



## Theory Questions (Mark Scheme)

Question		Suggested Solution	Mark	Marking Guidance
1	(a)	The <b>Choice</b> variable is converted to lower case to ensure the input is case-insensitive. This way, both “Y” and “y” will be treated the same, simplifying the logic. [1]	1 mark	<b>A:</b> any reasonable suggestion
	(b)	The <b>TrainingGame</b> variable determines whether the game is in training mode or random mode. In training mode, predefined numbers and targets are used, while in random mode, the numbers and targets are randomly generated. [1]	1 mark	<b>A:</b> any reasonable suggestion
2		The repeated call to <b>DisplayState()</b> updates the game display, showing the current targets, available numbers, and score after every move. [1] This helps the player track their progress and remaining resources. [1]	2 marks	1 mark for each point (MAX. 2)
5	(a)	The pattern string <code>^[0-9]+\$</code> means that the <b>Item</b> string that was input must: [1] a. Start with b. one or more numbers in the range 0–9 c. End after this digit / these digits  The function will check whether the string matches these criteria so therefore the <b>Item</b> string is a valid integer and can be cast to the integer data type for further processing. [1]	2 marks	
	(b)	Without the + character, the function would only match single-digit integers, i.e. 13 would be invalid. [1]	1 mark	<b>A:</b> any other example
7		In training mode, <b>NumbersAllowed</b> is populated with a fixed list of numbers [2, 3, 2, 8, 512]. In random mode, the list is filled by generating random numbers between 1 and <b>MaxNumber</b> inclusive until it has five elements. [2]	2 marks	
8	(a)	A stack is used to ensure operators are applied in the correct order, based on their precedence. Higher-precedence operators are applied before lower-precedence ones, and the stack helps manage this ordering. [2]	2 marks	
	(b)	After the while loop has completed, which will have removed any/all operators (currently only * or /) of a higher precedence from the stack, a selection statement is used to handle matching precedence operators. [1]  A matching precedence operator will be popped off the top of the stack onto the output prior to the new operator being pushed back onto the stack. [1]	2 marks	
9	(a)	<b>GetTarget()</b> generates a random target number between 1 and <b>MaxTarget</b> to be added to the Targets list. [1]	1 mark	
	(b)	In the training game, the <b>Targets</b> list is initialised as a list of literal integers. [1] In the random game, the <b>CreateTargets()</b> function is used to build a list containing five -1s representing empty cells followed by 15 random integers in the range 1 to 50 inclusive. [1]	2 marks	

Question		Suggested Solution	Mark	Marking Guidance
12	(a)	The <b>GameOver</b> variable is set to true when the first target in the <b>Targets</b> list is no longer available (i.e. <code>Targets[0] != -1</code> ). [1]	1 mark	
	(b)	It prevents the loop from running indefinitely, ensuring that the game ends when all relevant gameplay conditions have been met. [1]	1 mark	
13		Any 2 from: <ul style="list-style-type: none"> <li>The highest score could be stored in a file or a database. [1]</li> <li>At the start of the game, the file/database would be read to retrieve the previous high score. [1]</li> <li>After each game, if the new score exceeds the old high score, the file/database would be updated with the new value. [1]</li> </ul>	2 marks	<b>A:</b> any 2 points from 3
14	(a)	<b>CreateTargets / FillNumbers / ConvertToRPN / RemoveNumberUsed / UpdateTargets</b> [1]	1 mark	
	(b)	<b>TrainingGame</b> [1]	1 mark	
	(c)	<b>UserInput, Number</b> [1]	1 mark	
	(d)	<b>RemoveAt / Add</b> [1]	1 mark	
	(e)	<b>MaxTarget / MaxNumber / MaxNumberOfTargets</b> [1]	1 mark	
15		Any 2 from: <ul style="list-style-type: none"> <li><code>+ -</code> means 1 or more of preceding character/sequence [1]</li> <li><code>[0-9]+</code> means 1 or more digits from 0 to 9 [1]</li> <li><code>([0-9]+[\\+\\-\\*\\/])+</code> means 1 or more sequences of a number (operand) followed by an operator [1]</li> </ul>	2 marks	<b>A:</b> any 2 points from 3
16		Because regular expressions do not support recursion. [1] A regular expression cannot track the opening and closing of brackets / a regular expression has no notion of "state". [1]	1 mark	<b>A:</b> 1 from 2 points given
18		Decomposition: The program is broken into smaller tasks, each handled by specific functions. [1] Abstraction: Demonstrated by hiding the complexity of certain tasks behind clear, high-level functions that perform specific roles. [1]	2 marks	