Chess Validator

Problem Statement

Create a library which can validate a sequence of chess moves. Then create an application using your library which takes as input (from the command line or text file):

- The starting state of the board
- A sequence of moves

For each move your program should:

- Validate the move based on the rules of chess (detailed out below)
- If the move is invalid, throw an error and do not change the state of the board
- If the move is valid, update the board and display the updated state

Input

The input is the starting state of the board followed by a sequence of moves.

A board is represented as a 2D 8x8 grid of cells. Each cell of the board is represented by two characters.

The first character of an occupied cell indicates the color and the second character indicates the piece.

1. **Colors**: W (white) , B (black)
2. **Pieces**: R(ook), B(ishop), H(orse), K(ing), Q(ueen), P(awn)

Example: WB is a White Bishop

An Empty Cell is represented by “--”

A move is represented as follows:

```
<type-of-piece> <starting-cell> <ending-cell>
```

Where

- Type of piece: Same two character representation described above
- Starting and ending cells: A two digit integer where the first digit indicates the row and the second digit represents the column. Rows are numbered from top to bottom, starting at 0. Columns are numbered from left to right, starting at 0.

Sample input

```
Board
WR WH WB WQ WK WB WH WR
WP WP WP WP WP WP WP WP
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
```

```
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
BP BP BP BP BP BP BP BP
BR BH BB BQ BK BB BH BR
Moves
WP 12 22
BP 60 64
BH 71 50
WQ 03 30
```

Sample output

```
Move 1: WP 12 22 [Valid]

           WR WH WB WQ WK WB WH WR
           WP WP -- WP WP WP WP WP
           -- -- WP -- -- -- -- --
           -- -- -- -- -- -- -- --
           -- -- -- -- -- -- -- --
           -- -- -- -- -- -- -- --
           BP BP BP BP BP BP BP BP
           BR BH BB BQ BK BB BH BR

Move 2: BP 60 64 [Invalid]

           Error: Invalid

           move

Move 3: BH 71 50 [Valid]

           WR WH WB WQ WK WB WH WR
           WP WP -- WP WP WP WP WP
           -- -- WP -- -- -- -- --
           -- -- -- -- -- -- -- --
           -- -- -- -- -- -- -- --
           BH -- -- -- -- -- -- --
           BP BP BP BP BP BP BP BP
           BR -- BB BQ BK BB BH BR

Move 4: WQ 03 30 [Valid]

           WR WH WB -- WK WB WH WR
           WP WP -- WP WP WP WP WP
```

```
-- -- WP -- -- -- -- --
WQ -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
BH -- -- -- -- -- -- --
BP BP BP BP BP BP BP BP
BR -- BB BQ BK BB BH BR
```

Expectations

1. Clean API design for the library.
2. Clean internal design and implementation of the library and the application.
3. **Functional completeness** - Able to model all the different pieces and their movements correctly (as described below). Must complete movement for **Queen and Pawn**, if time permits implement for other pieces. Focus on completing a piece before moving to another one.
4. Extensibility
5. Take care of Exception and Corner case handling.

Chess summary

Chess board

A checkered board with 64 squares (eight rows and eight columns)

Chess pieces

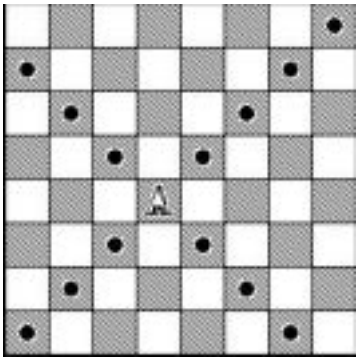
Two armies of 16 chess pieces, one army white, the other black. Each player controls one of the armies for the entire game. The pieces in each army include:

1 king , 1 queen , 2 rooks , 2 bishops , 2 knights, 8 pawns

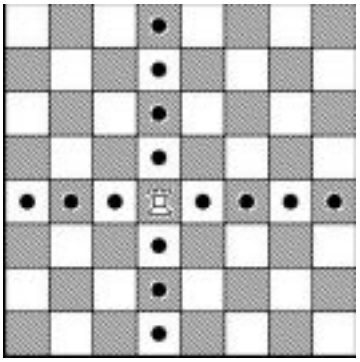
Valid moves

- A piece may not move to a square occupied by another piece of the same color.
- **Capture** – move of a piece to a square occupied by an opposing piece, which is removed from the board and from play.
- **Moving a pawn** – pawn move straight forward one space at a time, but capture diagonally (within a one-square range). First move of pawn can move two spaces in real chess.
- **Moving a horse (knight)** – Knight move two squares horizontally and one square vertically, or two squares vertically and one square horizontally, jumping directly to the destination while ignoring intervening spaces.

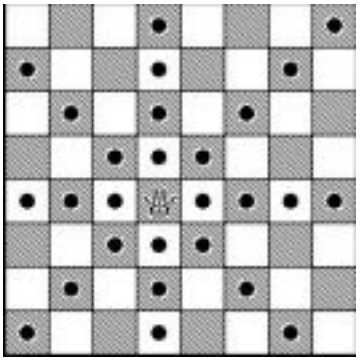
- **Moving a bishop** – Bishop move any distance diagonally (without crossing across any other piece). One bishop in each army moves diagonally on white squares only, and the other bishop is restricted to moving along black squares.



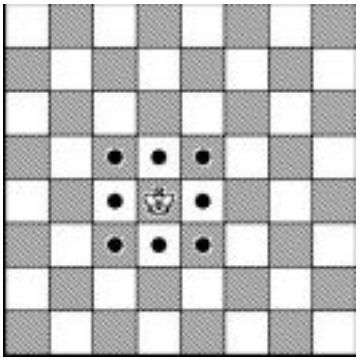
- **Moving a rook** – Rook may move any distance vertically or horizontally (without crossing across any other piece)



- **Moving the queen** – Queen can move like a rook or bishop.



- **Moving the king** – King may move one square in any direction.



Resume and Code Upload : https://forms.gle/aJknR4WuoZ3UPGGS9