



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



```
> python practical-3.py
```

```
=====
```

Components:

```
-----
```

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

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

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

$q_0: A$

$F: \{ 'A' \}$

```
=====
```

Transition Table:

```
-----
```

$\delta$		0	1
----------	--	---	---

```
-----
```

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

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

Enter a number: 21

$A \Rightarrow 1 \Rightarrow B \Rightarrow 0 \Rightarrow A \Rightarrow 1 \Rightarrow B \Rightarrow 0 \Rightarrow A \Rightarrow 1 \Rightarrow B$

10101 is rejected

Enter a number: 10

$A \Rightarrow 1 \Rightarrow B \Rightarrow 0 \Rightarrow A \Rightarrow 1 \Rightarrow B \Rightarrow 0 \Rightarrow A$

1010 is accepted

Enter a number: q