

GameBoard(int numRows, int numColumns, int numToWin)

GameBoardMem(int numRows, int numColumns, int numToWin)

Input: numRows = 3 numColumns = 3 numToWin = 3	Output: getNumRows = 3 getNumColumns = 3 getNumToWin = 3 Board contains only ‘ ‘	Reason: This case is unique because it passes all of the minimum values that follow the constructor's contract as arguments Function Name: testBoardConstructor_rows3_cols3_toWin3
--	---	---

Input: numRows = 100 numColumns = 100 numToWin = 100	Output: getNumRows = 100 getNumColumns = 100 getNumToWin = 100 Board contains only ‘ ‘	Reason: This case is unique because it passes all of the maximum values that follow the constructor's contract as arguments Function Name: testBoardConstructor_rows100_cols100_toWin25
--	---	--

Input: numRows = 23 numColumns = 74 numToWin = 13	Output: getNumRows = 23 getNumColumns = 74 getNumToWin = 13 Board contains only ‘ ‘	Reason: This case is unique because it passes values that are well within the minimum and maximum values provided in the constructor's contract and are not edge values Function Name: testBoardConstructor_rows23_cols74_toWin13
---	--	--

boolean checkIfFree(int c)

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>																										Output: checkIfFree(0) = true State of the board is unchanged	Reason: This case is unique because the column being checked for a free space has no tokens at all Function Name: testCheckIfFree_emptyCol

Input: State: <table><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr></table>	X					X					X					X					X					Output: checkIfFree(0) = false State of the board is unchanged	Reason: This case is unique because the column being checked for a free space is completely full Function Name: testCheckIfFree_fullCol
X																											
X																											
X																											
X																											
X																											

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr></table>						X					X					X					X					Output: checkIfFree(0) = true State of the board is unchanged	Reason: This case is unique because the column being checked for a free space is almost full but has a single remaining free space Function Name: testCheckIfFree_almostFullCol
X																											
X																											
X																											
X																											

boolean checkHorizWin(BoardPosition pos, char p)

Input: State: (number to win = 4) <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>X</td><td>O</td></tr></table> pos.getRow = 1 pos.getCol = 2 p = 'X'																X	X	X	X		O	O	O	X	O	Output: checkHorizWin = true State of the board is unchanged	Reason: This case is unique because the last x was placed in the middle of the row of 4 consecutive X tokens instead of on the end, so tokens must be checked on both sides of it Function Name: testCheckHorizWin_win_last_marker_middle
X	X	X	X																								
O	O	O	X	O																							
Input: State: (number to win = 4) <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>X</td><td>O</td></tr></table> pos.getRow = 1 pos.getCol = 0 p = 'X'																X	X	X	X		O	O	O	X	O	Output: checkHorizWin = true State of the board is unchanged	Reason: This case is unique because the last x was placed on the left end of the row of 4 consecutive X tokens, so tokens must be checked only to the right side of it Function Name: testCheckHorizWin_win_last_marker_left
X	X	X	X																								
O	O	O	X	O																							

Input: State: (number to win = 4) <table border="1"><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>X</td><td>O</td></tr></table> pos.getRow = 1 pos.getCol = 4 p = 'X'																	X	X	X	X	O	O	O	X	O	Output: checkHorizWin = true State of the board is unchanged	Reason: This case is unique because the last x was placed on the right end of the row of 4 consecutive X tokens, so tokens must be checked only to the left side of it Function Name: testCheckHorizWin_win_last_marker_right
	X	X	X	X																							
O	O	O	X	O																							

<p>Input:</p> <p>State: (number to win = 4)</p> <table border="1"><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>X</td><td></td><td>X</td><td>X</td></tr><tr><td>O</td><td>O</td><td>O</td><td>X</td><td>O</td></tr></table> <p>pos.getRow = 1 pos.getCol = 4 p = 'X'</p>																X	X		X	X	O	O	O	X	O	<p>Output:</p> <p>checkHorizWin = true</p> <p>State of the board is unchanged</p>	<p>Reason:</p> <p>This case is unique because the last X was placed on a row of the board that has 4 non-consecutive X tokens, so a win does not occur</p> <p>Function Name: testCheckHorizWin_noWin</p>
X	X		X	X																							
O	O	O	X	O																							

boolean checkVertWin(BoardPosition pos, char p)

Input: State: (number to win = 4) <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td></tr></table> pos.getRow = 2 pos.getCol = 0 p = 'O'											O	X				O	X				O	X				Output: checkVertWin = false State of the board is unchanged	Reason: This case is unique because there are not enough of the player's tokens stacked on top of one another to result in a win Function Name: testCheckVertWin_noWin
O	X																										
O	X																										
O	X																										

Input: State: (number to win = 4) <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td></tr></table> pos.getRow = 3 pos.getCol = 1 p = 'X'							X				O	X				O	X				O	X				Output: checkVertWin = true State of the board is unchanged	Reason: This case is unique because the player has placed the required number of tokens on top of one another and the bottom of the chain of tokens resulting in the win touches the bottom of the board Function Name: testCheckVertWin_win_tokens_touch_bottom
	X																										
O	X																										
O	X																										
O	X																										

<p>Input:</p> <p>State: (number to win = 4)</p> <table><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td></tr></table> <p>pos.getRow = 4 pos.getCol = 1 p = 'X'</p>								X						X					O	X					O	X					O	O	X				<p>Output:</p> <p>checkVertWin = true</p> <p>State of the board is unchanged</p>	<p>Reason:</p> <p>This case is unique because the player has placed the required number of tokens on top of one another and there is a different player's token below the bottom token in the chain resulting in the win</p> <p>Function Name: testCheckVertWin_win_other_player_below</p>
	X																																					
	X																																					
O	X																																					
O	X																																					
O	O	X																																				

<p>Input:</p> <p>State: (number to win = 4)</p> <table><tr><td></td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td></tr><tr><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td></tr></table> <p>pos.getRow = 5 pos.getCol = 1 p = 'X'</p>		X						X						X					O	X					O	O	X				O	O	X				<p>Output:</p> <p>checkVertWin = true</p> <p>State of the board is unchanged</p>	<p>Reason:</p> <p>This case is unique because the player has placed the required number of tokens on top of one another and the chain was completed with a token that resulted in the column becoming full</p> <p>Function Name: testCheckVertWin_win_full_col</p>
	X																																					
	X																																					
	X																																					
O	X																																					
O	O	X																																				
O	O	X																																				

boolean checkDiagWin(BoardPosition pos, char p)

Input: State: (number to win = 4) <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td></tr><tr><td></td><td>O</td><td>X</td><td></td><td></td></tr><tr><td></td><td>O</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>O</td><td>O</td><td>O</td><td>X</td></tr></table> pos.getRow = 0 pos.getCol = 4 p = 'X'							X					O	X				O	X	X			O	O	O	X	Output: checkDiagWin = true State of the board is unchanged	Reason: This case is unique because the player has placed the required number of tokens on a left diagonal and the player placed the final token at the bottom right of the diagonal Function Name: testCheckDiagWin_win_leftDiag_last_bottom
	X																										
	O	X																									
	O	X	X																								
	O	O	O	X																							

Input: State: (number to win = 4) <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td></tr><tr><td></td><td>O</td><td>X</td><td></td><td></td></tr><tr><td></td><td>O</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>O</td><td>O</td><td>O</td><td>X</td></tr></table> pos.getRow = 3 pos.getCol = 1 p = 'X'							X					O	X				O	X	X			O	O	O	X	Output: checkDiagWin = true State of the board is unchanged	Reason: This case is unique because the player has placed the required number of tokens on a left diagonal and the player placed the final token at the top left of the diagonal Function Name: testCheckDiagWin_win_leftDiag_last_top
	X																										
	O	X																									
	O	X	X																								
	O	O	O	X																							

<p>Input:</p> <p>State: (number to win = 4)</p> <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td></tr><tr><td></td><td>O</td><td>X</td><td></td><td></td></tr><tr><td></td><td>O</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>O</td><td>O</td><td>O</td><td>X</td></tr></table> <p>pos.getRow = 1 pos.getCol = 3 p = 'X'</p>							X					O	X				O	X	X			O	O	O	X	<p>Output:</p> <p>checkDiagWin = true</p> <p>State of the board is unchanged</p>	<p>Reason:</p> <p>This case is unique because the player has placed the required number of tokens on a left diagonal and the player placed the final token in the middle of the diagonal</p> <p>Function Name: testCheckDiagWin_win_leftDiag_last_mid</p>
	X																										
	O	X																									
	O	X	X																								
	O	O	O	X																							

<div><div><div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div>X</div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div>X</div></div><div><div>O</div></div></div> <div><div></div></div> <div><div></div></div> <div><div>X</div></div> <div><div>X</div></div> <div><div>O</div></div> <div><div></div></div> <div><div>X</div></div> <div><div>O</div></div> <div><div>O</div></div> <div><div>O</div></div> <div><div>pos.getRow = 0</div><div>pos.getCol = 1</div><div>p = 'X'</div></div>	<div><div><div>Output:</div><div>checkDiagWin = true</div><div>State of the board is unchanged</div></div></div>	<div><div><div>Reason:</div><div>This case is unique because the player has placed the required number of tokens on a right diagonal and the player placed the final token at the bottom left of the diagonal</div></div><div><div><div>Function Name:</div><div>testCheckDiagWin_win_right</div><div>Diag_last_bottom</div></div></div></div>
---	--	--

Input: State: (number to win = 4) <table border="1"><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td>X</td><td>O</td></tr><tr><td></td><td></td><td>X</td><td>X</td><td>O</td></tr><tr><td></td><td>X</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = 3 pos.getCol = 4 p = 'X'										X				X	O			X	X	O		X	O	O	O	Output: checkDiagWin = true State of the board is unchanged	Reason: This case is unique because the player has placed the required number of tokens on a right diagonal and the player placed the final token at the top right of the diagonal Function Name: testCheckDiagWin_win_right Diag_last_top
				X																							
			X	O																							
		X	X	O																							
	X	O	O	O																							

Input: State: (number to win = 4) <table border="1"><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td><td>X</td><td>O</td></tr><tr><td></td><td></td><td>X</td><td>X</td><td>O</td></tr><tr><td></td><td>X</td><td>O</td><td>O</td><td>O</td></tr></table> pos.getRow = 1 pos.getCol = 2 p = 'X'										X				X	O			X	X	O		X	O	O	O	Output: checkDiagWin = true State of the board is unchanged	Reason: This case is unique because the player has placed the required number of tokens on a right diagonal and the player placed the final token in the middle of the diagonal Function Name: testCheckDiagWin_win_right Diag_last_mid
				X																							
			X	O																							
		X	X	O																							
	X	O	O	O																							

Input: State: (number to win = 4) <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>X</td><td></td><td></td></tr><tr><td>O</td><td>X</td><td>O</td><td>X</td><td>O</td></tr><tr><td>X</td><td>O</td><td>O</td><td>X</td><td>X</td></tr><tr><td>O</td><td>X</td><td>O</td><td>O</td><td>X</td></tr></table> pos.getRow = 3 pos.getCol = 2 p = 'X'								X			O	X	O	X	O	X	O	O	X	X	O	X	O	O	X	Output: checkDiagWin = false State of the board is unchanged	Reason: This case is unique because the player has not placed the required number of tokens on either diagonal. There is one less token than necessary on both the left and right diagonals formed by the last placed token. Function Name: testCheckDiagWin_loss
		X																									
O	X	O	X	O																							
X	O	O	X	X																							
O	X	O	O	X																							

boolean checkTie()

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>																										Output: checkTie() = false State of the board is unchanged	Reason: This case is unique because the board is completely empty. It couldn't be any further from a tie Function Name: testCheckTie_emptyBoard

Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Output: checkTie() = true State of the board is unchanged	Reason: This case is unique because the board is completely full. This is the only scenario where a tie occurs Function Name: testCheckTie_fullBoard
X	X	X	X	X																							
X	X	X	X	X																							
X	X	X	X	X																							
X	X	X	X	X																							
X	X	X	X	X																							

Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Output: checkTie() = false State of the board is unchanged	Reason: This case is unique because the board has only 1 free space. It is 1 token away from a tie Function Name: testCheckTie_almostFullBoard
X	X	X	X																								
X	X	X	X	X																							
X	X	X	X	X																							
X	X	X	X	X																							
X	X	X	X	X																							

Input: State: <table><tr><td>X</td><td>X</td><td></td><td></td><td></td></tr><tr><td>X</td><td>X</td><td></td><td></td><td></td></tr><tr><td>X</td><td>X</td><td></td><td></td><td></td></tr><tr><td>X</td><td>X</td><td></td><td></td><td></td></tr><tr><td>X</td><td>X</td><td></td><td></td><td></td></tr></table>	X	X				X	X				X	X				X	X				X	X				Output: checkTie() = false State of the board is unchanged	Reason: This case is unique because the board is close to half full. A good many more tokens would be needed to cause a tie. Function Name: testCheckTie_halfFullBoard
X	X																										
X	X																										
X	X																										
X	X																										
X	X																										

char whatsAtPos(BoardPosition pos)

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>																										Output: For all positions in board, whatsAtPos = '' State of the board is unchanged	Reason: This case is unique because all positions in the board are blank spaces. Function Name: testWhatsAtPos_emptyBoard

<p>Input:</p> <p>State:</p> <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr></table> <p>pos.getRow = 0 pos.getColumn = 0</p>																					X					<p>Output:</p> <p>whatsAtPos = 'X'</p> <p>State of the board is unchanged</p>	<p>Reason:</p> <p>This case is unique because the X at the specified position is the only non-blank space in the board</p> <p>Function Name: testWhatsAtPos_nonEmptyBoard</p>
X																											

<p>Input:</p> <p>State:</p> <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr></table> <p>pos.getRow = 1 pos.getColumn = 0</p>																					X					<p>Output:</p> <p>whatsAtPos = ''</p> <p>State of the board is unchanged</p>	<p>Reason:</p> <p>This case is unique because the space being checked for a '' has a player token below it</p> <p>Function Name: testWhatsAtPos_blankSpaceAboveToken</p>
X																											

Input: State: <div> <div>X</div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div>X</div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div>X</div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div>X</div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div>X</div> <div></div> <div></div> <div></div> <div></div> </div>

pos.getColumn = 0

Input: State: <div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> </div> <div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> </div> <div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> </div> <div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> </div> <div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> </div>

boolean isPlayerAtPos(BoardPosition pos, char player)

Input: State: <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>
--

player = 'X'

<p>Input:</p> <p>State:</p> <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr></table> <p>pos.getRow = 0 pos.getColumn = 0 player = 'X'</p>																					X					<p>Output:</p> <p>isPlayerAtPos = true</p> <p>State of the board is unchanged</p>	<p>Reason:</p> <p>This case is unique because the X at the specified position is the only non-blank space in the board</p> <p>Function Name: testisPlayerAtPos_nonEmpty Board</p>
X																											

<div><div>Input:</div><div>State:</div><div><table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr></table></div><div><div>pos.getRow = 1</div><div>pos.getColumn = 0</div><div>player = 'X'</div></div></div> <div><div>Output:</div><div>isPlayerAtPos = false</div><div>State of the board is unchanged</div></div> <div><div>Reason:</div><div>This case is unique because it looks for the absence of a player token above a player token</div><div>Function Name: testIsPlayerAtPos_blankSpaceAboveToken</div></div>																					X				
X																									

Input: State: <table><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr></table> pos.getColumn = 0 player = 'X'	X					X					X					X					X					Output: For all positions in Column 0, isPlayeratPos = true State of the board is unchanged	Reason: This case is unique because the column whose positions are being checked is completely full of 'X' Function Name: testIsPlayerAtPos_fullCol
X																											
X																											
X																											
X																											
X																											

Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table> Player = 'X'	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Output: For all positions in the board, isPlayeratPos = false	Reason: This case is unique because all positions in the board are 'X's Function Name: testIsPlayerAtPos_fullBoard
X	X	X	X	X																							
X	X	X	X	X																							
X	X	X	X	X																							
X	X	X	X	X																							
X	X	X	X	X																							

void dropToken(char p, int c)

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> p = 'X' c = 0																										Output: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td></td><td></td><td></td><td></td></tr></table> 																					X					Reason: This case is unique because the token being placed is the first player token in the board Function Name: testDropToken_emptyBoard
X																																																				
Input: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table> p = 'X' c = 4	X	X	X	X		X	X	X	X		X	X	X	X		X	X	X	X		X	X	X	X		Output: State: <table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table> 	X	X	X	X		X	X	X	X		X	X	X	X		X	X	X	X		X	X	X	X	X	Reason: This case is unique because the token is being placed in the last available column Function Name: testDropToken_lastRemainin gCol
X	X	X	X																																																	
X	X	X	X																																																	
X	X	X	X																																																	
X	X	X	X																																																	
X	X	X	X																																																	
X	X	X	X																																																	
X	X	X	X																																																	
X	X	X	X																																																	
X	X	X	X																																																	
X	X	X	X	X																																																
Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>											Output: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>											Reason: This case is unique because the token is being placed on top of another token Function Name: testDropToken_colNotEmpty																														

X	O	X		

p = 'O'
c = 2

		O		
X	O	X		

Input: State: <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>O</td><td></td><td></td></tr><tr><td></td><td></td><td>X</td><td></td><td></td></tr><tr><td></td><td></td><td>O</td><td></td><td></td></tr><tr><td>X</td><td>O</td><td>X</td><td></td><td></td></tr></table> p = 'X' c = 2								O					X					O			X	O	X			Output: State: <table><tr><td></td><td></td><td>X</td><td></td><td></td></tr><tr><td></td><td></td><td>O</td><td></td><td></td></tr><tr><td></td><td></td><td>X</td><td></td><td></td></tr><tr><td></td><td></td><td>O</td><td></td><td></td></tr><tr><td>X</td><td>O</td><td>X</td><td></td><td></td></tr></table>			X					O					X					O			X	O	X			Reason: This case is unique because the token is being placed on top of another token to serve as the token that makes the column full Function Name: testDropToken_colAlmostFull
		O																																																		
		X																																																		
		O																																																		
X	O	X																																																		
		X																																																		
		O																																																		
		X																																																		
		O																																																		
X	O	X																																																		

Input:	Output:	Reason:																																																		
State:	State:	This case is unique because the token is being placed on top of another token to serve as the token that makes the column and the entire board full																																																		
<table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	<table><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Function Name: testDropToken_lastSpaceInBoard
X	X	X	X																																																	
X	X	X	X	X																																																
X	X	X	X	X																																																
X	X	X	X	X																																																
X	X	X	X	X																																																
X	X	X	X	X																																																
X	X	X	X	X																																																
X	X	X	X	X																																																
X	X	X	X	X																																																
X	X	X	X	X																																																
p = 'X' c = 4																																																				