MODULE M PROGRAMMING · M5

Arithmetic operators, including MOD, DIV and exponentiation: mark scheme

21 marks · Cambridge 0478 syllabus 8.1.4(f) (arithmetic operators + − / * ^ MOD DIV; precedence), 8.1.7 (library routines MOD, DIV), with 7.7 (trace tables), 8.1.4(d) (counting)

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. MOD and DIV may be written in the syllabus function form `MOD(X, 2)` or in the operator form `X MOD 2`. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 8.1.4(f), 8.1.7	One mark for each correct value: • `MOD(23, 4)`: 3 • `DIV(23, 4)`: 5 • `2 ^ 5`: 32 • `15 / 4`: 3.75	**4**	Correct answer only. Not 5.75 or "5 r 3" for DIV; not 10 for 2 ^ 5; not 3 or "3 r 3" for 15 / 4 (accept 3¾).
2(a) 0478 8.1.4(f), 8.1.7	`FullLanes ← DIV(Cars, 9)`	**1**	Accept `Cars DIV 9`. Accept = or <- for ←. Not `Cars / 9` (it does not give a whole number of lanes).
2(b) 0478 8.1.4(f), 8.1.7	`LastLane ← MOD(Cars, 9)`	**1**	Accept `Cars MOD 9` // `Cars − FullLanes * 9`. Accept = or <- for ←.
2(c) 0478 8.1.4(f), 8.1.7	• FullLanes: 6 • LastLane: 4	**2**	One mark each. Values must be against the right identifier. DIV(58, 9) = 6 and MOD(58, 9) = 4.

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>3(a)
0478 8.1.4(f), 8.1.7, 7.7</td><td>One mark for each correct column:
• Number: 4071, 407, 40, 4, 0
• Digit: 1, 7, 0, 4
• Total: 0, 1, 8, (8,) 12
• OUTPUT: Accepted
<table><tr><th>NUMBER</th><th>DIGIT</th><th>TOTAL</th><th>OUTPUT</th></tr><tr><td>4071</td><td></td><td>0</td><td></td></tr><tr><td>407</td><td>1</td><td>1</td><td></td></tr><tr><td>40</td><td>7</td><td>8</td><td></td></tr><tr><td>4</td><td>0</td><td>8</td><td></td></tr><tr><td>0</td><td>4</td><td>12</td><td>Accepted</td></tr></table></td><td>4</td><td>Values must be in the order shown; ignore the exact row each value is on and repeated values. Digit must include the 0. Total: the repeated 8 may be shown or left out. Number must end at 0. OUTPUT: ignore quotation marks; must be the only output.</td></tr>
<tr><td>3(b)
0478 8.1.4(f), 7.7</td><td>Rejected</td><td>1</td><td>7 + 6 = 13 and MOD(13, 3) = 1. Ignore quotation marks.</td></tr>
<tr><td>4(a)
0478 8.1.4(f)</td><td>19</td><td>1</td><td>Accept 19.0. Not 9 (that is the correct mean, not what this statement gives).</td></tr>
<tr><td>4(b)
0478 8.1.4(f)</td><td>Any one from:
• division has higher precedence than addition // division is done before addition
• only <code>Score3</code> is divided by 3 // the total of the three scores is not divided by 3</td><td>1</td><td>"It is in the wrong order" alone: not enough. "It needs brackets" alone: not enough without saying what is divided.</td></tr>
<tr><td>4(c)
0478 8.1.4(f)</td><td><code>Mean ← (Score1 + Score2 + Score3) / 3</code></td><td>1</td><td>Brackets must enclose all three scores. Accept = for ←. Accept an equivalent correct statement, e.g. <code>Mean ← Score1 / 3 + Score2 / 3 + Score3 / 3</code>.</td></tr>
</table>

QUESTION	ANSWER	MARKS	GUIDANCE

5(a)
0478 8.1.4(f), 8.1.7, 8.1.4(d)
	One mark per bullet point. Max 5:
	• counter initialised to 0 before the loop
	• loop that runs exactly 10 times, with a correct closing statement
	• input of a parcel number inside the loop
	• correct test for even using MOD, e.g. `MOD(ParcelNumber, 2) = 0` // `MOD(ParcelNumber, 2) <> 1`
	• counter increased by 1 only when the parcel number is even
	• count output with a suitable message after (outside) the loop
	Example answer:
	`EvenCount ← 0`
	`FOR Parcel ← 1 TO 10`
	`  OUTPUT "Enter the parcel number"`
	`  INPUT ParcelNumber`
	`  IF MOD(ParcelNumber, 2) = 0`
	`  THEN`
	`  EvenCount ← EvenCount + 1`
	`  ENDIF`
	`NEXT Parcel`
	`OUTPUT "Parcels for bay A: ", EvenCount`
		5	Allow a WHILE or REPEAT loop with a counter that gives exactly 10 inputs. Test not using MOD (e.g. `DIV` or `/`): do not award bullet 4. Output inside the loop: do not award the last bullet. Program code (not pseudocode): 0. Ignore missing DECLARE statements.

Total: 21 marks