MODULE M PROGRAMMING · M26

Random numbers: mark scheme

20 marks · OCR J277 spec 2.2.3 (random number generation: create and use random numbers in a program), with 2.2.2 (data types), 2.2.3 (arrays), 2.2.1 (selection, iteration), 2.1.2 (trace tables)

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative answers. Words in (brackets) are not needed for the mark. **...** means the second point depends on the first. Code answers are marked for logic, not exact syntax: any high-level language or OCR Exam Reference Language is accepted (e.g. Python `random.randint(1, 20)` for `random(1, 20)`); penalise a repeated syntax slip once, then follow through. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.2.3, 2.2.2	• `seat`: integer • `seat`: smallest 1 and largest 40 • `wind`: real • `wind`: smallest −3.0 and largest 3.0	**4**	1 mark per bullet. Accept float for real; accept int for integer. Both values needed for each range mark; accept −3 and 3. Do not accept 2 or 39 (random is inclusive at both ends), or 0 as the smallest `wind` value.
2(a) J277 2.2.3	`code = random(100, 999)`	**1**	Must be stored in code, with 100 and 999 as the limits. Accept a high-level language equivalent, e.g. `code = random.randint(100, 999)`. Do not accept `random(100, 1000)` in ERL or `randint(100, 1000)` in Python. Accept `randrange(100, 1000)` (Python's upper limit is excluded).
2(b) J277 2.2.3	`pause = random(0.5, 2.5)`	**1**	Must be stored in pause, with 0.5 and 2.5 as the limits. Accept `random.uniform(0.5, 2.5)` or another correct high-level method. Do not accept `random(1, 2)` or any version that gives whole numbers only.

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>3(a)
J277 2.2.3</td><td>1 mark each to max 2:
• The fact at index 0 / the first fact is never shown
• When 6 is chosen there is no element at index 6 / the index is outside the array ... so the program crashes / gives an error / cannot print a fact</td><td>2</td><td>"It picks the wrong fact" alone: not enough. Accept "index out of range" for the second bullet. Do not accept "it will not be random".</td></tr>
<tr><td>3(b)
J277 2.2.3</td><td><code>pick = random(0, 5)</code></td><td>1</td><td>Accept <code>pick = random(0, facts.length - 1)</code>. Accept <code>pick = random(1, 6) - 1</code>. Do not accept a change to line 02 only.</td></tr>
<tr><td>4(a)
J277 2.2.3, 2.1.2</td><td>1 mark per correct column:
• growth: 3, 4, 1, 2
• height: (5,) 8, 12, 13, 15
• days: (0,) 1, 2, 3, 4
• Output: Ready after 4 days (once, at the end)
<table><tr><th>GROWTH</th><th>HEIGHT</th><th>DAYS</th><th>OUTPUT</th></tr><tr><td></td><td>5</td><td>0</td><td></td></tr><tr><td>3</td><td>8</td><td>1</td><td></td></tr><tr><td>4</td><td>12</td><td>2</td><td></td></tr><tr><td>1</td><td>13</td><td>3</td><td></td></tr><tr><td>2</td><td>15</td><td>4</td><td>Ready after 4 days</td></tr></table></td><td>4</td><td>Values must be in the order shown; ignore which row each value is on and repeated values. Output: ignore quotation marks and small spacing slips; FT the candidate's final <code>days</code> value. Do not award the Output mark if an output is written on every row.</td></tr>
<tr><td>4(b)
J277 2.2.3</td><td>3</td><td>1</td><td>Correct answer only. The largest growth is 4 a day: 5 + 4 + 4 = 13 (not enough), 13 + 4 = 17. Do not accept 1, 2 or 10.</td></tr>
</table>

5(a)
J277 2.2.3, 2.2.1
	1 mark each to max 6:
	• Asks for / inputs the number of customers and stores it (as an integer)
	• Loop that runs once for each customer (uses the number input)
	• Generates a random whole number from 1 to 20 inside the loop and stores / uses it
	• Checks whether the number is 20 and outputs "Free drink" ...
	• ... otherwise outputs the number
	• Counts the free drinks (count set to 0 before the loop and increased by 1 only for a 20) ...
	• ... and outputs the count after the loop
	
	Example answer (ERL):
	`customers = int(input("Number of customers: "))`
	`freeDrinks = 0`
	`for c = 1 to customers`
	`    number = random(1, 20)`
	`    if number == 20 then`
	`        print("Free drink")`
	`        freeDrinks = freeDrinks + 1`
	`    else`
	`        print(number)`
	`    endif`
	`next c`
	`print("Free drinks won: " + str(freeDrinks))`
		6	Allow any logically valid method, e.g. a while loop with a counter. Random number generated once, before the loop: do not award bullet 3. `random(1, 21)` in ERL / `randint(1, 21)` in Python: do not award bullet 3. Count output inside the loop: do not award the last bullet. Accept `>= 20` for `== 20`. Missing `int()` / casting: do not penalise. Flowcharts or written English: 0.

Total: 20 marks