

## COMS4033A/7044A Assignment State Representation

### 1 Introduction

When tackling any kind of decision-making or search process, one of the key considerations is how to represent the state of the world. In chess (and chess variants) a standard representation is known as Forsyth–Edwards Notation (FEN), which encodes the position of all the pieces on the board, as well as other chess-related flags. You are encouraged to read more about FEN strings here: [https://en.wikipedia.org/wiki/Forsyth%E2%80%93Edwards\\_Notation](https://en.wikipedia.org/wiki/Forsyth%E2%80%93Edwards_Notation) although it is not strictly necessary to understand every aspect of the string—only that it fully represents a given state of the game.

The first part of the assignment will be dealing with the board's representation. This involves taking FEN strings as input, as well as converting boards to their corresponding FEN string.

## 2 Board Representation

In the first submission, you are required to read in as input a FEN string and set up the chess board appropriately. You should then output the current position using ASCII art. While this sounds difficult, the `python-chess` package that was installed when you installed the `reconchess` provides classes for doing so. Before continuing, please browse the documentation for this package here: <https://python-chess.readthedocs.io/en/latest/index.html> and be sure to check out the various classes in the core package

Having read through the documentation, write a Python program that accepts a FEN string as input and outputs an ASCII representation of the board, as in the following sample cases:

### Sample Input #1

```
rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
```

### Sample Output #1

```
r n b q k b n r
p p p p p p p p
. . . . .
. . . . .
. . . . .
. . . . .
P P P P P P P P
R N B Q K B N R
```

### Sample Input #2

```
r1bqkb1r/pppp1Qpp/2n2n2/4p3/2B1P3/8/PPPP1PPP/RNB1K1NR b KQkq - 0 4
```

### Sample Output #2

```
r . b q k b . r
p p p p . Q p p
. . n . . n . .
. . . . P . . .
. . B . P . . .
. . . . .
P P P P . P P P
R N B . K . N R
```

### 3 Move Execution

Now that we can represent the board, we must next test to ensure actions can be executed. Write a Python program that accepts a FEN string representing the current state of the board, and a move to be played. The program should output the resulting position using a FEN string. The move is described using a string that represents the start and end square that the pieces moves to and can be converted to a Move object using `Move.from_uci("a8a1")`, where `a8a1` is an example move describing a piece moving from square A8 to A1.

#### Sample Input #1

```
rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1  
e2e4
```

#### Sample Output #1

```
rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq - 0 1
```

#### Sample Input #2

```
8/8/8/2k5/4K3/8/8/8 w - - 4 45  
e4e5
```

#### Sample Output #2

```
8/8/8/2k1K3/8/8/8/8 b - - 5 45
```