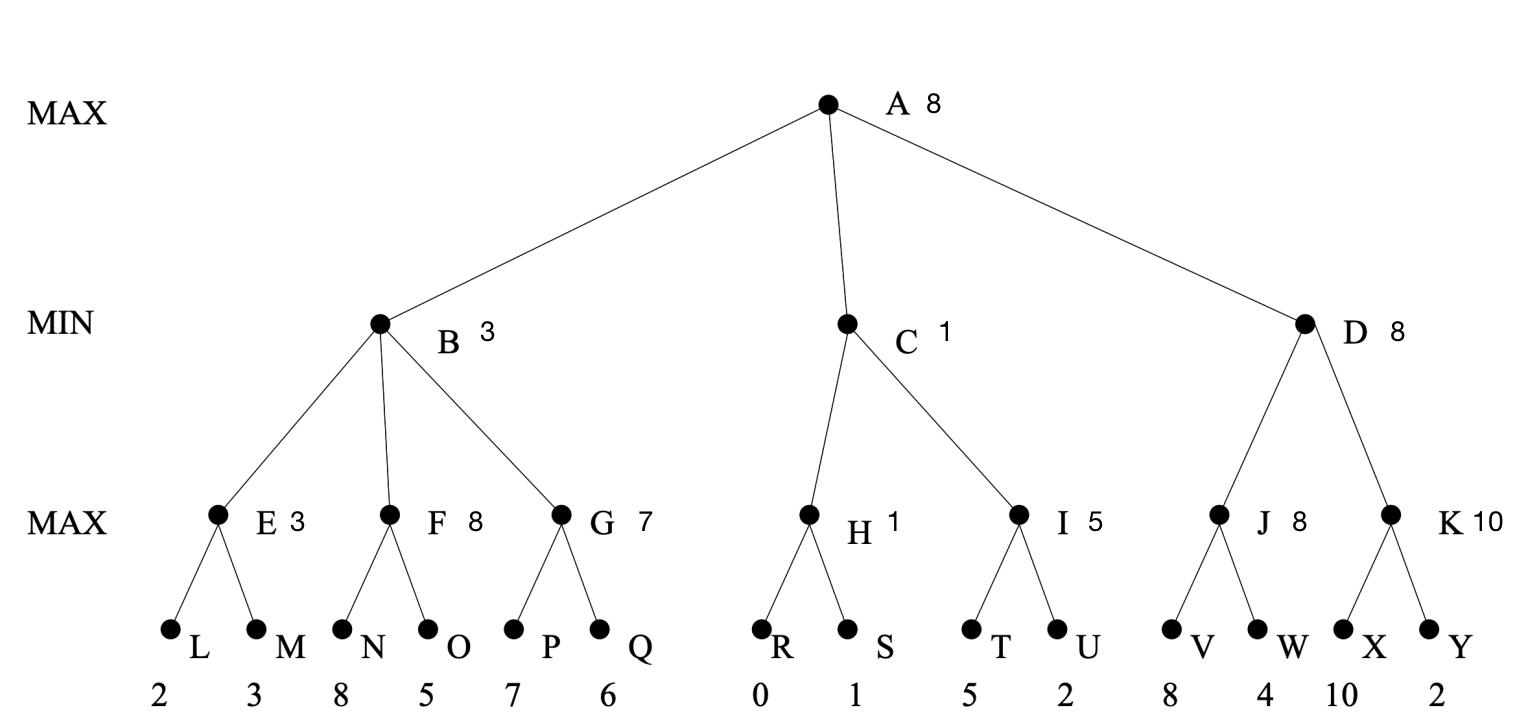
**Problem assignment 3**

*Due: Wednesday, October 2, 2019*

**Problem 1**

Part a.

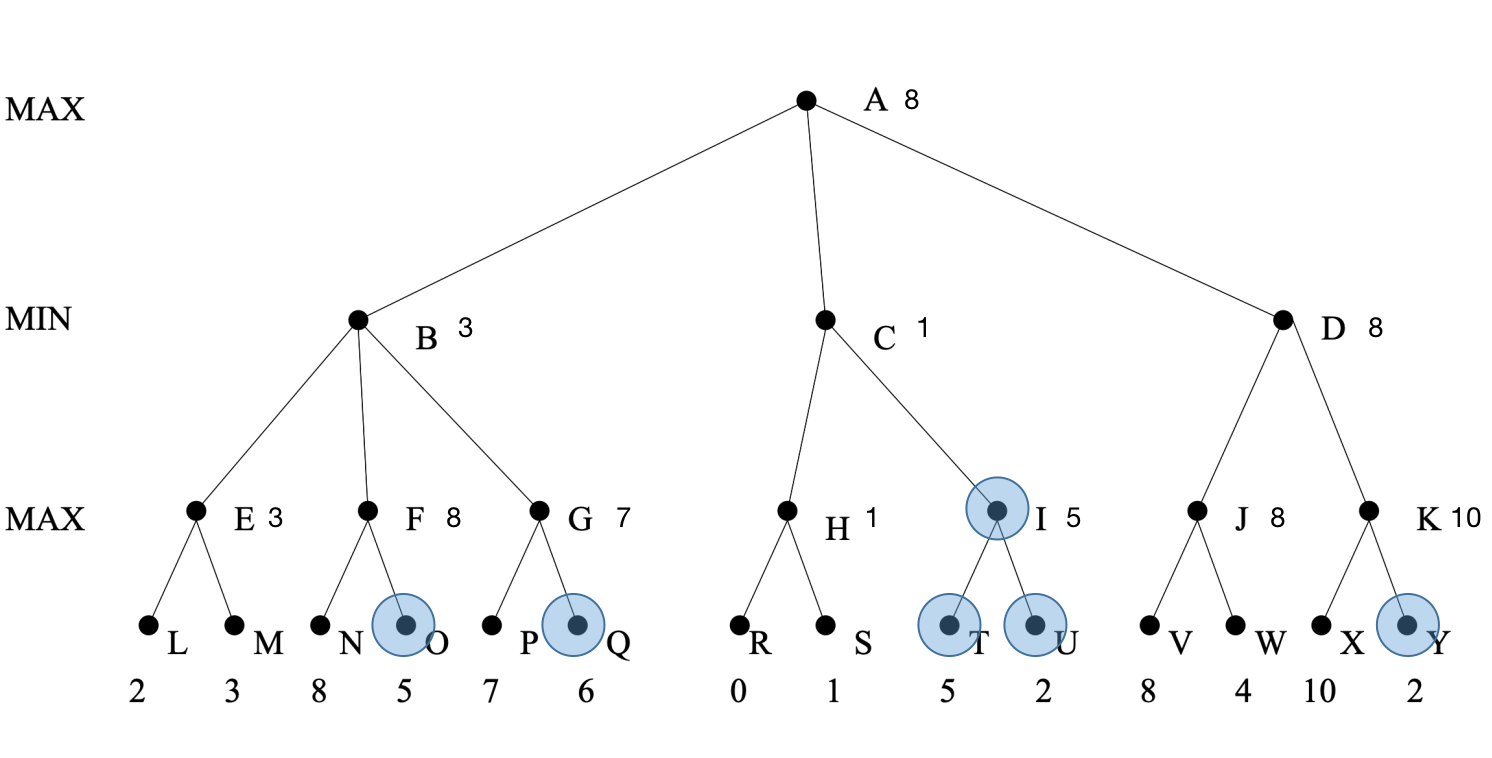


A-8, B-3, C-1, D-8, E-3, F-8, G-7, H-1, I-5, J-8, K-10, L-2, M-3, N-8, O-5, P-7, Q-6, R-0, S-1, T-5, U-2, V-8, W-4, X-10, Y-2.

The first player would choose the right now maximum utility. Among the 3 children of A, child D has the maximum values. So the first player would choose move D.

The solution path the rational players would play is A-D-J-V.

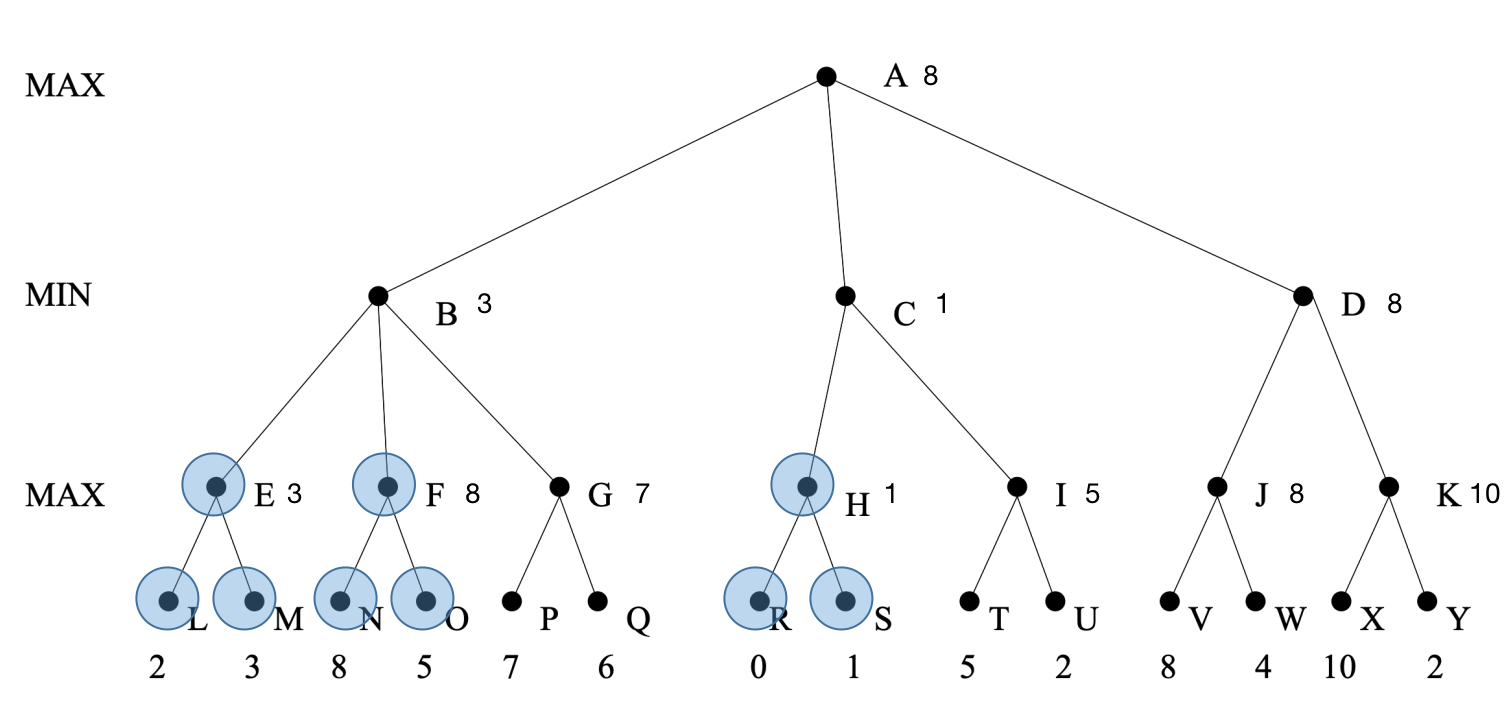
Parr b.



The nodes that are cut off from the tree and are never examined by the alpha beta procedure in left-to-right:

O, Q, I, T, U, Y.

Part c.



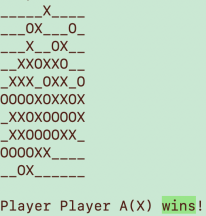
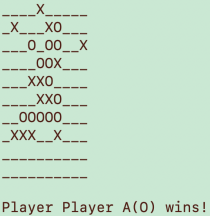
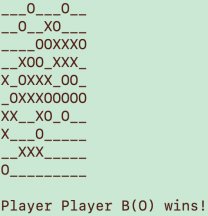
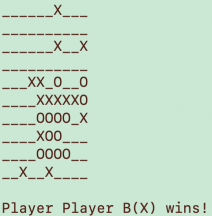
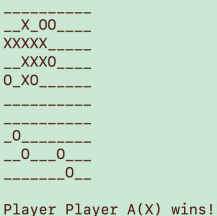
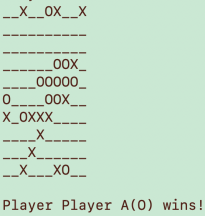
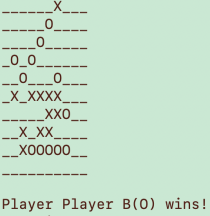
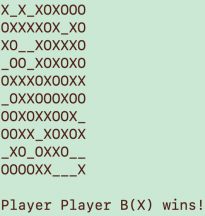
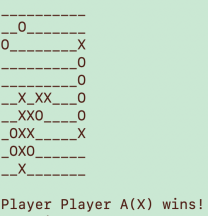
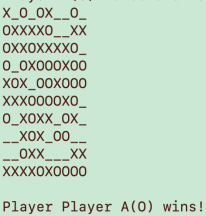
The nodes that are cut off from the tree and are never examined by the alpha beta procedure in right-to-left:

H, R, S, F, N, O, E, L, M.

Problem 2

Part a.

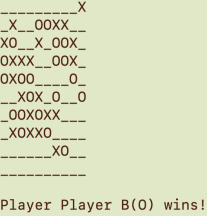
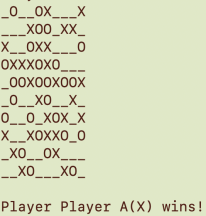
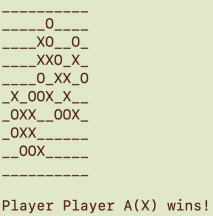
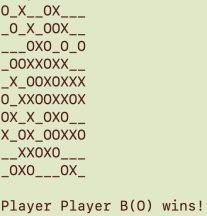
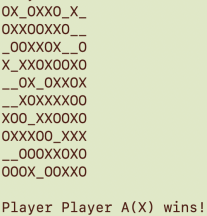
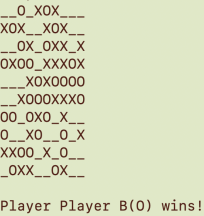
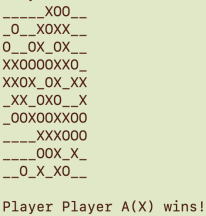
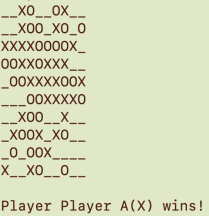
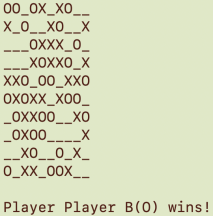
Player A wins: 6, Player B wins: 4, Tied: 0.

As figures showed, player A wins slightly more than B. ‘X’ wins equal to ‘O’. That’s because A uses basic heuristics and B uses naive heuristics. The basic heuristics give 8 different kinds of estimate of patterns and naive only gives 3. So, the basic one can make more reasonable heuristic of each pattern and react to the ‘danger’ earlier than naive one.

Part b.

Player A wins: 6, Player B wins: 4, Tied: 0.

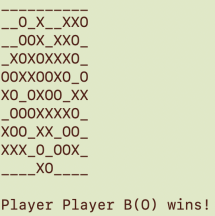
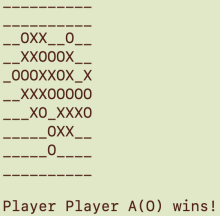
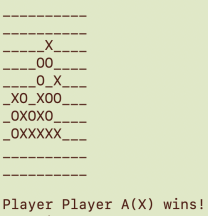
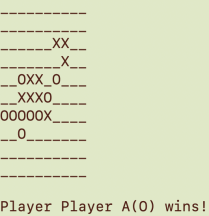
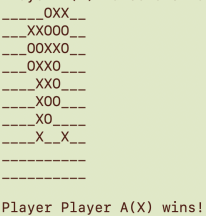
As figures showed, player A wins slightly more than B. Since player A and player B use the same heuristic method, the difference only lies in the starting. Player A always starts first. So starting first may somehow have advantage in the game.

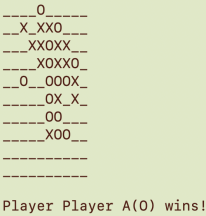
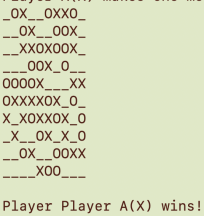
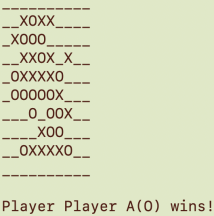
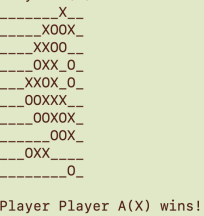
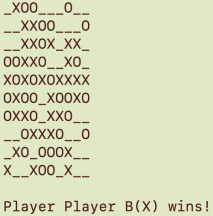
Part c.

Players A using K-ply = 2

Players B using K-ply = 1

Player A wins: 8, Player B wins: 2, Tied: 0.





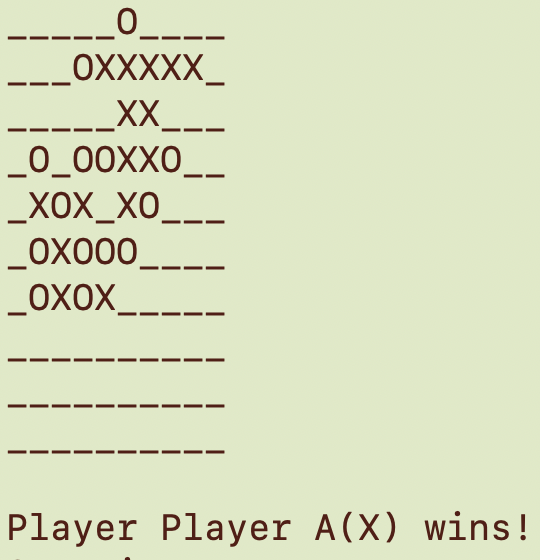
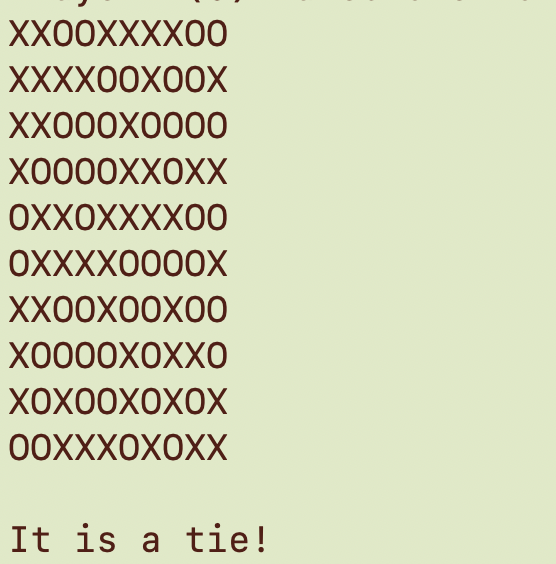
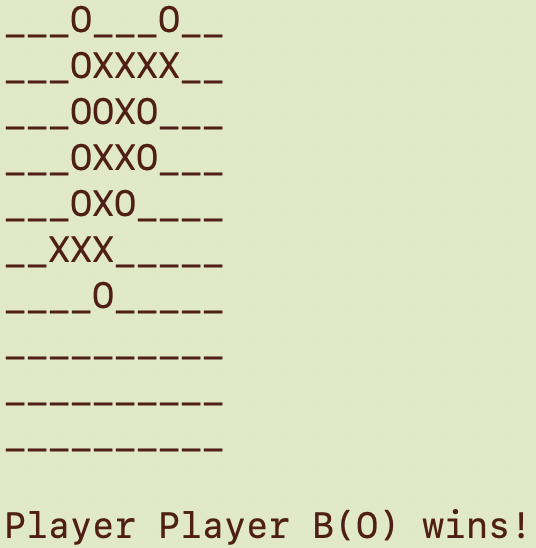
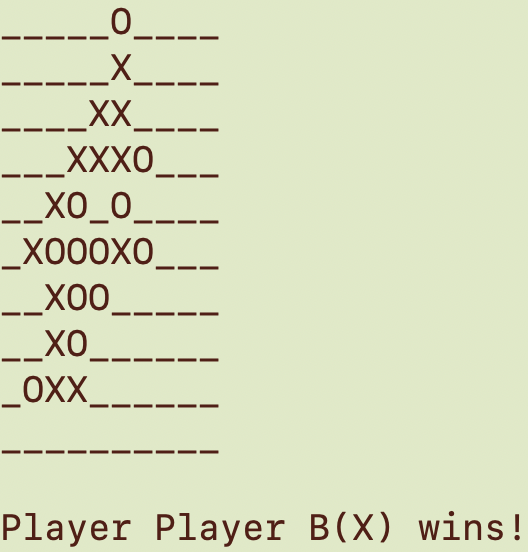
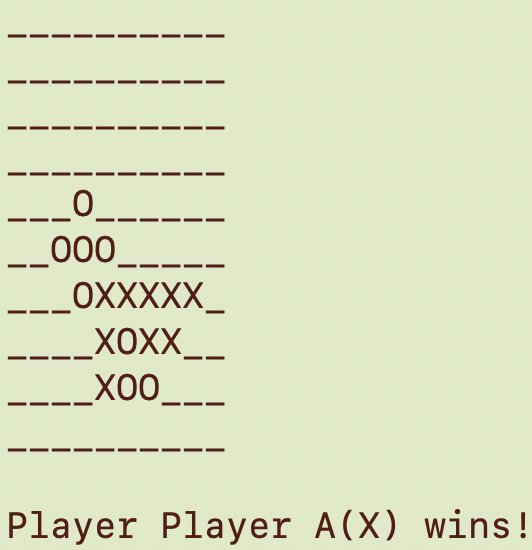
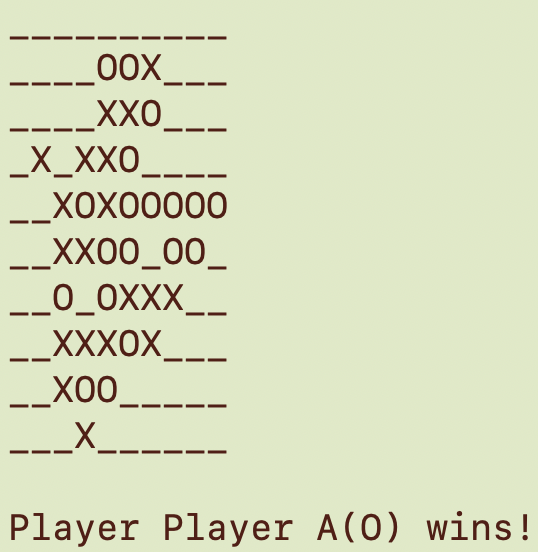
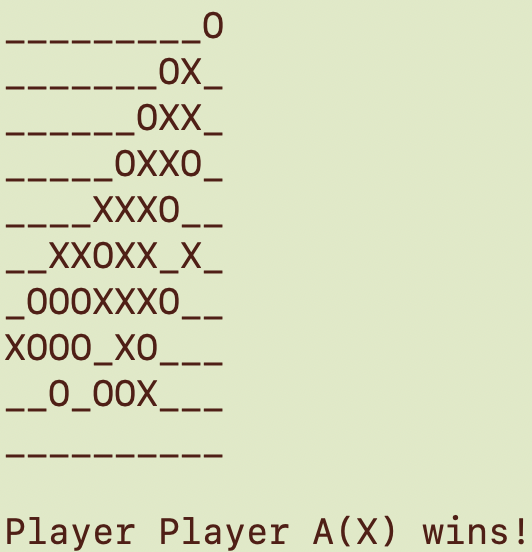
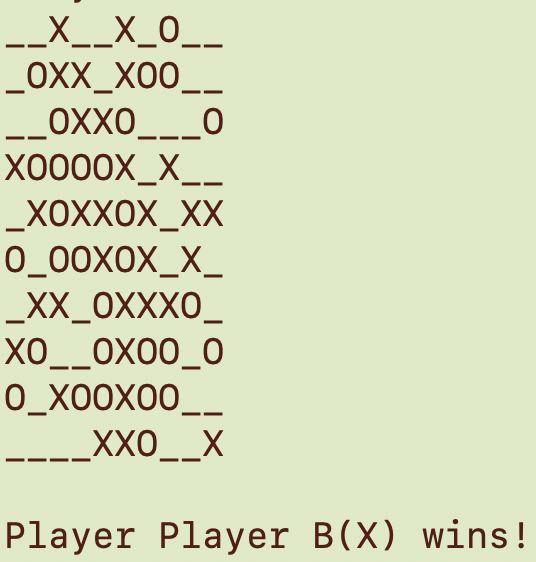
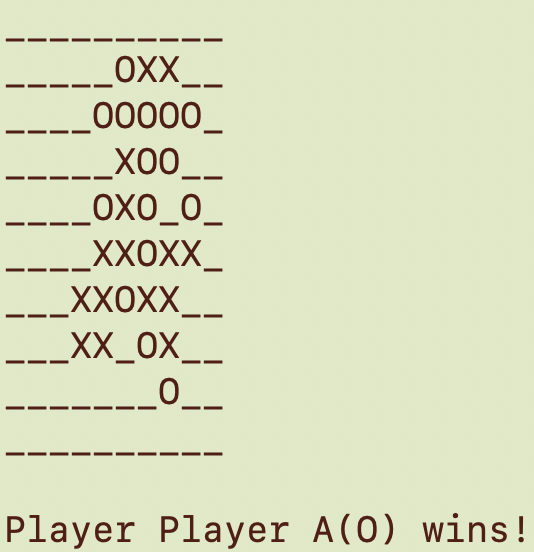
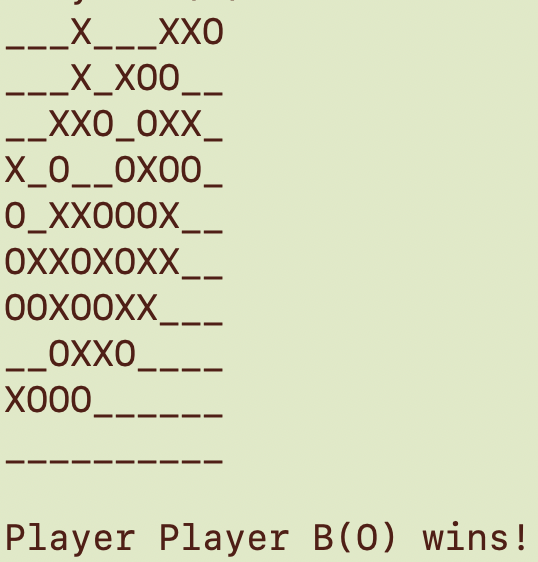
This time, player A nearly got the complete win of B. This indicates that when using the same heuristic method, the K plays an important role in the heuristic. As the K gets bigger, the adversarial search tree’s depth is increasing. Therefore, when choosing where to put the next step, higher K means more reasonable heuristic for possible steps. The players tend to make more rational moves. That’s why K-ply = 2 is better than K-ply = 1.

Players A using K-ply = 3

Players B using K-ply = 2

Player A wins: 5, Player B wins: 4, Tied: 1.

I waited for nearly whole long night to get the final results. The exponential explosion of the search space is really horrible.

However, the result doesn’t show great difference between K-ply=2 and K-ply=3, player A only win 1 more game than player B. This result means although with the adversarial search tree goes deeper, the estimation of each next-move is more reasonable, sometimes the depth of the adversarial search tree won’t increase the performance a lot. What’s worse, the time cost of the algorithm might rise greatly. As these time cost couldn’t lead to remarkable performance improvement, we shouldn’t always try to better the performance by just increase the K, some other methods should be applied, like pruning redundant branches.

Part d.

I’ve tried a lot of different patterns, however, none of them preformed very well at playing with basic heuristics.

My strategies are:

1. Try to maximize the utility of patterns ‘\_\_iii\_\_’ or ‘\_iiii\_’, these patterns may lead player to win the game.
2. Make great use of “active two”. “active two” means the patterns like ‘\_\_ii\_’ or ‘\_ii\_’ or ‘\_\_ii\_\_’. If we put next step at each of these empties, they will become three consecutive patterns, which is desirable for the player.
3. Make wise defense. While defending, try to keep the opponent before it reach 3 or 4 consecutive patterns so that we can put more cost on offense rather than defense.

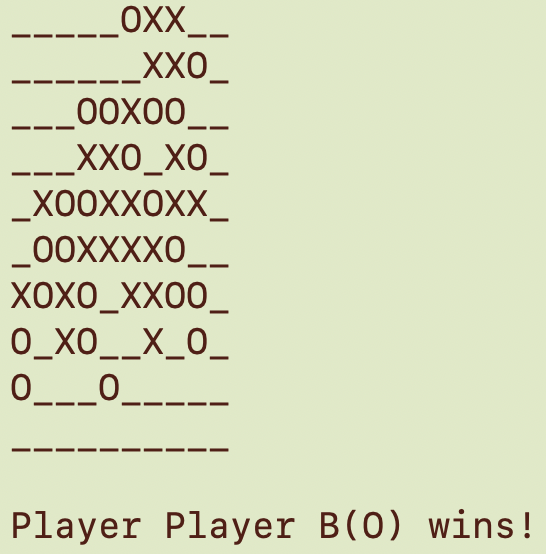
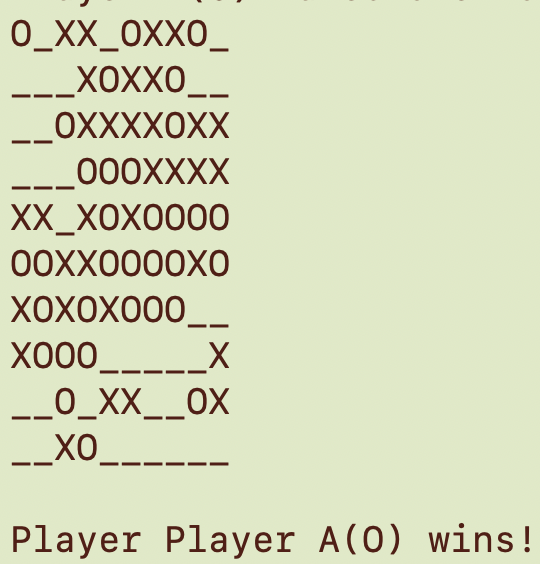
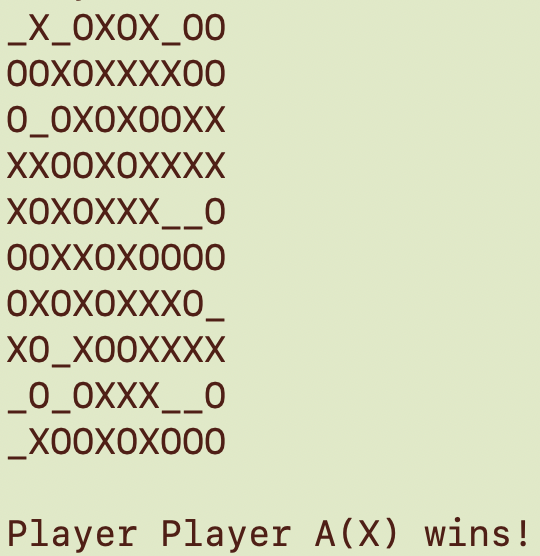
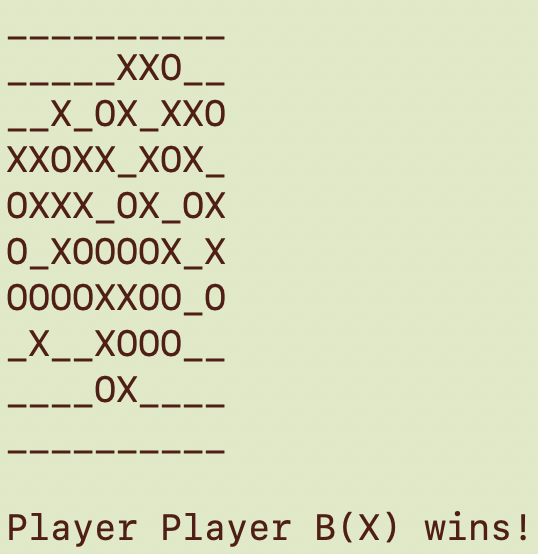
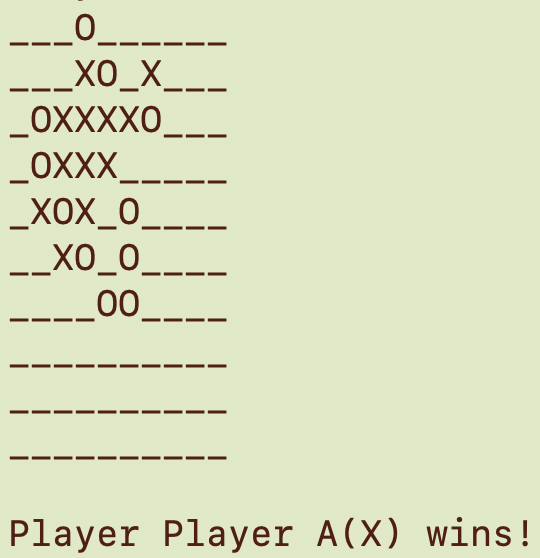
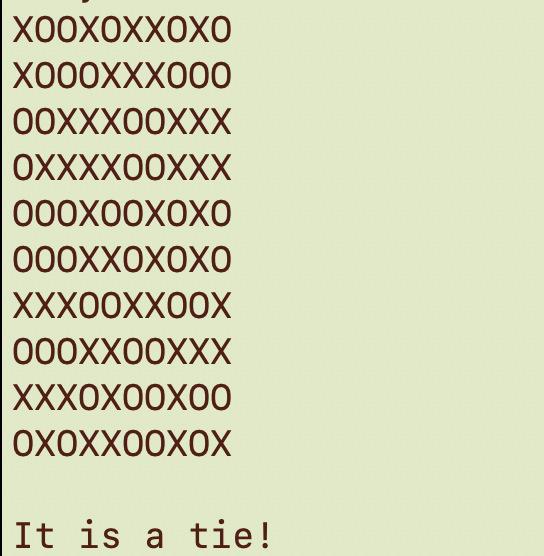
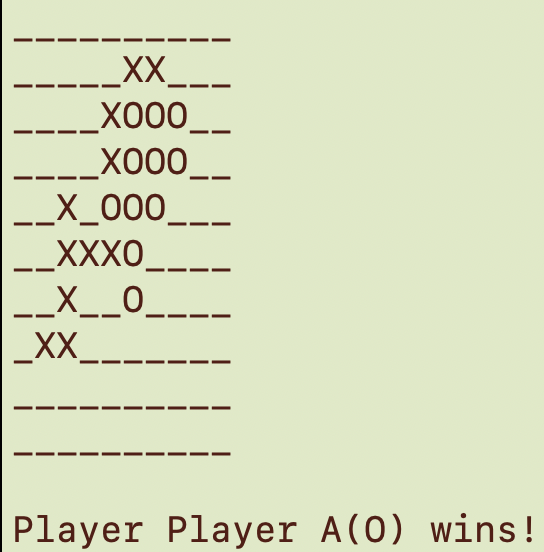
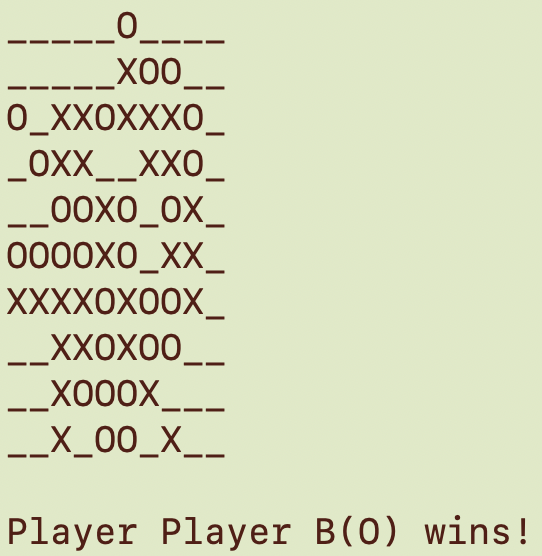
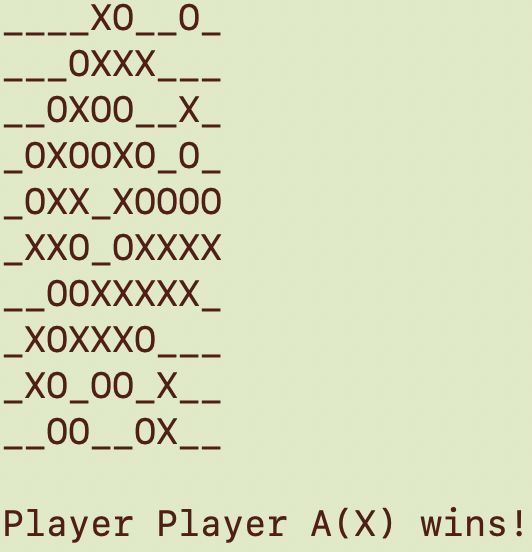
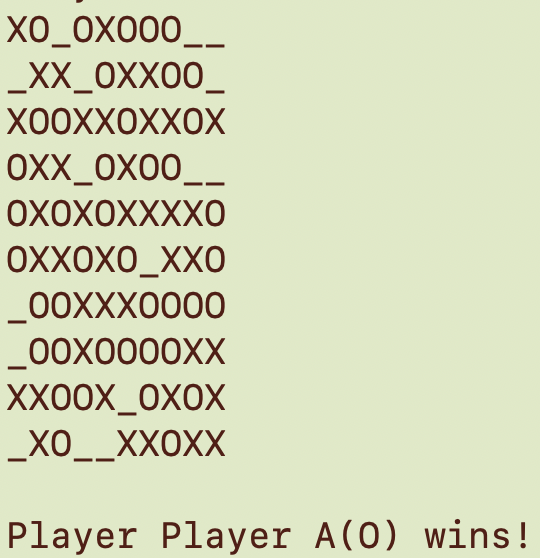
Here’s the patterns:

'iiiii': 100,  
 '\_iiii\_': 80, #  
 'ii\_ii': 80, #  
 '\_iii\_': 20,  
 '\_\_iii\_\_': 30, #  
 'iii\_\_': 20,  
 '\_i\_ii\_': 10,  
 '\_ii\_\_': 10,  
 '\_\_ii\_': 10,  
 'ii\_\_\_': 3, #  
 '\_\_i\_\_': 1,  
 'itttt\_': 4, #  
 'ittt\_': 1, #  
 '\_ttt\_': 5, #  
 '\_tt\_t\_': 2, #  
 'itt\_t\_': 1, #  
 't\_t\_t': 1 #

The Patterns with ‘#’ mean new added or modified.

Here’s the result.

Player A wins: 6, Player B wins: 3, Tied: 1.

As shown above, my heuristics of patterns perform well when the board become crowded.