BF550: Fall 2020
Project 0 is due by 12 pm on Monday, November 2

Part 1
Write a program that plays a game of m, n, k-Tic-tac-toe with a user. At the beginning the user chooses the values of m, n, k, and the type of AI that they will play against. Each round, the game prints our the state of the board, asks the user where they would like to place their mark, and implements this decision. Once one of the player won, the program declares the result and asks if the user would like to continue. The first player is selected at random. Here, m is the number of rows, n is the number of columns, and k is the length of the sequence of marks arranged horizontally, vertically, or diagonally that constitutes a win.

For this problem we will implement three types of AI:

- AI Level 0. The program then places its own mark on a randomly chosen available position.
- AI Level 1. The program checks if it can win by placing its mark in any of the available positions. If so, it makes the winning move. Otherwise, it places its mark on a randomly chosen available position.
- AI Level 2. The program checks if it can win by placing its mark in any of the available positions. If so, it makes the winning move. If not, it checks if the opponent can win the game on the next move by placing their mark on one of the available positions. If so, it blocks that position with its own mark. Otherwise, it places its mark on a randomly chosen available position.

Part 2
This part builds upon Part 1. AI0 makes random moves. AI1 checks if there are winning moves and makes them. AI2 does what AI1 does and also prevents the opponent from winning on the next move if possible. We will play these strategies against each other. The question below should be addressed for the standard board $(m, n, k) = (3, 3, 3)$, for $(m, n, k) = (4, 4, 4)$, and for $(m, n, k) = (4, 3, 3)$. Some games will result in draws. We will count these games towards the total number of games, but not towards winning. Thus, for each scenario you need to obtain two winning probabilities. One minus the sum of them will be the probability of a draw.

Determine the probabilities of winning for AI1 vs. AI0, AI2 vs. AI0, and AI2 vs. AI1. Perform enough simulations for each scenario to determine the smallest probability with 5% accuracy. Explain why you believe that you achieved this accuracy.