```
# By submitting this assignment, I agree to the following:
#   "Aggies do not lie, cheat, or steal, or tolerate those who do."
#   "I have not given or received any unauthorized aid on this assignment."
#
# Name:         HUY Q LAI                   132000359
#                BRANDON A WHITE      331004571
#                WILLIAM A ROBERTS   530008478
#                JOHN RIOS JR              631004090
# Section:      ENGR-102-569
# Assignment:   Lab 7a Activity 1a
# Date:         12 October 2021
#

from colorama import Fore, Style


# Constants

blank = "."

red = Fore.RED + Style.BRIGHT + chr(9920) + Fore.RESET + Style.NORMAL

black = Fore.BLACK + Style.BRIGHT + chr(9922) + Fore.RESET + Style.NORMAL

board = [list(blank * 8) eight times]

coords = [

        hardcoded to not allow illegal moves

]



def reset(b: list[list]) -> None:

        # Reset board


        # remove all pieces

        for row in range(len(b)):

                for col in range(len(b[row])):

                        b[row][col] = blank
```

```
        # Place all black pieces
        for i in range(1, 13):
                coord = to_index(i)
                board[coord[0]][coord[1]] = black

        # Place all red pieces
        for i in range(21, 33):
                coord = to_index(i)
                board[coord[0]][coord[1]] = red


def print_board(b: list[list]):
        # prints formatted board to console
        # loop through the whole board
        for row in b:
                for col in row:
                        print(col, end="")
                print()


def to_index(n: int) -> tuple[int, int]:
        # Converts a single int to an index in the board
        return coords[n - 1]
# Main game loop
reset(board)
while True:
        print_board(board)
```

```
try:
        origin = to_index(int(input("Input a number between 1 and 32: ")))
except ValueError:
        break

# Can not move empty
while board[origin[0]][origin[1]] == blank:
print("Empty square!")

try:
        origin = to_index(int(input("Input a number between 1 and 32: ")))
except ValueError:
        break

try:
        new = to_index(int(input("Input a number between 1 and 32: ")))
except ValueError:
        break

board[new[0]][new[1]] = board[origin[0]][origin[1]]
board[origin[0]][origin[1]] = blank
```