MODULE M PROGRAMMING · M24

Parameters and arguments: mark scheme

20 marks · OCR J277 spec 2.2.3 (the use of functions and procedures; how to use sub programs to produce structured code; arrays passed into sub programs) and 2.3.1 (maintainability: use of sub programs)

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative answers. Words in (brackets) are not needed for the mark. Code answers are marked for logic, not exact syntax: accept OCR Exam Reference Language or any high-level language, and allow any logically valid method. Minor syntax slips are not penalised; a repeated slip is penalised once then follow through. Capitals in keywords are ignored. Students do not need to use the word "argument" anywhere: "the value passed (to the parameter)" is equally creditworthy.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.2.3	`bill` and `percent`	**1**	Both needed for the mark. Do not accept `40` and `10` (these are the values passed in the call). Do not accept `extra` or `total` (variables, not parameters).
1(b) J277 2.2.3	(Line) 06	**1**	Accept 6. Do not accept 01 (that is where the function is defined).
1(c) J277 2.2.3	10	**1**	The second value in the call goes to the second parameter.
1(d) J277 2.2.3	44	**1**	Accept 44.0. Working: extra = 40 × 10 / 100 = 4; 40 + 4 = 44. Do not accept 4 or 440.
2(a) J277 2.2.3	Used in the same way as a variable: True Can only ever have one parameter: False Values matched to parameters in the order written: True	**3**	1 mark per correct row. Both boxes ticked in a row: 0 for that row.
3(a) J277 2.2.3	Line 09: `bolts low: order 8` Line 10: `nuts OK` Line 11: `washers OK`	**3**	1 mark per correct row. Ignore quotation marks and minor spacing. Line 09: 20 − 12 = 8. Line 11: 15 < 15 is False, so the `else` branch runs; `washers low: order 0` = 0 marks for that row.

QUESTION	ANSWER	MARKS	GUIDANCE
3(b) J277 2.2.3	1 mark each to max 2: The values are matched to the parameters by position / in order ... ... so `item` gets 20 / `quantity` gets `"bolts"` / `minimum` gets 12 / the values go into the wrong parameters A string (`"bolts"`) would be compared with a number on line 02 / the data types do not match what the procedure expects / it would cause an error (or wrong output)	**2**	"The values are in the wrong order" alone = 1. For 2 the answer must say what goes wrong as a result (wrong parameter gets the value, data type mismatch or error). Accept "should be `stockMessage("bolts", 12, 20)`" as showing the order point (1). Credit any other valid answer.
4(a) J277 2.2.3 / 2.3.1	1 mark each to max 2: The code is written only once / less code to write / shorter program The same procedure is reused for every room by passing a different room number Easier to maintain / a change to the sign only has to be made in one place Fewer chances of errors / only one piece of code to test / easier to debug Easier to read / understand / more structured program	**2**	Two different benefits needed. "Quicker" or "more efficient" alone = NE (say what is quicker, e.g. quicker to write). Credit any other valid answer.

5(a)
J277 2.2.3
	1 mark each to max 4:
	A counter is set to 0 (before the loop)
	A loop that visits every element of `readings` (e.g. `for i = 0 to readings.length - 1` / for each value)
	Each element is compared with the parameter `limit` using >, and the counter is increased by 1 when it is greater
	The count is returned (after the loop has finished)
	Example:
	`function countAbove(readings, limit)`
	`    count = 0`
	`    for i = 0 to readings.length - 1`
	`        if readings[i] > limit then`
	`            count = count + 1`
	`        endif`
	`    next i`
	`    return count`
	`endfunction`
		4	Must use the parameters: a fixed value (e.g. `> 30`) in place of `limit` loses the comparison mark, and inputting a new value into `limit` or `readings` (overwriting the parameter) loses the mark it affects. `>=` loses the comparison mark. Accept a loop to 6 / `range(7)` or `range(len(readings))` (a week has 7 values); allow an off-by-one bound explained by Python `range`. Return inside the loop loses the return mark. Printing the count instead of returning it loses the return mark. Do not accept flowcharts, structured English or bullet points.

5(b) J277 2.2.3	`countAbove(dayTemps, 30)`: function called with the array and 30, in that order The returned value is stored in `hotDays` (`hotDays = ...`) Example: `hotDays = countAbove(dayTemps, 30)`	**2**	`countAbove(30, dayTemps)` (wrong order) loses the first mark. Allow follow through if 5(a) used different parameter names. Do not accept `hotDays == countAbove(...)` (comparison, not assignment) for the second mark. A call on its own with no assignment = max 1.

Total: 20 marks