: 35, 52, 41, 60<br>Total: 0, 35, 87, 128, 188<br>Best: 0, 35, 52, 60<br>OUTPUT: Total 188 Best 60<br>Example completed table:
<table>
<tr><th>DAY</th><th>SOLD</th><th>TOTAL</th><th>BEST</th><th>OUTPUT</th></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td></td></tr>
<tr><td>1</td><td>35</td><td>35</td><td>35</td><td></td></tr>
<tr><td>2</td><td>52</td><td>87</td><td>52</td><td></td></tr>
<tr><td>3</td><td>41</td><td>128</td><td></td><td></td></tr>
<tr><td>4</td><td>60</td><td>188</td><td>60</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>Total<br>188 Best<br>60</td></tr>
</table>
</td>
<td>5</td>
<td>Values must be in the correct order. Repeated values on later rows (e.g. 52 written again for Best on the Day 3 row) are not penalised.<br>Best must not change on the Day 3 row: 41 is not greater than 52.<br>Ignore a final value of 5 for Day.<br>OUTPUT: no quotation marks needed; the output must appear once only. Do not accept 188 and 60 without the words.</td>
</tr>
<tr>
<td>3<br>0478 8.1.4(c)</td>
<td>FOR loop from 100 to 0 with a matching NEXT (using the same identifier)<br>STEP -25<br>OUTPUT of the loop variable inside the loop<br>Example answer:
<pre>FOR Level ← 100 TO 0 STEP -25
    OUTPUT Level
NEXT Level</pre>
</td>
<td>3</td>
<td>One mark per bullet point. Max 3.<br>Credit any count-controlled loop that outputs the same five values in order, e.g. FOR Count ← 4 TO 0 STEP -1 with OUTPUT Count * 25, or FOR Count ← 0 TO 4 with OUTPUT 100 - Count * 25 (award bullets 1 and 2 for a correct loop of this kind).<br>Do not award bullet 3 for outputting fixed values (e.g. five separate OUTPUT statements).<br>A REPEAT or WHILE loop scores 0: a count-controlled loop was asked for.</td>
</tr>
</table>

| QUESTION | ANSWER | MARKS | GUIDANCE |
|---|---|---|---|
| **4**<br>0478 8.1.4(c)<br>0478 8.2.1<br>0478 7.8 | One mark for each line identified **and** corrected:<br>Line 01: `DECLARE Post : INTEGER`<br>Line 03: `FOR Post ← 1 TO 12`<br>Line 04: `Marker[Post] ← Post * 5`<br>Line 05: `NEXT Post` | **4** | Line 01: the loop identifier must be an INTEGER. Accept "change REAL to INTEGER".<br>Line 03: the loop must reach 12 so that `Marker[12]` is given a value. Do not accept `0 TO 12` (index 0 is outside the array bounds).<br>Line 04: accept `5 * Post`.<br>Line 05: NEXT must use the loop identifier.<br>Line 02 and line 06 are correct; no mark for these. |
| **5**<br>0478 8.1.4(c)<br>0478 8.1.4(d) | FOR loop from 1 TO 12 with matching NEXT<br>INPUT of the number of passengers inside the loop<br>Total set to 0 before the loop and passengers added to it inside the loop<br>Counter set to 0 before the loop and increased by 1 inside an IF that checks for more than 10 (`> 10`)<br>OUTPUT of the total and the count with messages, after the loop<br>**Example answer:**<br>`Total ← 0`<br>`BusyJourneys ← 0`<br>`FOR Journey ← 1 TO 12`<br>`  INPUT Passengers`<br>`  Total ← Total + Passengers`<br>`  IF Passengers > 10`<br>`  THEN`<br>`  BusyJourneys ← BusyJourneys + 1`<br>`  ENDIF`<br>`NEXT Journey`<br>`OUTPUT "Total passengers: ", Total`<br>`OUTPUT "Journeys with more than 10 passengers: ", BusyJourneys` | **5** | One mark per bullet point. Max 5.<br>Accept `FOR Journey ← 0 TO 11`.<br>Do not award bullet 3 if the total is reset to 0 inside the loop. Do not award bullet 4 for `>= 10` (more than 10 means 11 or more); accept `>= 11`.<br>Do not award bullet 5 if the outputs are inside the loop. Messages may be in separate OUTPUT statements.<br>Declarations are not required. Answers must be in pseudocode: program code scores 0. |

**Total: 22 marks**