

For this assignment, you will create a tic-tac-toe game by implementing the following functions:

```
initialize()  
play_X(g, x, y)  
play_O(g, x, y)  
win(x,y)  
draw_board(g)  
play_a_game()
```

The initialize function should create a new empty game board and return a pointer to that data structure. play_X and play_O place their respective symbols in the specified cell, then call the win function to determine if that move resulted in a win. Those functions return one of three values, representing a valid move, an invalid move, and a move resulting in a win. The win function uses the knowledge of the most recent move to only check for ways in which a win could have occurred this turn instead of having to test all possible solutions. It returns values representing either a win or no win. The draw_board function prints the current state of the game to the console. play_a_game uses the previous functions to allow two people to play tic-tac-toe.

You should not need to modify the test suite. You only need to write the functions. Since QT SPIM expects all of your code to be in a single file, you can concatenate them together in a few ways. If you are on Windows, you can use the included batch file to do the work for you. Simply dragging your source file and dropping it on the batch file should be sufficient. If you are having trouble with the batch file, make sure that your file names match those below. You can also use a command line operation.

Windows: copy /Y "<Your Source File Name>"+ "Test Suite.asm" Output.asm

Unix: cat "<Your Source File Name>" "Test Suite.asm" > Output.asm

Your program should include appropriate comments indicating what the code should be doing and what registers are being used for. After displaying the results, your program should exit cleanly. You should test your programs using the SPIM simulator to ensure their functionality before submitting them. You should only submit your functions. You will not receive credit if you submit the test suite in any form. You should also not include any driver or debug code in your submission.

Objectives:

1. To introduce two-dimensional arrays.
2. To understand array indexing in multi-dimensional arrays.
3. To review dynamic memory allocation.
4. To review parameter checking.