Task	Expected response		Additional guidance	Marks available	
2	**Software design and development**				
2a	Array used to store readings			1	Implementation (15)
	Fixed loop repeating 5 times (to enter readings)			1	
	Input Validation	conditional loop used		1	
		correct condition for valid data	Until: reading >= 0 AND reading <= 100 or While: reading < 0 OR reading > 100	1	
		input of reading	Award 1 mark if not implemented within input validation loop	1	
		error message		1	
	Round reading to 0 decimal places			1	
	1 mark for correct conditions **1 mark** for using rounded reading		Strong: Reading > 80 Poor: Reading < 30 (Medium: Reading>=30 and <=80)	2	
	Single variable used to store signal pattern		Variable names may differ in code	1	
	Pattern concatenated			1	
	Suitable message and signal pattern output			1	
	Five (rounded) readings displayed as output with suitable messages within a loop		For example: For loop = 1 to 5 Print “Reading”, loop, “is”, reading(loop) End loop	1	
	Matches design: • same top level sequence (loop 5, display, loop 5) • nested if statements (or elseif) with correct structure used to determine signal strength letter			2	

Task	Expected response	Additional guidance	Marks available	
2	**Software design and development**			
2b	The test table completed to produce the required signal pattern output (MPSPS) for **1 mark**	Test table should contain real or integer values within the following ranges: Reading 1: >=30 and <=80 Reading 2: <30 Reading 3: >80 Reading 4: <30 Reading 5: >80 Input must be numeric. Do not accept % symbols.	1	Testing (5)
	Printed evidence of one successful run of the test table data		1	
2c	Completion of extreme test data for upper and lower limits of each signal strength **1 mark** for each pair: • poor: ○ 0 ○ 29 • medium: ○ 30 ○ 80 • strong: ○ 81 ○ 100	Candidates may write values in any order. Accept any values that round to the values given. Check rounding according to language used.	3	

Task	Expected response	Additional guidance	Marks available	
2	**Software design and development**			
2d	Evaluation of the following: Fitness for purpose **(1 mark)** • comparison of their solution (code and testing) with program analysis and expected output Efficiency **(1 mark)** • efficient use of at least one coding constructs Robustness **(1 mark)** • how robust the program is, including if it copes with unexpected inputs Readability **(2 marks)** • two comments on the readability of the candidate's own code	All evaluations must contain an element of evaluation rather than simple statements of terms. For example "I have used white space to highlight structures in my program" not "I have used white space". Efficiency answers may refer to: • loops used instead of five individual inputs or outputs • single variable only required for signal pattern rather than array of characters • complex selection structure could have been used in place of separate "ifs" • array used instead of five variables for readings	5	Evaluation (5)