Github link: https://github.com/Striletchi-Vlad/FLCD-L1
BNF Specification for the input file:

<automaton-file> ::= <states-section> <alphabet-section> <transitions-section> <initial-state-section> <final-states-section>

<states-section> ::= states <state-list>
<state-list> ::= <state> | <state> <state-list>
<state> ::= <identifier>

<alphabet-section> ::= alphabet <symbol-list>
<symbol-list> ::= <symbol> | <symbol> <symbol-list>
<symbol> ::= <character>

<transitions-section> ::= transitions <transition-list>
<transition-list> ::= <transition> | <transition> <transition-list>
<transition> ::= <state>,<symbol>,<state>

<initial-state-section> ::= initial_state <state>

<final-states-section> ::= final_states <state-list>

<identifier> ::= <letter_starting> <seq_digits>

Class: FiniteAutomaton

Description:
A class representing a finite automaton with methods for reading its structure from a file, displaying its elements, and verifying if a given sequence is accepted by the automaton.

Attributes:
- states: A set containing the states of the finite automaton.
- alphabet: A set containing the symbols in the alphabet of the finite automaton.
- transitions: A dictionary mapping (state, symbol) pairs to the resulting state after a transition.
- initial_state: The initial state of the finite automaton.
- final_states: A set containing the final states of the finite automaton.

Methods:
1. __init__(self):
   - Description: Initializes the FiniteAutomaton object with empty sets for states, alphabet, and final_states, an empty dictionary for transitions, and None for the initial_state.

2. read_from_file(self, filename: str):
   - Description: Reads the structure of the finite automaton from a given file.
   - Parameters:
     - filename (str): The name of the file containing the finite automaton information.

3. display_elements(self):
   - Description: Displays the elements of the finite automaton, including states, alphabet, transitions, initial_state, and final_states.

4. is_accepted(self, sequence: str) -> bool:

- Description: Verifies whether a given sequence is accepted by the finite automaton.
- Parameters:
  - sequence (str): The input sequence to be verified.
- Returns:
  - bool: True if the sequence is accepted, False otherwise.

Usage (Example):

```
if __name__ == "__main__":
    # Create a FiniteAutomaton instance
    fa = FiniteAutomaton()

    # Specify the filename for the automaton structure
    filename = "fa.txt"  # Replace with the actual filename

    # Read the automaton structure from the file
    fa.read_from_file(filename)

    while True:
        # Display menu options
        print("\n------ Menu ------")
        print("1. Display Elements")
        print("2. Verify Sequence")
        print("3. Exit")
        print()

        # Get user choice
        choice = input("Enter your choice (1/2/3): ")

        # Process user choice
        if choice == "1":
            fa.display_elements()
        elif choice == "2":
            sequence = input("Enter the sequence to verify: ")
            if fa.is_accepted(sequence):
                print("Sequence is accepted by the FA.")
            else:
                print("Sequence is not accepted by the FA.")
        elif choice == "3":
            print("Exiting the program.")
            break
        else:
            print("Invalid choice. Please enter 1, 2, or 3.")
```