MODULE M PROGRAMMING · M23

Functions and return values: mark scheme

22 marks · OCR J277 spec 2.2.3 (the use of functions and procedures; sub programs), with 2.2.1 (MOD, DIV, selection) used in context

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 later mark depends on the earlier one. 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 (e.g. == for assignment) is penalised once then follow through. Capitals in keywords are ignored. Do not accept maths symbols (×, ≥, ≤) or English ("is more than") in place of operators.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.2.3	Named sub program, defined once, called many times: Both Uses `return` to send a value back: Function Carries out a task but does not return a value: Procedure	**3**	1 mark per correct row. More than one tick in a row: 0 for that row.
2(a) J277 2.2.3	04 / 4	**1**	Accept `return cost` written out instead of the number.
2(b) J277 2.2.3	60	**1**	Working: 3 × 4 = 12 pages; 12 × 5 = 60. Accept 60p. Do not accept £60 or 0.6 on its own (the function returns pence).
2(c) J277 2.2.3	Two calls with the arguments in the correct order: `printCost(2, 5)` and `printCost(4, 3)` ... ... results added and assigned to `orderTotal` Example: `orderTotal = printCost(2, 5) + printCost(4, 3)`	**2**	(Value 50 + 60 = 110.) Arguments swapped in one or both calls (`printCost(5, 2)`) gives the same value here but does not match the parameters (pages, copies): do not award MP1, allow MP2 FT. Do not accept `orderTotal = 110` (no function used) or two statements (the question asks for one). Do not accept == for assignment.
3(a) J277 2.2.3 / 2.2.1	`boxesNeeded(16)`: 2 `boxesNeeded(17)`: 3 `boxesNeeded(5)`: 1	**3**	1 mark per correct row. Working: 16 DIV 8 = 2, 16 MOD 8 = 0 so no extra box; 17 DIV 8 = 2, 17 MOD 8 = 1 so 2 + 1 = 3; 5 DIV 8 = 0, 5 MOD 8 = 5 so 0 + 1 = 1.

QUESTION	ANSWER	MARKS	GUIDANCE
3(b) J277 2.2.3	20: `Use a trolley` 16: `Carry by hand`	**2**	Working: `boxesNeeded(20)` returns 3 and 3 > 2 is True; `boxesNeeded(16)` returns 2 and 2 > 2 is False. Ignore quotation marks. Do not accept the number of boxes (3 / 2) as the output.
4(a) J277 2.2.3	Line 04 (identified) ... ... `return mean` Line 06 (identified) ... ... `result = meanReading(4, 9, 8)`	**4**	The correction mark for each error needs the right line number. Line 04: accept `return total / 3`; do not accept any line that still prints instead of returning. Line 06: the call must be kept and its returned value stored in `result`. Do not accept == for assignment. Do not accept changing line 07 only (e.g. calling the function inside the print): the program must store the mean in `result`. Do not accept changes to lines 01 to 03.

QUESTION	ANSWER	MARKS	GUIDANCE

5(a)
J277 2.2.3 / 2.2.1
	1 mark each to max 6:
	Function defined with a parameter (e.g. `function isComfortable(temp)` ... `endfunction` / `def isComfortable(temp):`)
	Correct check of both limits inside the function: `temp >= 18 AND temp <= 24`
	Returns `True` when comfortable and `False` otherwise (must use return)
	Inputs the room temperature in the main program and stores it in a variable
	Calls `isComfortable()` with the entered temperature as the argument ...
	... uses the returned value in selection to output `Comfortable` and `Adjust heating` in the correct cases
	Example:
	`function isComfortable(temp)`
	`    if temp >= 18 AND temp <= 24 then`
	`        return True`
	`    else`
	`        return False`
	`    endif`
	`endfunction`
	
	`roomTemp = float(input("Enter the room temperature"))`
	`if isComfortable(roomTemp) == True then`
	`    print("Comfortable")`
	`else`
	`    print("Adjust heating")`
	`endif`
		6	Accept `return temp >= 18 AND temp <= 24` for MP2 and MP3. Temperatures can include decimals, so `temp > 17` or `temp < 25` lose MP2, as do `temp > 18` or `temp < 24`. Each side must be a full comparison (`temp >= 18 AND <= 24` loses MP2). Accept `if isComfortable(roomTemp) then` (no `== True`), or storing the returned value in a variable first. Do not penalise missing or different casting of the input. MP1: a parameter that is overwritten by an `input` inside the function does not get MP1. MP3: printing inside the function instead of returning does not get MP3. Accept the function written after the main program. Accept any variable names except reserved words (e.g. `input`). Do not accept flowcharts, structured English or bullet points.

Total: 22 marks