AI - CHECKERS SCORING DESIGN
One submission in the online lecture survey per 2-4 person team (from the same lab)
due by midnight Sunday, 11/10. Include all team member names.

Checkers is played by two opponents. Each player has 12 checkers (of their color) on the black squares of the first three rows on the opposite ends of the board. The object of the game is to capture all of the opponent's checkers, or block them so they cannot be moved. The checkers are moved diagonally and each player moves alternately one of his checkers. In order to "capture" an opponent's checker, your checker must jump over the opponent checker when there is a vacant square behind it. Single men may move diagonally forward. When a checker has reached the last row of his opponent's side, it becomes a "King" and can now move diagonally forward or backward. The "King" is "crowned" by placing another checker on top of it. When there is a "jump" available, the player must jump. You are allowed to jump as many of the opponent's men (with a single checker of yours) on the same move if there are vacant squares diagonally behind each. Rows are numbered from top to bottom and left to right (from 1 to 8)

0	1	2	3	4	5	6	7	8
1		W		W		W		W
2	W		W		W		W	
3		W		W		W		W
4	_		_		_		_	
5		_		_		_		_
6	B		B		B		B	
7		B		B		B		B
8	B		B		B		B	

To learn how or practice playing checkers, you can find free apps to play checkers against another person.

Your team's assignment is to design various heuristics (at least 3) to "score the board" for a game theory program to play and win checkers. Basically, when shown a checkers game in progress (not yet an end game) and given a player color (black or white), you need ideas for how you can determine a relative score for the board for that player color that somehow predicts their chance of winning the game (positive score for good chance of winning and negative score for good chance of losing).

In lab next week your team will code various "score the board" methods and try them out with game theory code I provide. Your ultimate goal is to pick one of your "score the board" methods to make the best checkers playing computer program to defeat your fellow CS100 students' programs.

Your team should answer these questions in the online lecture survey, due by midnight Sunday night. Whoever submits the survey for the team will get the responses back via email. Include all team member names.

1. Do you think all pieces should have the same value towards winning?
2. Do you think all locations on the board should have the same value towards winning?
3. Are you going to change your strategy as the game progresses?
4. Are you going to change your strategy depending on who goes first?
5. The min-max algorithm only chooses a new move when the board score exceeds the current best and does not choose a new move if the board score ties the current best. Is your strategy going to include something so your computer does not always play the same move given the same board?
6. Write designs for your 3 (at least) heuristics to "score the board" that you will code and test in lab next week.