Lab 1
Vacuum Cleaner

```python
import random

def clean_room(rooms, position):
    if rooms[position] == 1:
        print(f"Room {position+1} is Dirty. Cleaning...")
        rooms[position] = 0
        print(f"Room {position+1} is now Clean.")
    else:
        print(f"Room {position+1} is already Clean.")

def move(position, total_rooms):
    position = (position + 1) % total_rooms
    print(f"Moving to Room {position+1}")
    return position

def run(rooms, steps):
    position = 0
    for _ in range(steps):
        clean_room(rooms, position)
        position = move(position, len(rooms))
    print("Final Room Status:", rooms)
rooms = [random.choice([0, 1]) for _ in range(4)]
print("Initial Room Status:", rooms)

run(rooms, 8)
```

```
    print("Final Room Status:", rooms)
rooms = [random.choice([0, 1]) for _ in range(4)]
print("Initial Room Status:", rooms)

run(rooms, 8)
```

```
Initial Room Status: [1, 1, 1, 0]
Room 1 is Dirty. Cleaning...
Room 1 is now Clean.
Moving to Room 2
Room 2 is Dirty. Cleaning...
Room 2 is now Clean.
Moving to Room 3
Room 3 is Dirty. Cleaning...
Room 3 is now Clean.
Moving to Room 4
Room 4 is already Clean.
Moving to Room 1
Room 1 is already Clean.
Moving to Room 2
Room 2 is already Clean.
Moving to Room 3
Room 3 is already Clean.
Moving to Room 4
Room 4 is already Clean.
Moving to Room 1
Final Room Status: [0, 0, 0, 0]
```

LAB 2
Tic-Tac-Toe

[3]:
```python
def print_board(board):
    print("\n  0   1   2")
    for i in range(3):
        print(f"{i} {board[i][0]} | {board[i][1]} | {board[i][2]}")
        if i < 2:
            print("  -----------")
    print()

def check_winner(board):
    for row in board:
        if row[0] == row[1] == row[2] != '-':
            return row[0]
    for col in range(3):
        if board[0][col] == board[1][col] == board[2][col] != '-':
            return board[0][col]
    if board[0][0] == board[1][1] == board[2][2] != '-':
        return board[0][0]
    if board[0][2] == board[1][1] == board[2][0] != '-':
        return board[0][2]

    return None

def is_board_full(board):
    for row in board:
        if '-' in row:
            return False
    return True

def is_valid_move(board, row, col):
```

```python
def is_valid_move(board, row, col):
    return 0 <= row < 3 and 0 <= col < 3 and board[row][col] == '-'

def get_player_move(player):
    while True:
        try:
            move = input(f"Player {player}, enter your move (row col): ")
            row, col = map(int, move.split())
            return row, col
        except (ValueError, IndexError):
            print("Invalid input! Please enter row and column as two numbers (0-2).")

def play_tic_tac_toe():
    board = [['-' for _ in range(3)] for _ in range(3)]
    current_player = 'X'

    print("Welcome to Tic-Tac-Toe!")
    print("Enter moves as 'row col' (e.g., '1 2')")
    print("Positions are numbered 0, 1, 2")
    while True:
        print_board(board)
        row, col = get_player_move(current_player)
        if is_valid_move(board, row, col):
            board[row][col] = current_player
            winner = check_winner(board)
            if winner:
                print_board(board)
                print(f"🎉 Player {winner} wins!")
```

```
                print_board(board)
                print(f"🎉 Player {winner} wins!")
                break
            if is_board_full(board):
                print_board(board)
                print("It's a tie!")
                break
            current_player = 'O' if current_player == 'X' else 'X'
        else:
            print("Invalid move! That position is taken or out of bounds.")

def main():
    while True:
        play_tic_tac_toe()
        play_again = input("\nDo you want to play again? (y/n): ").lower()
        if play_again != 'y':
            print("Thanks for playing!")
            break
if __name__ == "__main__":
    main()
```

```
Welcome to Tic-Tac-Toe!
Enter moves as 'row col' (e.g., '1 2')
Positions are numbered 0, 1, 2

  0   1   2
0 - | - | -
  -----------
1 - | - | -
  -----------
2 - | - | -

Player X, enter your move (row col):  0 1

  0   1   2
0 - | X | -
  -----------
1 - | - | -
  -----------
2 - | - | -

Player O, enter your move (row col):  0 0

  0   1   2
0 O | X | -
  -----------
1 - | - | -
  -----------
2 - | - | -

Player X, enter your move (row col):  1 1

  0   1   2
0 O | X | -
  -----------
1 - | X | -
  -----------
2 - | - | -

Player O, enter your move (row col):  0 2

  0   1   2
0 O | X | O
  -----------
1 - | X | -
  -----------
2 - | - | -

Player X, enter your move (row col):  2 1

  0   1   2
0 O | X | O
  -----------
1 - | X | -
  -----------
2 - | X | -

🎉 Player X wins!
```