

CCRPOG1 MP: Silly Tic-tac-toe Test Cases

GAME FUNCTIONS

void **gameTurn**(char cBoard[][SIZE], char cPlayer,int iPlayer1Abilities[], int iPlayer2Abilities[], int iNumOfAbilities, int iIsAbilityHidden);

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|-----------|---|---|--|---|---|---|---|---|--|---|--|--|--|---|---|---|---|---|---|---|--|---|--|---|------|
| Player X has used first ability | gameTurn(cBoard, 'X', {3,2}, {5,6}, 2, 0) | Use that ability (iSkipTurn is 1), update the board and pass to the next player (O) | Skips normal turn (iSkipTurn is 1), board updates and passes to the next player (O) | Pass | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O has placed on space 13 on normal turn | gameTurn(cBoard, 'O', {6}, {4}, 1, 0) *cBoard: <table border="1"> <tr><td>X</td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td></td><td>O</td><td>X</td><td>O</td></tr> <tr><td>O</td><td></td><td>O</td><td></td><td></td></tr> <tr><td></td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td>X</td><td>O</td><td>O</td><td></td><td>O</td></tr> </table> | X | X | X | O | X | | | O | X | O | O | | O | | | | X | X | O | X | X | O | O | | O | Player didn't use an ability (iSkipTurn is 0) so place symbol on space 13, update the board and pass the turn to next player (X) | Player didn't use an ability (iSkipTurn is 0) and places symbol on space 13, board updates and passes the turn to next player (X) | Pass |
| X | X | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | O | | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player X has used second ability | gameTurn(cBoard, 'X', {2,1,6,3}, {1,2,3,4}, 4, 1) *1 is "AnotherTwo" | Player will have 2 turns and will continue to loop until there is no more turns. | Player has 2 turns and will continue to loop until there is no more turns. | Pass | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O has placed on space 25 while having 1 more turn. | gameTurn(cBoard, 'O', {0,6}, {0,0}, 2, 0) *cBoard: <table border="1"> <tr><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td>O</td><td>X</td><td>X</td><td>O</td></tr> <tr><td>O</td><td></td><td>O</td><td></td><td></td></tr> <tr><td></td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td>O</td><td>X</td><td></td><td>O</td></tr> </table> | O | X | X | O | X | | O | X | X | O | O | | O | | | | X | X | O | X | | O | X | | O | Place symbol on space 25, update the board, end the turn by stopping the loop (iTurns is 0) and call gameLogic function. | Places symbol on space 25, updates the board, ends the turn by stopping the loop (iTurns is 0) and calls gameLogic function. | Pass |
| O | X | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| | O | X | X | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| | O | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player X, on normal turn, inputted 26 | Where do you want to place? [1-25]: 26 | Show error "Enter number from 1 to 25 only." | Shows error "Enter number from 1 to 25 only." | Pass | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O, on normal turn, inputted 1 but that space is already occupied by Player X | Where do you want to place? [1-25]: 1 | Show error "Place is already occupied, enter other number from 1 to 25." | Shows error "Place is already occupied, enter other number from 1 to 25." | Pass | | | | | | | | | | | | | | | | | | | | | | | | | |

void **gameLogic**(char cBoard[][SIZE], char cPlayer, char cPreviousPlayer, int iPlayer1Abilities[], int iPlayer2Abilities[], int iNumOfAbilities, int iIsAbilityHidden)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|---|---------------------------|-------------------------|-----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|------|
| Player X placed on space 9 | gameLogic(cBoard, 'O', 'X', {3,2}, {5,6}, 2, 0) *cBoard: <table border="1"> <tr><td>X</td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td></td><td></td><td>X</td><td></td></tr> <tr><td></td><td></td><td>X</td><td>O</td><td></td></tr> <tr><td></td><td>X</td><td></td><td>O</td><td></td></tr> <tr><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td></tr> </table> | X | X | X | O | X | | | | X | | | | X | O | | | X | | O | | X | O | O | O | O | Game stops (iGameStop is 1) and call function gameWinner(X) | Game stops (iGameStop is 1) and calls function gameWinner(X) | Pass |
| X | X | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | O | O | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O placed on space 17 | gameLogic(cBoard, 'X', 'O', {2}, {1}, 1, 0) *cBoard: <table border="1"> <tr><td>X</td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td>X</td><td>X</td><td>O</td><td>X</td><td>O</td></tr> <tr><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td>X</td></tr> <tr><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td></tr> </table> | X | X | X | O | X | X | X | O | X | O | O | O | X | O | O | X | O | X | O | X | X | O | O | O | O | Game stops (iGameStop is 1) and call function gameDraw | Game stops (iGameStop is 1) and call function gameDraw | Pass |
| X | X | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | X | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | O | X | O | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | O | O | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O placed on space 25 | gameLogic(cBoard, 'X', 'O', {1,2,3,4}, {3,2,1,5}, 4, 0) *cBoard: <table border="1"> <tr><td>X</td><td>O</td><td></td><td>X</td><td></td></tr> <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></td><td></td><td></td><td>X</td><td></td></tr> <tr><td>O</td><td></td><td>O</td><td></td><td>O</td></tr> </table> | X | O | | X | | | | | | | | | X | | | | | | X | | O | | O | | O | Game continues (iGameStop is 0) and pass to the next player (X) | Game continues (iGameStop is 0) and passes to the next player (X) | Pass |
| X | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | O | | O | | | | | | | | | | | | | | | | | | | | | | | | | |

int **gameCheck**(char cBoard[][SIZE], char cPlayer)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------------------|--|---------------------------|-------------------------|-----------|---|---|--|--|--|---|--|--|--|---|---|--|--|---|--|---|--|---|---|---|---|---|-----------------------------|------------------------------|------|
| Player X placed on space 9 | gameCheck(cBoard, 'X') *cBoard: <table border="1"> <tr><td>X</td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td></td><td></td><td>X</td><td></td></tr> <tr><td></td><td></td><td>X</td><td>O</td><td></td></tr> <tr><td></td><td>X</td><td></td><td>O</td><td></td></tr> <tr><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td></tr> </table> | X | X | X | O | X | | | | X | | | | X | O | | | X | | O | | X | O | O | O | O | Return 1 (Player winner) | Returns 1 (Player winner) | Pass |
| X | X | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | O | O | O | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------------|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---------------------------|---------------------------|------|
| Player O placed on space 17 | gameCheck(cBoard, 'O') <i>*cBoard:</i> <table border="1"> <tr><td>X</td><td>X</td><td>X</td><td>O</td><td>X</td></tr> <tr><td>X</td><td>X</td><td>O</td><td>X</td><td>O</td></tr> <tr><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td>X</td></tr> <tr><td>X</td><td>O</td><td>O</td><td>O</td><td>O</td></tr> </table> | X | X | X | O | X | X | X | O | X | O | O | O | X | O | O | X | O | X | O | X | X | O | O | O | O | Return 2 (Game draw) | Returns 2 (Game draw) | Pass |
| X | X | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | X | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | O | X | O | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | O | O | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O placed on space 25 | gameCheck(cBoard, 'O') <i>*cBoard:</i> <table border="1"> <tr><td>X</td><td>O</td><td></td><td>X</td><td></td></tr> <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></td><td></td><td></td><td>X</td><td></td></tr> <tr><td>O</td><td></td><td>O</td><td></td><td>O</td></tr> </table> | X | O | | X | | | | | | | | | X | | | | | | X | | O | | O | | O | Return 0 (Game continues) | Return 0 (Game continues) | Pass |
| X | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | O | | O | | | | | | | | | | | | | | | | | | | | | | | | | |

void **gameStart()**

| Description | Input value/parameters | Expected output/result | Actual output/result | Pass/Fail |
|--|--|--|--|-----------|
| Players want 5 abilities and hidden abilities' names | How many abilities do you want? [0-8]: 5 Would you like the abilities' names to be hidden during the game? [y/n]: y | Proceed to call gameChooseAbility with the following parameters: 1. Empty cBoard 2. 5 – no. of abilities 3. 1 – hidden true | Proceed to call gameChooseAbility with the following parameters: 1. Empty cBoard 2. 5 – no. of abilities 3. 1 – hidden true | Pass |
| Players want 0 abilities | How many abilities do you want? [0-8]: 0 | Proceed to start the game (shows the board and Player 1's turn) | Proceeds to start the game (shows the board and Player 1's turn) | Pass |

void **gameChooseAbility**(char cBoard[][SIZE], int iNumOfAbilities, int iIsAbilityHidden);

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---|---------------------------------------|---|--|-----------|
| Player X selected the first ability again | Select second ability [1-8]: 1 | Show error "Please select another unique ability." | Shows error "Please select another unique ability." | Pass |
| Player O inputted 9 | Select third ability [1-8]: 9 | Show error "Please select from 1 to 8 only." | Shows error "Please select from 1 to 8 only." | Pass |
| Player X inputted a unique ability | Select first ability [1-8]: 2 | No errors. Store the ability in the iPlayer1Abilities array and proceed to ask player for the next ability. | No errors. Stores the ability in the iPlayer1Abilities array and proceed to ask player for the next ability. | Pass |

int **gameUseAbility**(char cBoard[][SIZE], char cPlayer, int *iTurns, int iAbilities[], int iOpponentAbilities[], int iNumOfAbilities, int iIsAbilityHidden, char strGameStatus[], int iAbility1);

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---|----------------------------|---|---|-----------|
| Player X inputted 0 | Use an ability? 0 | Use normal turn (return 0) | Use normal turn (return 0) | Pass |
| Player O inputted 4 *iNumOfAbilities is 3 [AVAILABLE ABILITIES] 1-AnotherTwo 2-Counter 3-ReSymbol | Use an ability? 4 | Show error "Enter number from the choices only." | Shows error "Enter number from the choices only." | Pass |
| Player X inputted 2 *iNumOfAbilities is 3 [AVAILABLE ABILITIES] 1-AnotherTwo 2-Counter 3-Aequalis | Use an ability? 2 | Use normal turn (return 0) | Use normal turn (return 0) | Pass |
| Player O wants to use an ability but Player X has countered the ability | Use an ability? 1 | Assign that ability to 0 and proceed to use normal turn | Assigns that ability to 0 and proceeds to use normal turn | Pass |

| | | | | |
|---|-------------------|--|--|------|
| Player X inputted 2 <i>*iNumOfAbilities is 3</i> [AVAILABLE ABILITIES] 1-AnotherTwo 3-Aequalis | Use an ability? 2 | Show error "Ability is used/removed. Select another ability or use normal turn." | Shows error "Ability is used/removed. Select another ability or use normal turn." | Pass |
| Player O inputted 1 <i>*iNumOfAbilities is 2</i> [AVAILABLE ABILITIES] 1-ReSymbol 2-Aequalis | Use an ability? 1 | Proceed to use the ability, assign 0 to that ability (used) and return 1 (skip turn) | Proceed to use the ability, assign 0 to that ability (used) and return 1 (skip turn) | Pass |

void **gameUpdate**(char cBoard[][SIZE], char cPlayer)

| Description | Input value/parameters | Expected output/result | Actual output/result | Pass/Fail | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|--|----------------------|-----------|--|---|--|--|--|--|--|--|--|--|---|--|--|--|--|--|---|--|---|--|---|--|---|--|---|---|--|---|--|--|--|--|--|--|--|--|---|--|--|--|--|--|---|--|---|--|---|--|---|------|
| Update screen after Player O placed on space 25 | gameUpdate({ {'X', 'O', '', 'X', ''}, {'', '', '', '', ''}, {'', '', 'X', '', ''}, {'', '', '', 'X', ''}, {'O', '', 'O', '', 'O'} }, X); | Clear the screen, show the game title, the board and Player 1 (X) turn <table border="1"> <tr><td>X</td><td>O</td><td></td><td>X</td><td></td></tr> <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></td><td></td><td></td><td>X</td><td></td></tr> <tr><td>O</td><td></td><td>O</td><td></td><td>O</td></tr> </table> | X | O | | X | | | | | | | | | X | | | | | | X | | O | | O | | O | Clears the screen, shows the game title, the board and Player 1 (X) turn <table border="1"> <tr><td>X</td><td>O</td><td></td><td>X</td><td></td></tr> <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></td><td></td><td></td><td>X</td><td></td></tr> <tr><td>O</td><td></td><td>O</td><td></td><td>O</td></tr> </table> | X | O | | X | | | | | | | | | X | | | | | | X | | O | | O | | O | Pass |
| X | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | O | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | O | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

void **gameUpdateWithStatus**(char cBoard[][SIZE], char cPlayer , char strGameStatus[])

| Description | Input value/parameters | Expected output/result | Actual output/result | Pass/Fail | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|----------------------|-----------|--|--|--|--|---|--|---|--|--|--|--|--|---|--|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---|--|---|--|--|--|--|--|---|--|---|--|--|--|--|--|--|--|--|------|
| Update screen after Player X used "Blank four corners and center" or "DISAPPEAR" | gameUpdate({ {'', '', '', '', ''}, {'', 'O', '', 'X', ''}, {'', '', '', '', 'X'}, {'', 'X', '', '', ''}, {'', '', '', '', ''} }, O, strGameStatus); <i>*strGameStatus:</i> "1Cleared all corners and middle space." | Clear the screen, show the game title, the board, game message and Player 2 (O) turn <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>O</td><td></td><td>X</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>X</td></tr> <tr><td></td><td>X</td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table> "[PLAYER 1] Cleared all corners and middle space." | | | | | | | O | | X | | | | | | X | | X | | | | | | | | | Clears the screen, shows the game title, the board, game message and Player 2 (O) turn <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>O</td><td></td><td>X</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>X</td></tr> <tr><td></td><td>X</td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table> "[PLAYER 1] Cleared all corners and middle space." | | | | | | | O | | X | | | | | | X | | X | | | | | | | | | Pass |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

void **gameWinner**(char cPlayer)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---------------------------|----------------------------|--------------------------------------|---------------------------------------|-----------|
| Player X is the winner | gameWinner('X') | Show "PLAYER 1 IS THE WINNER!" | Shows "PLAYER 1 IS THE WINNER!" | Pass |
| Player O is the winner | gameWinner('O') | Show "PLAYER 2 IS THE WINNER!" | Shows "PLAYER 2 IS THE WINNER!" | Pass |

char **getFromBoard**(char cBoard[][SIZE], int iPlace)

*cBoard:

| | | | | |
|---|---|---|---|---|
| X | X | X | O | X |
| | | O | X | O |
| O | | O | | |
| | X | X | O | X |
| X | O | O | | O |

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|--------------------------------|-----------------------------|---------------------------|-------------------------|-----------|
| Get character from space 13 | getFromBoard(cBoard, 13) | Return 'O' | Return 'O' | Pass |
| Get character from space 1 | getFromBoard(cBoard, 1) | Return 'X' | Return 'X' | Pass |
| Get character from space 7 | getFromBoard(cBoard, 7) | Return '' | Return '' | Pass |

(Inputs for getHorizontalRow, getVerticalRow, and getDiagonalRow)

**iRequirement is 2 – used on Free Row*

**cBoard:*

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | A | B | C | D | E | 2 |
| A | | X | | O | X | |
| B | | | | | | |
| C | O | | | | | |
| D | | | X | O | | |
| E | X | | | | O | |

int **getHorizontalRow**(char cBoard[][SIZE], char cPlayer, int iRequirement)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---|----------------------------------|--|----------------------|-----------|
| Get Player X occupied spaces on each horizontal row | getHorizontalRow(cBoard, 'X', 2) | Return 2 (there are 2 occupied spaces on row A) | Return 2 | Pass |
| Get Player O occupied spaces on each horizontal row | getHorizontalRow(cBoard, 'O', 2) | Return 0 | Return 0 | Pass |

int **getVerticalRow**(char cBoard[][SIZE], char cPlayer, int iRequirement)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---|--------------------------------|---|----------------------|-----------|
| Get Player X occupied spaces on each vertical row | getVerticalRow(cBoard, 'X', 2) | Return 0 | Return 0 | Pass |
| Get Player O occupied spaces on each vertical row | getVerticalRow(cBoard, 'O', 2) | Return 2 (there are 2 occupied spaces on column D) | Return 2 | Pass |

int **getDiagonalRow**(char cBoard[][SIZE], char cPlayer, int iRequirement)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---|--------------------------------|---|----------------------|-----------|
| Get Player X occupied spaces on each diagonal row | getVerticalRow(cBoard, 'X', 2) | Return 2 (there are 2 occupied spaces on diagonal 2) | Return 2 | Pass |

| | | | | |
|---|--------------------------------|---|----------|------|
| Get Player O occupied spaces on each diagonal row | getVerticalRow(cBoard, 'O', 2) | Return 2 (there are 2 occupied spaces on diagonal 1) | Return 2 | Pass |
|---|--------------------------------|---|----------|------|

void **placeToBoard**(char cBoard[][SIZE], int iPlace, char cInput)

| Description | Input value/parameters | Expected output/result | Actual output/result | Pass/Fail |
|---------------------------------|--|---|---|-----------|
| Player X has placed on input 13 | gameUpdate({ {'X', 'O', '', 'X', ''}, {'', '', '', '', ''}, {'', '', '', '', ''}, {'', '', 'X', ''}, {'O', '', 'O', '', ''} }, X); | cBoard = { {'X', 'O', '', 'X', ''}, {'', '', '', '', ''}, {'', '', 'X', ''}, {'', 'X', '', ''}, {'', '', 'X', ''}, {'O', '', 'O', '', ''} } | cBoard = { {'X', 'O', '', 'X', ''}, {'', '', '', '', ''}, {'', '', 'X', ''}, {'', 'X', '', ''}, {'', '', 'X', ''}, {'O', '', 'O', '', ''} } | Pass |

void **showGameStatus**(char cPlayer, char strGameStatus[])

| Description | Input value/parameters | Expected output/result | Actual output/result | Pass/Fail |
|--|--|--|---|-----------|
| Player X used "Free Row" ability but doesn't have 2 spaces occupied in a row | showGameStatus('O', strGameStatus) *strGameStatus: "1You don't have a row with 2 spaces occupied, hence Free Row will have no effect." | Show "[PLAYER 1] You don't have a row with 2 spaces occupied, hence Free Row will have no effect." | Shows "[PLAYER 1] You don't have a row with 2 spaces occupied, hence Free Row will have no effect." | Pass |

void **showAbilityUsed**(int iAbility)

| Description | Input value/parameters | Expected output/result | Actual output/result | Pass/Fail |
|----------------------------------|------------------------|--------------------------------------|---------------------------------------|-----------|
| Player used "AnotherTwo" ability | showAbilityUsed(0) | Show "[ABILITY USED: AnotherTwo]" | Shows "[ABILITY USED: AnotherTwo]" | Pass |

void **showPlayerTurn**(char cPlayer)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|-----------------|----------------------------|----------------------------------|-----------------------------------|-----------|
| Player 1's turn | showPlayerTurn('X') | Show "Player 1's turn (X)" | Shows "Player 1's turn (X)" | Pass |
| Player 2's turn | showPlayerTurn('O') | Show "Player 2's turn (O)" | Shows "Player 2's turn (O)" | Pass |

ABILITIES FUNCTIONS

**strGameStatus is empty for ability 2 to 7*

void **ability1**(int **iTurns)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|-----------------------------|---|--|---|-----------|
| Player used "AnotherTwo" | ability1(iTurns) <i>*iTurns is the address of the original iTurns variable</i> | Add 2 turns for the player and show "[ABILITY USED: AnotherTwo] Added 2 turns for the player" | Adds 2 turns for the player and shows "[ABILITY USED: AnotherTwo] Added 2 turns for the player" | Pass |

void **ability2**(char cBoard[][SIZE], char cPlayer, char strGameStatus[], int iAbility1)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---|---|---|---|-----------|
| Player X used "ReSymbol" but Player O doesn't have a symbol on the board | ability2(cBoard, 'X', strGameStatus, 0) | Copy string "1There is no opponent's symbol to replace, hence ReSymbol has no effect." | Copies string "1There is no opponent's symbol to replace, hence ReSymbol has no effect." | Pass |
| Player O inputted 0 | Choose a place to replace opponent's symbol [1-25]: 0 | Show error "Enter number from 1 to 25 only." | Shows error "Enter number from 1 to 25 only." | Pass |
| Player X inputted 2 but that space is empty | Choose a place to replace opponent's symbol [1-25]: 2 | Show error "Place is not occupied by opponent, enter other number from 1 to 25." | Shows error "Place is not occupied by opponent, enter other number from 1 to 25." | Pass |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|--|---|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|---|--|---|---|---|------|
| Player O inputted 1 and that space is occupied by opponent | Choose a place to replace opponent's symbol [1-25]: 1 <i>*iAbility1 is 1</i> <i>*cBoard:</i> <table border="1"> <tr><td>X</td><td></td><td>X</td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td>O</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></td><td>X</td><td></td></tr> <tr><td></td><td></td><td>O</td><td></td><td>X</td></tr> </table> | X | | X | | O | | X | O | | | | | O | | | | | | X | | | | O | | X | Replace symbol and show “[ABILITY USED: ReSymbol] Replaced opponent’s symbol” | Replaces symbol and shows “[ABILITY USED: ReSymbol] Replaced opponent’s symbol” | Pass |
| X | | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | |

void **ability3**(char cBoard[][SIZE], char cPlayer, char strGameStatus[], int iAbility1)

| Description | Input value/parameters | Expected output/result | Actual output/result | Pass/Fail | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------------------|--|------------------------|----------------------|-----------|--|---|--|---|---|---|--|---|---|---|---|---|---|---|---|---|---|---|---|--|--|---|---|--|--|---|--|--|--|---|---|---|--|---|---|--|---|---|---|---|---|---|---|--|---|--|--|--|--|--|--|---|--|--|--|---|---|---|--|---|---|--|---|---|---|---|---|---|---|--|---|--|--|--|------|
| Player O used “DISAPPEAR” | Ability3(cBoard, O, strGameStatus, 0) <i>*cBoard:</i> <table border="1"> <tr><td>X</td><td></td><td>X</td><td></td><td>X</td></tr> <tr><td></td><td>X</td><td>O</td><td>X</td><td></td></tr> <tr><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td>X</td></tr> <tr><td>X</td><td>O</td><td></td><td></td><td>O</td></tr> </table> | X | | X | | X | | X | O | X | | O | O | X | O | O | X | O | X | O | X | X | O | | | O | Replace corners and the middle space with blank <table border="1"> <tr><td></td><td></td><td>X</td><td></td><td></td></tr> <tr><td></td><td>X</td><td>O</td><td>X</td><td></td></tr> <tr><td>O</td><td>O</td><td></td><td>O</td><td>O</td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td>O</td><td></td><td></td><td></td></tr> </table> and copy string “1Cleared all corners and middle space.” | | | X | | | | X | O | X | | O | O | | O | O | X | O | X | O | X | | O | | | | Replaces corners and the middle space with blank <table border="1"> <tr><td></td><td></td><td>X</td><td></td><td></td></tr> <tr><td></td><td>X</td><td>O</td><td>X</td><td></td></tr> <tr><td>O</td><td>O</td><td></td><td>O</td><td>O</td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td>O</td><td></td><td></td><td></td></tr> </table> and copies string “1Cleared all corners and middle space.” | | | X | | | | X | O | X | | O | O | | O | O | X | O | X | O | X | | O | | | | Pass |
| X | | X | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | O | X | O | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | O | | O | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | O | | O | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

void **ability4**(char cBoard[][SIZE], char cPlayer, char strGameStatus[], int iAbility1)

| Description | Input value/parameters | Expected output/result | Actual output/result | Pass/Fail |
|--|---|--|--|-----------|
| Player X used “Free Row” but doesn’t have 2 spaces occupied on any type of row | ability4(cBoard, ‘X’, strGameStatus, 0) | Copy string “1You don't have a row with 2 spaces occupied, hence Free Row will have no effect.” | Copies string “1You don't have a row with 2 spaces occupied, hence Free Row will have no effect.” | Pass |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|--|------|--|---|--|---|---|---|--|---|--|---|---|---|---|---|---|---|---|---|--|---|--|--|---|---|------|---|--|---|--|---|--|---|--|--|--|--|--|--|---|---|---|---|--|---|--|---|--|--|---|--|--|---|--|---|--|---|--|---|--|--|--|--|--|--|---|---|---|---|--|---|--|---|--|--|------|
| Player O inputted 0 when selecting type of row | Select type of row [1-horizontal]: 0 | Show error "Enter number from choices only." | Shows error "Enter number from choices only." | Pass | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player X inputted 2 when selecting diagonal row but that row has no spaces occupied by the player | Select row to clear opponent's symbols. [1-2]: 2 *cBoard: <table border="1"> <tr><td>X</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>X</td><td>O</td><td></td><td></td></tr> <tr><td>O</td><td></td><td></td><td>O</td><td>O</td></tr> <tr><td>X</td><td></td><td>X</td><td>O</td><td>X</td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table> | X | | | | | | X | O | | | O | | | O | O | X | | X | O | X | | | | | | Show error "Enter a row number with 2 spaces you occupy." | Shows error "Enter a row number with 2 spaces you occupy." | Pass | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | | O | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O inputted 6 when selecting horizontal row | Select row to clear opponent's symbols. [1-5]: 6 | Show error "Enter number from 1 to 5 only." | Shows error "Enter number from 1 to 5 only." | Pass | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player X inputted 2 when selecting vertical row and that row has 3 occupied spaces by the player | Select row to clear opponent's symbols. [1-5]: 3 *iAbility1 is 1 *cBoard: <table border="1"> <tr><td></td><td></td><td>X</td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td>O</td><td>X</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>O</td><td></td></tr> <tr><td>O</td><td></td><td>X</td><td></td><td></td></tr> </table> | | | X | | O | | X | O | X | | | | O | | | X | O | X | O | | O | | X | | | Replace the 3 rd column with blank <table border="1"> <tr><td></td><td></td><td>X</td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td></td><td>X</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td></td></tr> <tr><td>O</td><td></td><td>X</td><td></td><td></td></tr> </table> and show "[ABILITY USED: Free Row] Cleared opponent's symbol" | | | X | | O | | X | | X | | | | | | | X | O | X | O | | O | | X | | | Replaces the 3 rd column with blank <table border="1"> <tr><td></td><td></td><td>X</td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td></td><td>X</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td></td></tr> <tr><td>O</td><td></td><td>X</td><td></td><td></td></tr> </table> and shows "[ABILITY USED: Free Row] Cleared opponent's symbol" | | | X | | O | | X | | X | | | | | | | X | O | X | O | | O | | X | | | Pass |
| | | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

void **ability5**(char cBoard[][SIZE], char cPlayer, char strGameStatus[], int iAbility1)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---------------------------------|--|---|--|-----------|
| Opponent (Player O) inputted 26 | Where does PLAYER 2 (O) want to place? [1-25]: 26 | Show error "Enter number from 1 to 25 only." | Shows error "Enter number from 1 to 25 only." | Pass |
| Opponent (Player X) inputted a | Where does PLAYER 1 (X) want to place? [1-25]: 5 | Show error "Place is already occupied, enter | Shows error "Place is already occupied, enter | Pass |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|-----------------------------|-----------------------------|---|--|---|--|---|---|---|--|--|--|---|--|--|--|--|---|--|--|---|---|---|---|--|---|--|---|--|--|--|--|--|---|--|---|--|---|---|---|--|--|--|---|---|--|---|---|---|---|--|---|--|---|--|--|---|--|--|---|--|---|--|---|---|---|--|--|--|---|---|--|---|---|---|---|--|---|--|---|--|--|------|
| space that is occupied | | other number from 1 to 25." | other number from 1 to 25." | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opponent (Player O) inputted 14 and that space is blank | Where does PLAYER 2 (O) want to place? [1-25]: 14 <i>*cBoard:</i> <table border="1"> <tr><td></td><td></td><td>X</td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td>O</td><td>X</td><td></td></tr> <tr><td></td><td></td><td>O</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>O</td><td></td></tr> <tr><td>O</td><td></td><td>X</td><td></td><td></td></tr> </table> | | | X | | O | | X | O | X | | | | O | | | | | O | | | X | O | X | O | | O | | X | | | Place Player X's symbol on space 14 <table border="1"> <tr><td></td><td></td><td>X</td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td>O</td><td>X</td><td></td></tr> <tr><td></td><td></td><td>O</td><td>X</td><td></td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td></td></tr> <tr><td>O</td><td></td><td>X</td><td></td><td></td></tr> </table> | | | X | | O | | X | O | X | | | | O | X | | X | O | X | O | | O | | X | | | Places Player X's symbol on space 14 <table border="1"> <tr><td></td><td></td><td>X</td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td>O</td><td>X</td><td></td></tr> <tr><td></td><td></td><td>O</td><td>X</td><td></td></tr> <tr><td>X</td><td>O</td><td>X</td><td>O</td><td></td></tr> <tr><td>O</td><td></td><td>X</td><td></td><td></td></tr> </table> | | | X | | O | | X | O | X | | | | O | X | | X | O | X | O | | O | | X | | | Pass |
| | | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | O | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

void **ability6**(char cBoard[][SIZE], char cPlayer, char strGameStatus[], int iAbility1)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|-----------|--|---|--|---|---|--|--|--|--|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|---|--|---|--|---|--|--|--|---|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|---|--|---|---|---|--|--|--|---|--|---|---|--|--|--|--|---|--|--|--|--|--|---|--|--|--|---|--|---|------|
| Player X inputted 26 | Where do you want to remove? [1-25]: 26 | Show error "Enter number from 1 to 25 only." | Shows error "Enter number from 1 to 25 only." | Pass | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O inputted 5 but that space is blank | Where do you want to remove? [1-25]: 5 | Show error "Enter a number that the opponent occupies." | Shows error "Enter a number that the opponent occupies." | Pass | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player X used "Aequalis" but the opponent has less occupied spaces | ability6(cBoard, 'X', strGameStatus, 0) | Copy string "1The opponent doesn't have more spaces occupied, hence Aequalis will have no effect." | Copies string "1The opponent doesn't have more spaces occupied, hence Aequalis will have no effect." | Pass | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Player O inputted 3 and that space is occupied by opponent | Where do you want to remove? [1-25]: 3 <i>*cBoard:</i> <table border="1"> <tr><td>X</td><td></td><td>X</td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td>O</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>O</td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td>X</td><td></td></tr> <tr><td></td><td></td><td>O</td><td></td><td>X</td></tr> </table> | X | | X | | O | | X | O | | | | | O | | | | | O | | | | | | X | | | | O | | X | Replace space 3 with blank <table border="1"> <tr><td>X</td><td></td><td></td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td>O</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></td><td>X</td><td></td></tr> <tr><td></td><td></td><td>O</td><td></td><td>X</td></tr> </table> | X | | | | O | | X | O | | | | | O | | | | | | X | | | | O | | X | Replaces space 3 with blank <table border="1"> <tr><td>X</td><td></td><td></td><td></td><td>O</td></tr> <tr><td></td><td>X</td><td>O</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></td><td>X</td><td></td></tr> <tr><td></td><td></td><td>O</td><td></td><td>X</td></tr> </table> | X | | | | O | | X | O | | | | | O | | | | | | X | | | | O | | X | Pass |
| X | | X | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | | | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| X | | | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | X | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | O | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

void **ability7**(int iOpponentAbilities[], int iNumOfAbilities, int iIsAbilityHidden, char strGameStatus[], int iAbility1)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|---|--|--|---|-----------|
| Player X inputted 3 but the number of abilities is 2 only | Enter which ability to remove from opponent [2-Free Row;]: 3 <i>*iNumOfAbilities is 2</i> | Show error "Enter number from the choices only." | Shows error "Enter number from the choices only." | Pass |
| Player O inputted 3 but that opponent's ability is used | Enter which ability to remove from opponent [1-Vanish; 2-Free Row;]: 3 <i>*iNumOfAbilities is 3</i> | Show error "Ability doesn't exist, select another ability." | Shows error "Ability doesn't exist, select another ability." | Pass |
| Player X used "Vanish" but the opponent doesn't have any abilities left | ability7({0,0}, 2, 0, strGameStatus, 0) | Copy string "1The opponent has no abilities left, hence Vanish will have no effect." | Copies string "1The opponent has no abilities left, hence Vanish will have no effect." | Pass |
| Player X inputted 1 when choosing to remove an ability | Enter which ability to remove from opponent [1-Counter;]: 1 <i>*iAbility1 is 0</i> | Assign that opponent's ability to 0 and copy string "1Removed an ability from opponent." | Assigns that opponent's ability to 0 and copies string "1Removed an ability from opponent." | Pass |

int **ability8**(char cPlayer, int* iAbility)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|--|--|---|--|-----------|
| Opponent (Player X) inputted a different character | Would PLAYER 1 like to counter the ability? [y/n]: m | Show error “Please choose Y or N only.” | Shows error “Please choose Y or N only.” | Pass |
| Opponent (Player O) doesn’t want to counter ability | Would PLAYER 2 like to counter the ability? [y/n]: n | Return 0 | Return 0 | Pass |
| Opponent (Player X) doesn’t want to counter ability | Would PLAYER 1 like to counter the ability? [y/n]: y | Assign that opponent’s counter ability to 0, show “ PLAYER 1 has countered PLAYER 2’s ability ”, and return 1 | Assigns that opponent’s counter ability to 0, shows “ PLAYER 1 has countered PLAYER 2’s ability ”, and return 1 | Pass |

OTHER FUNCTIONS

void **getOpponent**(char cPlayer, char *cOpponent)

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|-----------------------------|----------------------------|----------------------------|-----------------------------|-----------|
| Get opponent of Player X | getOpponent('X', ' ') | Assign 'O' to cOpponent | Assigns 'O' to cOpponent | Pass |
| Get opponent of Player O | getOpponent('O', ' ') | Assign 'X' to cOpponent | Assigns 'X' to cOpponent | Pass |

void **getAbilityName**(int iLoop, char strAbilityName[])

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|--------------------------------|---|---------------------------|-----------------------------|-----------|
| Get ability name of input 1 | getAbilityName(1, strAbilityName) <i>*strAbilityName is empty</i> | Copy string “ReSymbol” | Copies string “ReSymbol” | Pass |

int **promptUser()**

| Description | Input value/ parameters | Expected output/result | Actual output/result | Pass/Fail |
|-----------------|----------------------------|--|--|-----------|
| User inputted 1 | Choose [1 or 2]: 1 | Return 1 | Return 1 | Pass |
| User inputted 2 | Choose [1 or 2]: 2 | Return 2 | Return 2 | Pass |
| User inputted 3 | Choose [1 or 2]: 3 | Show error "Please choose properly." | Show error "Please choose properly." | Pass |