# Chess Bounty

1. **Introduction**

Cyber Defenders Program is a great program that helps us improve our skills such as communication, research, daily stand up, coding, etc. However, we need more practice to understand Python deeper, and this game would teach you how to manage a list in Python.

Last Friday, Cit and me took a 1-hour chess game, and it was a draw game. It reminded me about an assignment that I have done couple years ago. Here it is!

There are only three chessman left, one Black King, one White King, and one White Rook. Black King is trying to survive. White King and White Rook are trying to get the Black King.
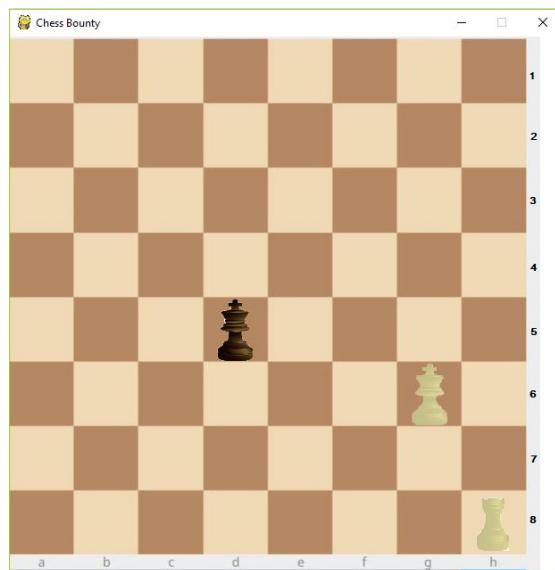

2. **Chess Basic**

The Rook moves horizontally and vertically any number of squares, forwards or backwards. In the diagram the Rook can move to any of the highlighted squares.
If an enemy piece is standing on a square to which the Rook can move, it can be captured
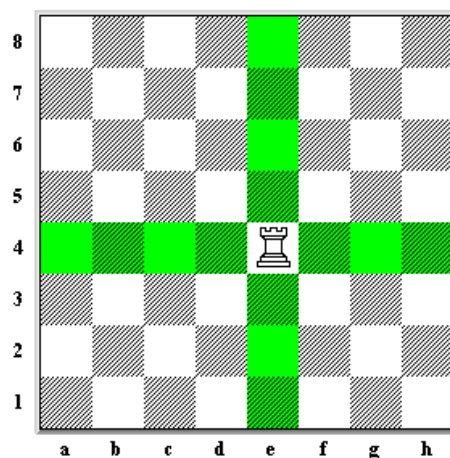
The king is the most important piece, but is one of the weakest. The king can only move one square in any direction - up, down, to the sides, and diagonally. The king may never move himself into check (where he could be captured). When the king is attacked by another piece this is called "check".
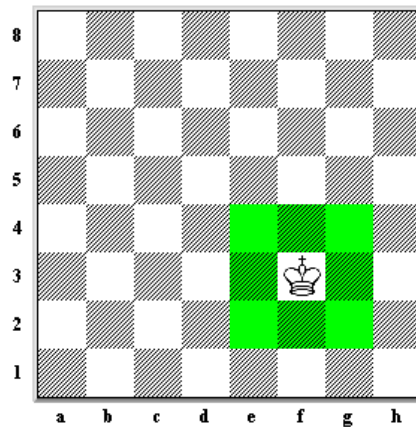
3. **Translate chess problem to coding?**
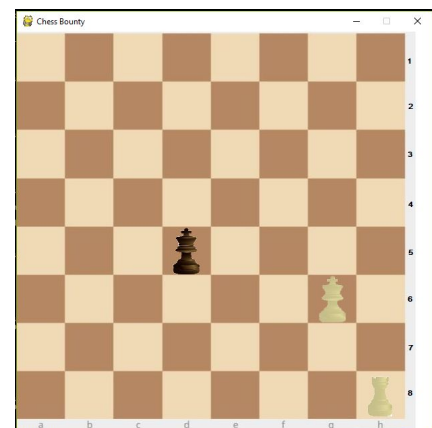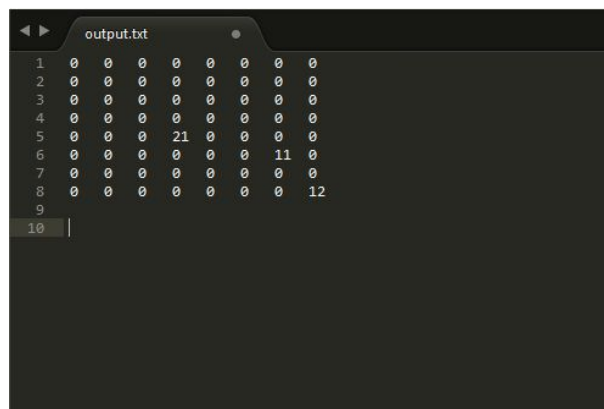   a. How to manage a chess board as a list in Python?
      White King = 11, White Rook = 12, Black King = 21
      ● List in Python: Using **2-dimension list**
      [[0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 21, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 11, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 12]]
      ● In text file (output file) and Visualization

   b. How do the program work?
      ● **ChessBounty.py** (main class): Run algorithm and visualization
      ● **Board.py**: is to generate a random board status (Kings and Rook are generate random position), check moves if it is valid or not, check game if it finish or not, write board status and moves in output file (output.txt).

- **Black.py**: is to develop Black King algorithms
- **White.py**: is to develop White King and White Rook algorithms (your task)
- **BoardUI.py**: is to visualize moves of games

4. **Task**

   Your task is to implement an algorithm that guide the White King and White Rook get the Black King

   ```
   def decide(self, boardStatus,i):
       '''
       Your work should be implemented here
       '''
       return boardStatus
   ```

   In the White.py, you will implement your code in decide function. *boardStatus* is a status of the board after the Black King move. After this function is called, your move should be returned within the boardStatus variable (White King's or White Rook's). Your move will be sent to my Black.py, my Black file will decide and send its move back to your White file for a new move. The process will repeat many times before my Black King die.
   YOU CANNOT SKIP YOUR MOVE
   Example:
   boardStatus (input) is [[0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 21, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 11, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, **12**]]
   Return *boardStatus* (output) is [[0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 21, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 11, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, **12**, 0, 0, 0]]
   It mean your White Rook (=12) moved form **h8** to **e8**

5. **Run the program**

   Clone the program from github: https://github.com/nguyenhoaihuy/ChessBounty.git
   Use a command line `python ChessBounty.py` to run the program
   A board will popup, you can use -> to show moves or SPACE bar to reset the board.
   For test board: Use the command line `python ChessBounty.py -t`

6. **How do I check your program?**

   When you think your code is perfect, email me you White.py. I will use 3 random test cases to check your program because I don't want you just code for specific board.

7. **Hint**

   Here is how I 'train" my Black King
   - List all possibilities that my Black King could go
   - Grade each possibility to decide which one should go next
       - If a cell in that list is the move of White rook, -50 point
       - If a cell in that list is the move of White King, -50 point
       - If a cell in that list is the White Rook's position, +20 (because the Rook may be protected by the King)
       - If a cell in that list is the White King's posision, +100 (White King die :))

- More close to the Rook, more point (because Black king wants to get the Rook to have a draw game)

Good Luck!!!!