Task	Expected response	Additional guidance	Marks available	
2	**Software design and development**			
2a	Array used in program		1	Implementation (15)
	Use of the following variables: ♦ total hits ♦ average ♦ points	Variable names may differ in code All three variables are required for 1 mark	1	
	Fixed loop repeating six times (to enter player hits)		1	
	Input validation — conditional loop used		1	
	Input validation — correct loop condition	hits >= 0 and hits <= 30	1	
	Input validation — input of player hits	Award 1 mark if not implemented within input validation loop	1	
	Input validation — error message		1	
	Running total calculated correctly		1	
	Round function used with average		1	
	Calculation of bonus points: ♦ for one bonus point ♦ for an additional bonus point		2	
	Selection (if) used to display message showing one bonus point earned	(totalHits > 50) Output must be within selection	1	
	Selection (if) used to display message showing additional bonus point earned	(average >= 10) Output must be within selection	1	
	Selection (if) used to display message showing zero bonus points earned	(totalHits <= 50) Output must be within selection	1	
	Matches design — same sequence of events as flow chart		1	

Task	Expected response	Additional guidance	Marks available	
2	**Software design and development**			
2b	Both test tables completed to produce the required output (one bonus point or two)	Table 1 hits should total 51-59 Table 2 hits should total >=60	1	Testing (5)
	In first table, printed evidence of successful run of test data	Both inputs and outputs should be printed	1	
	In second table, printed evidence of successful run of test data	Both inputs and outputs should be printed	1	
2c	Completion of test data for input validation of player's hits for 1 mark each: ♦ extreme: 0 and 30 ♦ exceptional: any suitable, eg 1, 31	Only accept numerical answers for exceptional test data	2	
2d	Evaluation of the following for 1 mark each: ♦ whether the program is fit for purpose, including explanation of code ♦ efficient use of coding constructs ♦ how robust the program is, including if it copes with unexpected inputs Evaluation of the following for 2 marks: ♦ readability — 1 mark for each comment on the readability of the candidate's own code	Efficiency answers may refer to: ♦ two loops not required for inputs and running total ♦ single variable only required for hits if implemented in one loop ♦ complex selection structure could have been used in place of three "ifs" ♦ array used instead of six variables for hits	5	Evaluation (5)