MODULE M PROGRAMMING · M8

Selection: IF, ELSE IF and ELSE: mark scheme

20 marks · Cambridge 0478 syllabus 8.1.4(b) (selection: IF statements), 8.1.5 (nested statements), 8.1.4(f) (relational and logical operators), 7.7 (dry-run outputs), 7.8 (identify and correct errors), 7.9 (write and amend algorithms)

How to use. One mark per bullet point up to the max shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Algorithms must be in pseudocode: program code is not credited. Ignore minor syntax slips (e.g. `<-` or = for ←, THEN on the same line as IF, lower-case keywords) if the meaning is clear. Conditions must be precise: `Height > 2.0` is acceptable, "Height is greater than 2" is not. Ignore quotation marks round output text. Credit any other valid method that meets the requirement.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 8.1.4(b)	B	**1**	More than one tick: 0.
1(b) 0478 8.1.4(b)	When the condition (in the IF) is false // when the condition is not met	**1**	Allow "when the THEN statements are not run" // "when the condition gives No". Do not accept "when the answer is no" with no reference to the condition.

2(a) 0478 8.1.4(b), 8.1.4(f), 7.9	One mark for each:
	`IF Charge < 15` (with THEN)
	`OUTPUT "Return to base"` inside the IF and `ENDIF` to close it
	`IF Charge < 15`
	`  THEN`
	`  OUTPUT "Return to base"`
	`ENDIF`
		2	Do not accept `Charge <= 15` or `Charge > 15`. An ELSE clause that outputs something: do not award the second mark. An empty ELSE clause: ignore.

3(a) 0478 7.7, 8.1.5	One mark for each correct output: 90: `Pump low` 95: `Pump off` 40: `Pump low` 12: `Pump high`	**4**	Output text must be copied exactly (ignore capitals). More than one output in a row: 0 for that row.
3(b) 0478 8.1.4(b), 8.1.5	The condition on line 02 / `Level > 90` is true (for 95), so only the THEN clause runs // line 06 is inside the ELSE clause (of the first IF), which only runs when `Level` is 90 or less	**1**	Do not accept "because the pump is off" or "95 is more than 90" alone (must link to which clause runs).
4(a) 0478 7.7, 8.1.5	`Silver`	**1**	Do not accept `Gold`.

QUESTION	ANSWER	MARKS	GUIDANCE
4(b) 0478 7.8, 8.1.5	Any value of 250 or more is also 100 or more, so the condition on line 02 is true and `Silver` is given // the test on line 06 is in the ELSE clause, which only runs for values below 100, and these can never be 250 or more	**1**	Do not accept "there is a logic error" or "the order is wrong" alone.

4(c)
0478 7.8, 7.9, 8.1.5
	One mark for each:
	First IF tests `Points >= 250` and sets `Level` to `"Gold"` in its THEN clause
	Nested IF in the ELSE clause tests `Points >= 100` and sets `Level` to `"Silver"`
	Inner ELSE sets `Level` to `"Standard"` and both IF statements are closed with ENDIF
	`IF Points >= 250`
	`  THEN`
	`    Level ← "Gold"`
	`  ELSE`
	`    IF Points >= 100`
	`      THEN`
	`        Level ← "Silver"`
	`      ELSE`
	`        Level ← "Standard"`
	`    ENDIF`
	`ENDIF`
		3	Allow `> 249` and `> 99`.
			Allow tests from the lowest band up, e.g. `IF Points < 100` … `Standard`, then nested `IF Points < 250` … `Silver` … ELSE `Gold`.
			Allow three separate IF statements with full ranges, e.g. `IF Points >= 100 AND Points < 250`: one mark for each correct level, ENDIFs needed. Do not accept `Points >= 100 AND < 250`.

QUESTION	ANSWER	MARKS	GUIDANCE

5(a)
0478 7.9, 8.1.4(b), 8.1.5, 8.1.3
	One mark per bullet point. Max 6:
	`INPUT Height` (any sensible variable name)
	IF tests `Height <= 2.0` and outputs `Lane A`
	IF tests `Height <= 3.5` in the ELSE clause of the Lane A test // `Height > 2.0 AND Height <= 3.5`, and outputs `Lane B`
	IF tests `Height <= 4.5` in the ELSE clause of the Lane B test // `Height > 3.5 AND Height <= 4.5`, and outputs `Lane C`
	Outputs `Too tall` in the final ELSE // when `Height > 4.5`
	Only one output for any height: IF statements nested in the ELSE clauses // separate IF statements with ranges that do not overlap
	Every IF has THEN (and ELSE where used) and a matching ENDIF
	`INPUT Height`
	`IF Height <= 2.0`
	`  THEN`
	`    OUTPUT "Lane A"`
	`  ELSE`
	`    IF Height <= 3.5`
	`      THEN`
	`        OUTPUT "Lane B"`
	`      ELSE`
	`        IF Height <= 4.5`
	`          THEN`
	`            OUTPUT "Lane C"`
	`          ELSE`
	`            OUTPUT "Too tall"`
	`        ENDIF`
	`    ENDIF`
	`ENDIF`
		6	Boundaries must be exact: 2.0 is Lane A, 3.5 is Lane B, 4.5 is Lane C. Allow 2 for `2.0`. Allow tests in reverse order (e.g. `Height > 4.5` first). The open tests `Height <= 3.5` and `Height <= 4.5` earn the Lane B and Lane C bullets only when nested in the ELSE clause of the previous IF; separate IF statements must use full ranges (e.g. `Height > 2.0 AND Height <= 3.5`). Separate IF statements with open conditions give more than one output: do not award the Lane B, Lane C or "only one output" bullets. Each side of AND must be a full comparison. A prompt with the input is not required. Do not credit CASE for the range tests.

Total: 20 marks