



```
1  # DFA accepting strings ending with 101.
2
3  dfa_data = {
4      "alphabet": {"0", "1"},
5      "input_states": {"A", "B", "C", "D"},
6      "transition_table": {
7          "A": {"0": "A", "1": "B"},
8          "B": {"0": "C", "1": "B"},
9          "C": {"0": "A", "1": "D"},
10         "D": {"0": "C", "1": "B"},
11     },
12     "initial_state": "A",
13     "final_states": {"D"},
14 }
15
16
17 class DeterministicFiniteAutomata:
18     def __init__(self, **kwargs):
19         self.input_states = kwargs.get("input_states")
20         self.alphabet = kwargs.get("alphabet")
21         self.initial_state = kwargs.get("initial_state")
22         self.final_states = kwargs.get("final_states")
23         self.transition_table = kwargs.get("transition_table")
24
25     def print_components(self):
26         print("=" * 24)
27         print("Components:")
28         print("-" * 24)
29         print(f"Q: {self.input_states}")
30         print(f"Σ: {self.alphabet}")
31         print(f"δ: Q × Σ → Q")
32         print(f"q₀: {self.initial_state}")
33         print(f"F: {self.final_states}")
34
35     def print_transition_table(self):
36         print("=" * 24)
37         print("Transition Table:")
38         print("-" * 24)
39
40         # Heading row
41         print(f"{'δ' | '<5}', end='')")
42         for symbol in sorted(self.alphabet):
43             print(f"{symbol:<5}", end='')
44         print()
45         print("-" * (len(self.alphabet) + 1) * 4)
46
47         # Data
48         for state in sorted(self.input_states):
49             print(f"{'{state}' | '<5}', end='')")
50             for symbol in sorted(self.alphabet):
51                 print(f"{self.transition_table[state][symbol]:<5}", end='')
52             print()
53
54     def is_accepted(self, string: str) → bool:
55         current = self.initial_state
56         print(current, end='')
57         for symbol in string:
58             current = self.transition_table[current][symbol]
59             print(f"={symbol}⇒ {current}", end='')
60         print()
61         return current in self.final_states
62
63
64 dfa = DeterministicFiniteAutomata(**dfa_data)
65
66 dfa.print_components()
67 dfa.print_transition_table()
68
69
70 if __name__ == "__main__":
71     while True:
72         string = input(f"Enter a string: ")
73         if string.lower() == "q":
74             break
75         if dfa.is_accepted(string):
76             print(f"{string} is accepted")
77             continue
78         print(f"{string} is rejected")
79
```



```
> python practical-2.py
```

```
=====
```

Components:

```
-----
```

$Q: \{'C', 'D', 'B', 'A'\}$

$\Sigma: \{'1', '0'\}$

$\delta: Q \times \Sigma \rightarrow Q$

$q_0: A$

$F: \{'D'\}$

```
=====
```

Transition Table:

```
-----
```

δ		0	1
----------	--	---	---

```
-----
```

A		A	B
---	--	---	---

B		C	B
---	--	---	---

C		A	D
---	--	---	---

D		C	B
---	--	---	---

Enter a string: 10111

$A \xRightarrow{1} B \xRightarrow{0} C \xRightarrow{1} D \xRightarrow{1} B \xRightarrow{1} B$

10111 is rejected

Enter a string: 11101

$A \xRightarrow{1} B \xRightarrow{1} B \xRightarrow{1} B \xRightarrow{0} C \xRightarrow{1} D$

11101 is accepted

Enter a string: q