CSC 290	Introduction to Artificial Intelligence	ASSIGNMENT 5
PLEASE SUBMIT YOUR WORK ON E-Mail (Also keep a copy on sent items for just in case).		

Due Date: Apr 10

The assignment is individual, so please do not make any information exchange or do not discuss about the answers or ask anybody for help except the instructor. You can use any Internet resources for reading, you can search for the answers on any search engine (like Google) or you can use any textbooks.
For all the parts below, please write your code in Lisp Programming Language and test with Dr. Racket and RackLog package.

Assumptions

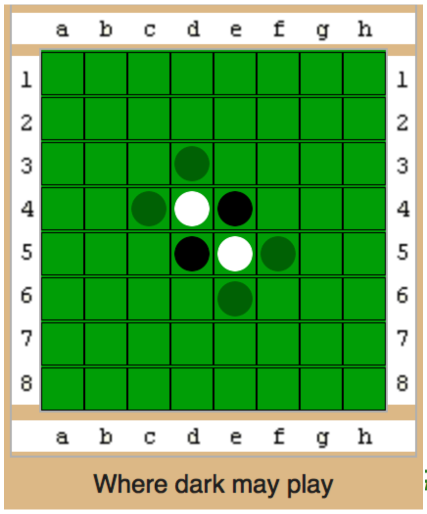
Assume you have asked to implement a reversi game. Implement the game rules as a knowledge base and implement an intelligent agent that you can ask for all possible moves regarding the knowledge base. So your agent will have two types of predicates: 1. Tell Statements to teach game rules and any board position, 2. Ask statements to query all the possible moves according to the position on board and game rules.
You can read the game rules from Wikipedia page: https://en.wikipedia.org/wiki/Reversi
Also please don't go fail to the discussion between classical or modern versions. This is totally irrelevant to your assignment, you are asked to return the possible moves for a given situation, you do not care how the game starts or what is the initial placement of pieces on the board.

Questions

1) Implement a knowledge based agent for listing all possible moves for any given position with the Reversi game rules.

Testing

You are free to define your data structure for the board. For example you can define a board with 8 lists, each holding 8 members (so 8x8 =64 cells) or you can define a data structure to hold only cells with pieces or any other data structure you fell proper for the assignment.
Lets consider the initial game setup like below and a sample function call to the knowledge base is demonstrated besides it.



```
(%which (move)
        (%reversi 'black move)
('d 3)
(%more)
('c 4)
(%more)
('f 5)
(%more)
('e 6)
(%more)
#false
```

The demo execution above is just a sample to give idea, so you can use completely different function or relation names, you can code the cell coordinates with different notation (like numbers for both column and row coordinates at the same time), etc.