

This program is written in Python.

```
import chess
import random
import asyncio # Enables non-blocking I/O operations
import time

def check_board(stalemate, checkmate): # Checking board for seeing whether there is stalemate or
checkmate.
    if(stalemate==1 and checkmate==1):
        print(board.is_checkmate(), " - Checkmate")
        print(board.is_stalemate(), " - Stalemate")

        if board.is_stalemate(): # Stalemate
            print("StaleMate, Game Over")
            time.sleep(10)
            sys.exit()

        elif board.is_checkmate(): # Checkmate
            print("Game Over")
            time.sleep(10)
            sys.exit()

    elif(stalemate==0 and checkmate==1):
        print(board.is_checkmate(), " - Checkmate")

        if board.is_checkmate():
            print("Game Over")
            time.sleep(10)
            sys.exit()

    else:
        pass;
    print("=====")

async def main():
    global board
    chess.STARTING_FEN = 'rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1' #
Starting position
    print(chess.Board(chess.STARTING_FEN)) # Printing a small chess board
    board = chess.Board()

    print("""
```

Understanding the notation:

To input moves according to notation, you can use the board below. The squares are labeled with numbers and letters.

```
|---|---|---|---|---|---|---|
| a8 | b8 | c8 | d8 | e8 | f8 | g8 | h8 |
|---|---|---|---|---|---|---|
| a7 | b7 | c7 | d7 | e7 | f7 | g7 | h7 |
|---|---|---|---|---|---|---|
| a6 | b6 | c6 | d6 | e6 | f6 | g6 | h6 |
```

```

|---|---|---|---|---|---|---|
| a5 | b5 | c5 | d5 | e5 | f5 | g5 | h5 |
|---|---|---|---|---|---|---|
| a4 | b4 | c4 | d4 | e4 | f4 | g4 | h4 |
|---|---|---|---|---|---|---|
| a3 | b3 | c3 | d3 | e3 | f3 | g3 | h3 |
|---|---|---|---|---|---|---|
| a2 | b2 | c2 | d2 | e2 | f2 | g2 | h2 |
|---|---|---|---|---|---|---|
| a1 | b1 | c1 | d1 | e1 | f1 | g1 | h1 |
|---|---|---|---|---|---|---|

```

```

|---|---|---|---|---|---|---|
| r | k | b | q | k | b | k | r |
|---|---|---|---|---|---|---|
| p | p | p | p | p | p | p | p |
|---|---|---|---|---|---|---|
| | | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | | |
|---|---|---|---|---|---|---|
| P | P | P | P | P | P | P | P |
|---|---|---|---|---|---|---|
| R | K | B | Q | K | B | K | R |
|---|---|---|---|---|---|---|

```

```

    """)

```

```

    print("")

```

The chessboard shown above consists of certain pieces.

Pieces written in lowercase belong to black, and pieces written in uppercase belong to white.

P, p: Pawn
R, r: Rook
N, n: Knight
B, b: Bishop
Q, q: Queen
K, k: King

```

    """)

```

```

    white_list = ["White", "white", "w", "W"] # List-1

```

```

    black_list = ["Black", "black", "b", "B"] # List-2

```

```

    inputa = input("Would you like to be White or Black in this chess game?: ") # Input

```

```

    if(inputa in white_list):

```

 b = input("Please play a move (Write the move in appropriate notation): ") # Algebraic notation is the standard method for recording and describing the moves in a game of chess.

 # See:

[https://en.wikipedia.org/wiki/Algebraic_notation_\(chess\)](https://en.wikipedia.org/wiki/Algebraic_notation_(chess))

```

    while True:

```

```

    try: # Is it legal move?
        board.push_san(b)
        break;
    except:
        b = input("This is not a legal move, try again: ")
    print("=====")
elif(not(inputa in black_list)):
    print('You made a spelling mistake! Your side can only be White or Black. We continue with the
default preference "Black".')
    print("=====")
else:
    print("=====")
check_board(0,1)

while True:
    legal_moves = list(board.legal_moves) # Listing the legal moves.
    a = random.choice(legal_moves)
    await asyncio.sleep(2)
    board.push(a)
    print(board)
    board = chess.Board(board.fen())
    print("=====")
    print("YOUR TURN!")
    print("=====")
    check_board(1,1)
    legal_moves = list(board.legal_moves)
    b = input("Please play a move (Write the move in appropriate notation): ")
    while True:
        try:
            board.push_san(b)
            break;
        except:
            b = input("This is not a legal move, try again: ")
    print("=====")
    board = chess.Board(board.fen())
    check_board(1,1)

asyncio.run(main())

```

```
# This program is written in Python.
```

```
import chess
import random
import asyncio # Enables non-blocking I/O operations
import time

def check_board(stalemate, checkmate): # Checking board for seeing whether there is stalemate or checkmate.
    if(stalemate==1 and checkmate==1):
        print(board.is_checkmate(), " - Checkmate")
        print(board.is_stalemate(), " - Stalemate")

        if board.is_stalemate(): # Stalemate
            print("StaleMate, Game Over")
            time.sleep(10)
            sys.exit()

        elif board.is_checkmate(): # Checkmate
            print("Game Over")
            time.sleep(10)
            sys.exit()

    elif(stalemate==0 and checkmate==1):
        print(board.is_checkmate(), " - Checkmate")

        if board.is_checkmate():
            print("Game Over")
            time.sleep(10)
            sys.exit()

    else:
        pass;
    print("=====")

async def main():
    global board
    chess.STARTING_FEN = 'rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1' # Starting position
    print(chess.Board(chess.STARTING_FEN)) # Printing a small chess board
    board = chess.Board()
```

```
print("""
```

Understanding the notation:

To input moves according to notation, you can use the board below. The squares are labeled with numbers and letters.

a8	b8	c8	d8	e8	f8	g8	h8
a7	b7	c7	d7	e7	f7	g7	h7
a6	b6	c6	d6	e6	f6	g6	h6
a5	b5	c5	d5	e5	f5	g5	h5
a4	b4	c4	d4	e4	f4	g4	h4
a3	b3	c3	d3	e3	f3	g3	h3
a2	b2	c2	d2	e2	f2	g2	h2
a1	b1	c1	d1	e1	f1	g1	h1

r	k	b	q	k	b	k	r
p	p	p	p	p	p	p	p
P	P	P	P	P	P	P	P
R	K	B	Q	K	B	K	R

```
""")
```

```

print("""
The chessboard shown above consists of certain pieces.

Pieces written in lowercase belong to black, and pieces written in uppercase belong to white.

P, p: Pawn
R, r: Rook
N, n: Knight
B, b: Bishop
Q, q: Queen
K, k: King

""")
white_list = ["White", "white", "w", "W"] # List-1
black_list = ["Black", "black", "b", "B"] # List-2
inputa = input("Would you like to be White or Black in this chess game?: ") # Input
if(inputa in white_list):
    b = input("Please play a move (Write the move in appropriate notation): ") # Algebraic notation is the standard method for recording and describing the moves in a game of chess.
    # See: https://en.wikipedia.org/wiki/Algebraic\_notation\_\(chess\)
    while True:
        try: # Is it legal move?
            board.push_san(b)
            break;
        except:
            b = input("This is not a legal move, try again: ")
    print("=====")
elif(not(inputa in black_list)):
    print('You made a spelling mistake! Your side can only be White or Black. We continue with the default preference "Black".')
    print("=====")
else:
    print("=====")
check_board(0,1)

while True:
    legal_moves = list(board.legal_moves) # Listing the legal moves.
    a = random.choice(legal_moves)
    await asyncio.sleep(2)
    board.push(a)
    print(board)
    board = chess.Board(board.fen())
    print("=====")
    print("YOUR TURN!")
    print("=====")
    check_board(1,1)
    legal_moves = list(board.legal_moves)
    b = input("Please play a move (Write the move in appropriate notation): ")

```

```
while True:
    try:
        board.push_san(b)
        break;
    except:
        b = input("This is not a legal move, try again: ")
print("=====")
board = chess.Board(board.fen())
check_board(1,1)
```

```
asyncio.run(main())
```