MARKS

5. A new computer game is being developed. During the game, a player must avoid being captured by a group of robots.

The game will be played on a 10 by 10 grid. The player and robots can move horizontally, vertically or diagonally to an adjacent position on the grid. After each player move, the robots will move closer to the player.

A player wins the game when all the robots are destroyed. Robots will be destroyed when two or more robots collide, leaving a pile of rubble in their place. Any robot that collides with a pile of rubble will also be destroyed.

When any robot moves to the same position as the player, the player is captured, and the game is over.

(a) When the game is implemented, a 2-D array of string values is used to store the positions of the player and the robots.

(i) State two additional functional requirements for this game. **2**

(ii) Using a programming language of your choice, write code to declare a 2-D array to represent the game board. **1**

(iii) Using a programming language of your choice, write code to assign the value 'Robot' to **six** random, empty locations on the board. **3**

The grid below represents the contents of the 2-D array at the start of a game.

	Robot								Robot
							Robot		Robot
				Player					
		Robot							
				Robot					

Movement of the player is controlled using several procedures. Partially completed code for the `moveUp` procedure is shown below.

```
PROCEDURE moveUp()
   <find player row and column>
   IF <the player is not already on the top row> THEN
      <move the player up one row>
   END IF
END PROCEDURE
```

(b) Using a programming language of your choice, write code to implement this procedure. **3**

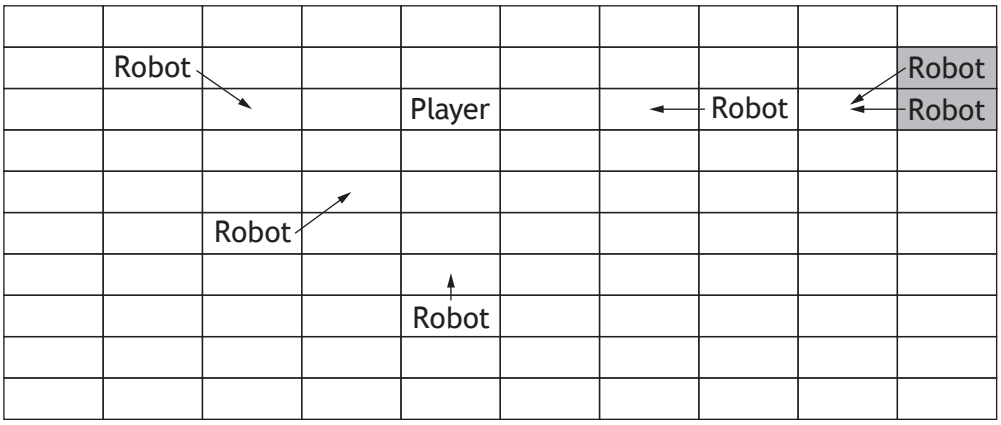
[Turn over for next question

DO NOT WRITE ON THIS PAGE

5. (continued)

Whenever a player moves up one row, all the robots will move closer to the player.

The grid below shows the new position of the player after one move, and the existing position of the robots. Each robot will move to the cell indicated by the arrowhead.



The movement of the two robots in the shaded cells will result in a collision. These robots will be destroyed and replaced with a pile of rubble.

The grid below shows the state of the game after all of the robot moves have been completed.

		Robot		Player		Robot		Rubble	
			Robot						
				Robot					

MARKS

5. (continued)

(c) The procedure `robotMove(robotX,robotY,playerX,playerY)` is used to control the movement of a robot. The parameters identify the current position of the robot and the new position of the player.

This procedure:

- clears the current position of the robot
- assigns the robot to a new position which is closer to player
- checks for a collision between robots which destroys both robots and creates a pile of rubble in their place
- checks for a collision with an existing pile of rubble which destroys the robot
- checks for the robot capturing the player.

Using pseudocode, design the procedure `robotMove`. **5**