Question	Answer	Marks

11	**Data Structures required** with names as given in the scenario:
	Arrays or lists `Grid`
	
	Requirements (techniques)
	R1 Set up game – generate random cell, clear all other cells in array, set player start position and start player moves counter (iteration, use of arrays and library routines (round and random))
	R2 Input and check move – is it valid? (input, output, iteration and selection)
	R3 Decide outcome – has move found the X? If so, give appropriate output. If not increment counter and continue. If 10 moves exceeded, give appropriate output (use of arrays, iteration, selection and output).
	
	Example 15-mark answer in pseudocode
	`// Set up game`
	`FOR Row ← 1 TO 5`
	`  FOR Column ← 1 TO 5`
	`  Grid[Row, Column] ← ''// set grid cells to be empty`
	`  NEXT Column`
	`NEXT Row`
		15

Question	Answer	Marks

11	REPEAT // not in cell 1,1
	XRow ← ROUND ((RANDOM() * 4) + 1, 0) // Random row position between 1 and 5 in GRID
	XColumn ← ROUND ((RANDOM() * 4) + 1, 0) // Random column position between 1 and 5 in GRID
	UNTIL XRow <> 1 and XColumn <> 1 // not in cell 1,1
	
	Grid [XRow, XColumn] ← 'X'
	MaxMove ← 10
	NumberMoves ← 0
	PlayerRow ← 1
	PlayerColumn ← 1
	Win ← FALSE
	
	// during game
	WHILE NumberMoves < MaxMove AND NOT Win
	MoveError ← FALSE
	OUTPUT "Please enter your move, L – Left, R – Right, U – Up or D - Down"
	INPUT UPPER(PlayerMove)
	REPEAT
	CASE OF PlayerMove
	'L' : TempColumn ← PlayerColumn – 1
	'R' : TempColumn ← PlayerColumn + 1
	'U' : TempRow ← PlayerRow – 1
	'D' : TempRow ← PlayerRow + 1
	OTHERWISE MoveError ← TRUE
	ENDCASE
	
	// check for out-of-range moves
	IF TempColumn < 1 or TempColumn > 5
	THEN
	MoveError ← TRUE
	ELSE
	PlayerColumn ← TempColumn
	ENDIF	

Question	Answer	Marks

11	IF TempRow < 1 or TempRow > 5
	  THEN
	  MoveError ← TRUE
	  ELSE
	  PlayerRow ← TempRow
	  ENDIF
	
	// check win if X Found
	  IF Grid [PlayerRow, PlayerColumn] = 'X'
	  THEN
	  OUTPUT "You Win"
	  Win ← TRUE
	  ELSE
	  IF NOT MoveError
	  THEN
	  NumberMoves ← NumberMoves + 1
	  ENDIF
	  ENDIF
	  UNTIL NOT MoveError
	ENDWHILE
	
	IF NOT Win
	  THEN
	  OUTPUT "You Lose"
	ENDIF	

Marking Instructions in italics			
AO2: Apply knowledge and understanding of the principles and concepts of computer science to a given context, including the analysis and design of computational or programming problems			
0	**1–3**	**4–6**	**7–9**
No creditable response.	At least one programming technique has been used. *Any use of selection, iteration, counting, totalling, input and output.*	Some programming techniques used are appropriate to the problem. *More than one technique seen applied to the scenario, check the list of techniques needed.*	The range of programming techniques used is appropriate to the problem. *All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check the list of techniques needed.*
	Some data has been stored but not appropriately. *Any **use** of variables or arrays or other language dependent data structures e.g. Python lists.*	Some of the data structures chosen are appropriate and store some of the data required. *More than one data structure **used** to store data required by the scenario.*	The data structures chosen are appropriate and store all the data required. *The data structures **used** store all the data required by the scenario.*

Marking Instructions in italics			
AO3: Provide solutions to problems by: • **evaluating computer systems** • **making reasoned judgements** • **presenting conclusions**			
0	**1–2**	**3–4**	**5–6**
No creditable response	Program seen without relevant comments.	Program seen with some relevant comment(s).	The program has been fully commented
	Some identifier names used are appropriate. *Some of the data structures used have meaningful names.*	The majority of identifiers used are appropriately named. *Most of the data structures used have meaningful names.*	Suitable identifiers with names meaningful to their purpose have been used throughout. *All of the data structures used have meaningful names.*
	The solution is illogical.	The solution contains parts that may be illogical	The program is in a logical order.
	The solution is inaccurate in many places. *Solution contains few lines of code with errors that attempt to perform a task given in the scenario*	The solution contains parts that are inaccurate. *Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.*	The solution is accurate. *Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.*
	The solution attempts at least one of the requirements. *Solution contains lines of code that attempt at least one task given in the scenario.*	The solution attempts to meet most of the requirements. *Solution contains lines of code that attempt most tasks given in the scenario.*	The solution meets all the requirements given in the question. *Solution performs all the tasks given in the scenario.*