https://github.com/AlexaAndreas99/FLCD

FA.in structure:

line 1(Q): the set of states
line 2(EPS): the alphabet(Integers numbers from 0-9)
line 3(Q0): initial state
line 4(Qf): final states
line 5->n(delta): transitions

Every character in FA.in should be separated by an empty space
Every line in FA.in is stored in the FiniteAutomata class, and the structure of the transitions is as follows: a dictionary of dictionaries. For each state create a dictionary which is keyed by the letters of the alphabet and the a global dictionary which is keyed by the states.For example:

FA.in

```
A B C
0 1
A
C
A 1 = B
A 0 = A
B 1 = B
B 0 = C
```

the transitions should look like this:

```
{A:{'1':'B', '0':'A'},
 B:{'1':'B', '0':'C'},
 C:{}}
```

With that structure it is pretty easy to verify if a sequence is accepted by the FA. For every char in the sequence we verify if it is in transition[state] and if not we stop and decline, and if it is we continue to the end and verify if the last state is in the final states.

```
FA.in ::= Q \n eps \n Q0 \n QF \n Trans
<Q> ::= <state>’ ’|<state>
<state> ::= <letter>
<letter> ::= ‘a’ | .. | ‘z’ | ‘A’ | .. | ‘Z’

<EPS> ::= <digit>
<digit> ::= ‘0’ | .. | ‘9’

<Q0> ::= <state>

<Qf> ::= <state> ‘ ’ | <state>

<delta> ::= <Trans> ‘\n’ | <Trans>
<Trans> ::= <state> ‘ ’ <digit> ‘ ’ = ‘ ’ <state>
```