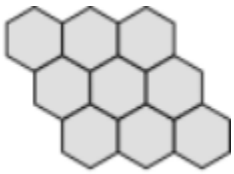
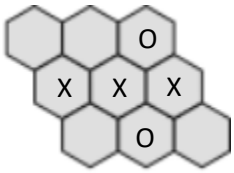
4 The Hex game involves an n × n hexagonal board. An example of a 3 × 3 Hex board is thus as follows.



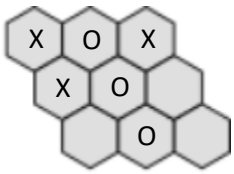
Hex is a two-player game, where one player must build a bridge that extends from left to right, and the other player must build a bridge that extends from top to bottom.

Each player takes turns to play, and may place a piece in any empty cell.

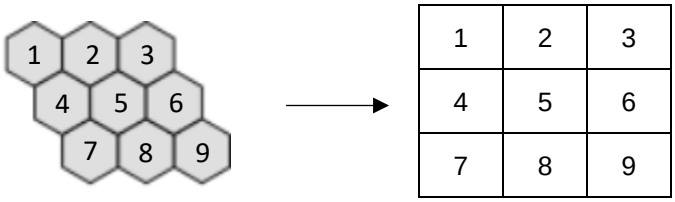
The following is an example board where the X player (going from left to right) has won the game.



The following is an example board where the O player (going from top to bottom) has won the game.



The representation for a Hex Board may be based on a standard 2-dimensional Array. Essentially, the Hex Board may be referenced as a 2-dimensional Array as follows.



You are tasked to design an object oriented programming class to store the Hex Board. This class, `HexBoard`, should be implemented as follows.

HexBoard
-board: ARRAY OF ARRAY OF STRING -turn: INTEGER
+constructor(INTEGER) +playX(INTEGER, INTEGER) +playO(INTEGER, INTEGER) +checkWinX(): BOOLEAN +checkWinO(): BOOLEAN +printBoard()

Attribute/Method	Description
`HexBoard.constructor (INTEGER)`	Initialises the `board` attribute as a 2D Array of Strings. The size of each array (both outer and inner arrays) are based on the specified integer value. The `turn` attribute is initialised as 0.
`HexBoard.playX (INTEGER, INTEGER)`	This method allows the X player to make a move by specifying the coordinates where he or she wished to place an X piece.
`HexBoard.playO (INTEGER, INTEGER)`	This method allows the O player to make a move by specifying the coordinates where he or she wished to place an O piece.
`HexBoard.checkWinX(): BOOLEAN`	This method checks the board and returns True if the X player has won the game, or else returns False.
`HexBoard.checkWinO(): BOOLEAN`	This method checks the board and returns True if the O player has won the game, or else returns False.
`HexBoard.printBoard()`	This method prints the contents of the board using the 2D Array representation.

Task 4.1

Write the program code to implement the `HexBoard` class, excluding the `checkWinX` and `checkWinO` methods. Your solution **must work for any board size**.

Evidence 19

- The program code for **Task 4.1**. [4]

Task 4.2

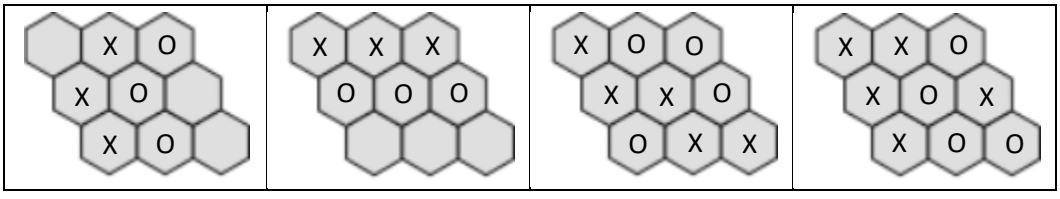
Write the program code to implement the `checkWinX` and `checkWinO` methods.

Evidence 20

- The program code for **Task 4.2**. [12]

Task 4.3

Write the program code to test the following 4 test cases for both X and O.



Note that X wins by forming a bridge from left to right, while O wins by forming a bridge from top to bottom.

Evidence 21

- The screenshots of the inputs and outputs to test each of the above cases. [4]

BLANK PAGE