1) (10 pts) DSN (Recursive Coding)

Write a recursive function that determines if player X, who goes first, can win a game of tic-tac-toe (played on a 3 by 3 int board). You can use the following helper functions. You don't have to implement any of the listed helper functions. **Functions that do not use recursion will receive 0 points**. Note: board is a 3 by 3 array of integers, storing either EMPTY(0), X(1), or O(2). The three given functions return the current state of the board, as it is, from player X's perspective, not what "could happen" in the future. myTurn is 1, when it's X's turn and 0 when it's O's turn. You will have to place and unplace X's and O's in your solution. Finally, the intention here is that player 'O' plays pessimistic. Namely, if there is ANY set of moves that players X and O could make from the current state of the board that result in X winning, the function should return 1.

```
#define EMPTY 0
#define X 1
#define O 2

int XWin(int ** board); // returns 1 if I have won and 0 otherwise
int XLose(int ** board); // returns 1 if I have lost and 0 otherwise
int tied(int ** board); // returns 1 if the board is in a tied state

int canXWin(int ** board, int myTurn) {

}
```