MODULE M PROGRAMMING · M26

Library routines and random numbers: mark scheme

22 marks · Cambridge 0478 syllabus 8.1.7 (library routines MOD, DIV, ROUND, RANDOM), with 7.7 (trace tables), 7.5 (validation), 7.9 (write algorithms), 8.1.4 (selection, iteration, totalling)

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. Algorithm answers must be in pseudocode; marks are for the features listed, not for one exact algorithm. Accept = or <- for ←, and MOD/DIV in the operator form (`47 MOD 6`). Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 8.1.7	Any **two** from: • a pre-written / ready-made piece of code // a routine that has already been written • that has already been tested • stored in a (program) library // supplied with the programming language / translator / development environment • called / used by name in a program (when needed) // the programmer does not need to write the code • carries out a common task, e.g. MOD / DIV / ROUND / RANDOM	**2**	Max 1 for naming examples only (e.g. "MOD and RANDOM"). Do not accept "a library of books" or "a place where programs are stored" alone. Credit any other valid point.
2(a) 0478 8.1.7	One mark for each correct value: • `ROUND(3.14159, 2)`: 3.14 • `ROUND(18.96, 0)`: 19 • `DIV(47, 6)`: 7 • `MOD(47, 6)`: 5	**4**	Correct answer only. Accept 19.0. Not 3.141 or 3.1 for row 1; not 18 or 18.9 / 19.9 for row 2; not 7.83 or "7 r 5" for DIV; not 0.83 for MOD.
3(a) 0478 8.1.7	Smallest 3 **and** largest 7	**1**	Both needed. RANDOM() * 4 gives 0 to 4; ROUND gives 0 to 4; + 3 gives 3 to 7. Not 4 and 7, or 3 and 6.
3(b) 0478 8.1.7	One mark per bullet point: • `RANDOM()` multiplied by 49 and 1 added (gives the range 1 to 50) • `ROUND( ... , 0)` used to give a whole number, with the result assigned to `Ticket` `Ticket ← ROUND(RANDOM() * 49, 0) + 1`	**2**	Accept `ROUND(RANDOM() * 49 + 1, 0)` // `ROUND(1 + RANDOM() * 49, 0)`. `ROUND(RANDOM() * 50, 0)` or `ROUND(RANDOM() * 50, 0) + 1` (wrong range): award bullet 2 only. No ROUND (e.g. `RANDOM() * 49 + 1`): award bullet 1 only. Program code (e.g. `randint`): 0.

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>3(c)
0478 8.1.7</td><td>One mark per bullet point:
• a number from 0 to 10 (inclusive) // between 0.0 and 10.0
• rounded to one decimal place // in steps of 0.1 // e.g. 6.3</td><td>2</td><td>"A random number" alone: 0. "A whole number from 0 to 10": bullet 1 only. Not "0 to 1".</td></tr>
<tr><td>4(a)
0478 8.1.7, 7.7</td><td>One mark for each correct column:
• Hour: 1, 2, 3, 4
• Parcels: 18, 27, 11, 25
• Total: (0,) 18, 45, 56, 81
• OUTPUT: Busy hour 2 // 6 full crates and 9 parcels left over (only these two, in this order)
<table><tr><th>HOUR</th><th>PARCELS</th><th>TOTAL</th><th>OUTPUT</th></tr><tr><td></td><td></td><td>0</td><td></td></tr><tr><td>1</td><td>18</td><td>18</td><td></td></tr><tr><td>2</td><td>27</td><td>45</td><td>Busy hour 2</td></tr><tr><td>3</td><td>11</td><td>56</td><td></td></tr><tr><td>4</td><td>25</td><td>81</td><td></td></tr><tr><td></td><td></td><td></td><td>6 full crates and 9 parcels left over</td></tr></table></td><td>4</td><td>Working: 0.42 × 20 = 8.4 → 8; 0.87 × 20 = 17.4 → 17; 0.06 × 20 = 1.2 → 1; 0.73 × 20 = 14.6 → 15; each + 10. Values must be in the order shown; ignore which row each value is on. Ignore quotation marks in OUTPUT; FT the candidate's own final Total for the last output. 25 is not > 25, so there is no output for hour 4: an extra "Busy hour 4" loses the OUTPUT mark.</td></tr>
<tr><td>4(b)
0478 8.1.7</td><td>Smallest 10 and largest 30</td><td>1</td><td>Both needed. Not 11 and 30, or 10 and 29.</td></tr>
<tr><td>4(c)
0478 8.1.7</td><td>Any one from:
• RANDOM() returns a real number / a number with decimal places (between 0 and 1) ... so ROUND makes the number of parcels a whole number / integer
• there cannot be part of a parcel // the number of parcels must be a whole number / integer</td><td>1</td><td>"To round the number" alone: 0. Accept "to remove the decimal places".</td></tr>
</table>

QUESTION	ANSWER	MARKS	GUIDANCE

5(a)
0478 8.1.7, 7.5, 7.9, 8.1.4
	One mark per bullet point. Max 5:
	• `RANDOM()` scaled to give the range 1 to 500 (e.g. `* 499` and `+ 1`) ...
	• ... with `ROUND( ... , 0)` to give a whole number, stored in `WinningNumber`
	• input of the ticket number inside a condition-controlled loop
	• loop ends only when the ticket number is >= 1 AND <= 500 (correct closing statement, e.g. UNTIL / ENDWHILE)
	• IF that compares the ticket number with `WinningNumber` and outputs "Winner" when they are equal
	• otherwise outputs a message that includes `WinningNumber`
	Example answer:
	`WinningNumber ← ROUND(RANDOM() * 499, 0) + 1`
	`REPEAT`
	`  OUTPUT "Enter your ticket number (1 to 500)"`
	`  INPUT TicketNumber`
	`UNTIL TicketNumber >= 1 AND TicketNumber <= 500`
	`IF TicketNumber = WinningNumber`
	`  THEN`
	`  OUTPUT "Winner"`
	`  ELSE`
	`  OUTPUT "Not a winner. The winning number was ", WinningNumber`
	`ENDIF`
		5	Allow a WHILE loop with the input repeated inside it (e.g. input before the loop, then `WHILE TicketNumber < 1 OR TicketNumber > 500 DO` ... input ... `ENDWHILE`). A single IF with one re-entry (no loop): do not award bullets 3 or 4. Condition missing the variable on one side (e.g. `>= 1 AND <= 500`): do not award bullet 4. Random number generated inside the loop: do not penalise. Program code (not pseudocode): 0. Ignore missing DECLARE statements.

Total: 22 marks