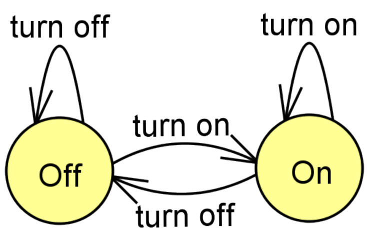
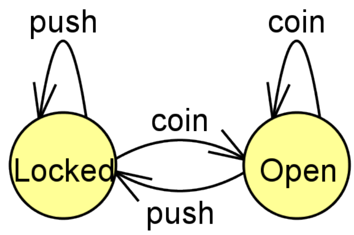
DETERMINISTIC FINITE AUTOMATA (DFA)

Exercise 1

Create an automaton that models a light-switch (alternatively, a turnstile). Give the transition table for it.



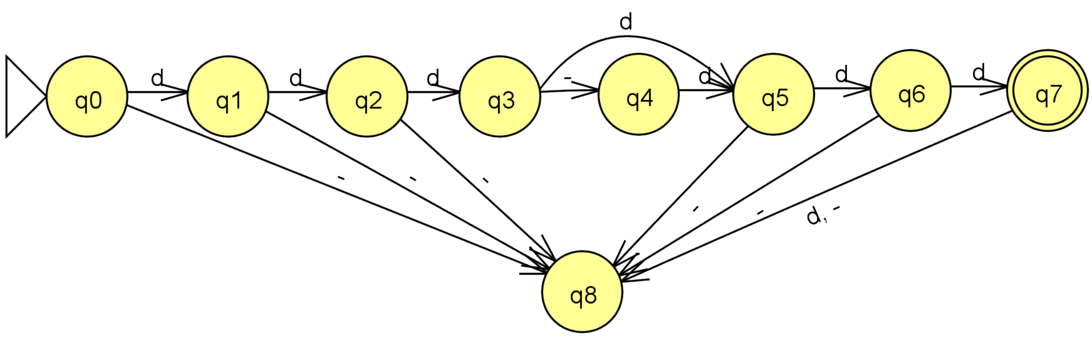


	turn off	turn on
Off	t.off	t.on
On	t.off	t.on

Exercise 2

Create an automaton that is able to recognize phone numbers. The proper format for a phone number is either 6 digits, or 3 digits + a dash + 3 digits.

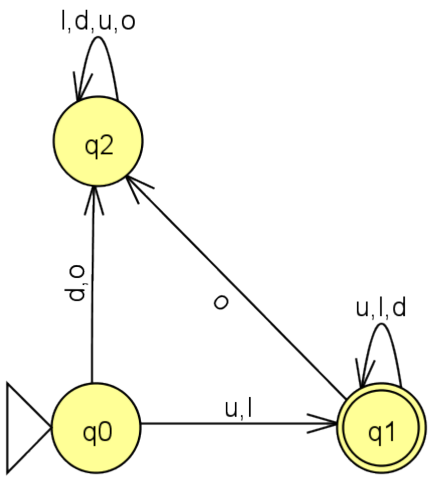
The alphabet for this problem is $\Sigma = \{0, ..., 9, -\}$. However, as the exact type of the digits is not important in this case, we will consider them uniformly as d, and use the alphabet $\Sigma = \{d, -\}$ instead.
The language for our automaton is $L = \{dddddd, ddd - ddd\}$. Any other string over Σ^* should be rejected.



Exercise 3

Create an automaton, that recognizes if a given string is a legal C++ identifier, or not. Such an identifier can contain letters, digits and underscores, but cannot start with a digit. Any other characters are forbidden. Also provide a transition table for the problem.

The alphabet for this problem is $\Sigma = \{l, d, u, o\}$ (standing for letter, digit, underscore, other). Strings containing o or starting with d should be rejected.



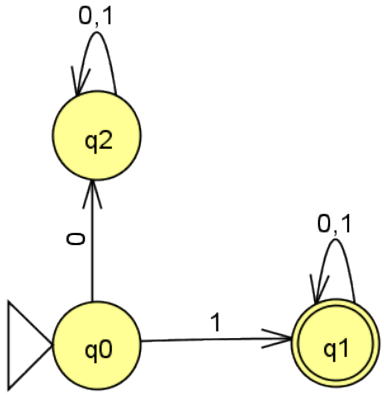
	l	d	u	o
$\rightarrow q_0$	q_1	q_2	q_1	q_2
$*q_1$	q_1	q_1	q_1	q_2
q_2	q_2	q_2	q_2	q_2

(a) How would you modify the above automaton, if strings containing double sequential underscores were not legal identifiers?

(b) How would you modify the above automaton, if the string containing only underscores was not a legal identifier?

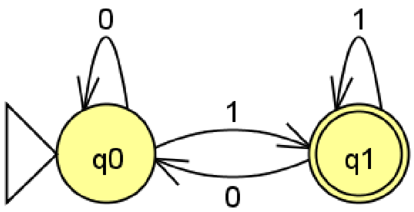
Exercise 4

Create an automaton that accepts all strings starting with 1 over alphabet $\Sigma = \{0, 1\}$.



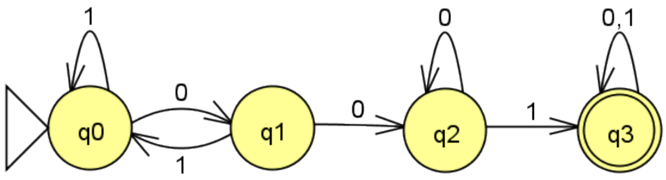
Exercise 5

Create an automaton that accepts all strings ending with 1 over alphabet $\Sigma = \{0, 1\}$.



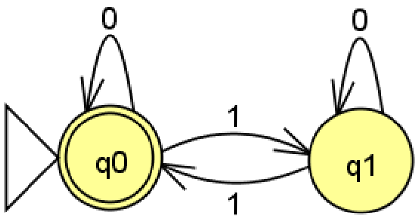
Exercise 6

Create an automaton that accepts all strings containing the substring 001 over alphabet $\Sigma = \{0, 1\}$.



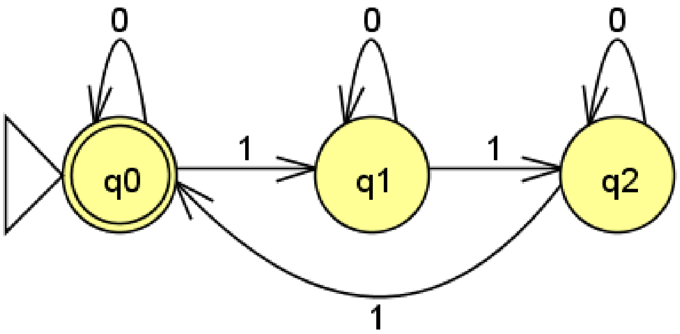
Exercise 7

Create an automaton that accepts all strings with an even number of 1s over alphabet $\Sigma = \{0, 1\}$.



Exercise 8

Create an automaton that accepts all strings over alphabet $\Sigma = \{0, 1\}$ where the total number of ones can be divided by 3.



Exercise 9

Let L be the set of strings over alphabet $\Sigma = \{a, b\}$ that all start with aba. Create an automaton M that accepts L.

Exercise 10

Let language $L = \{b^n ab^m | n, m \geq 0\}$ and alphabet $\Sigma = \{a, b\}$. Create an automaton M that accepts L ($L = L(M)$).

Exercise 11

Let language $L = \{b^n ab^m | n, m \geq 1\}$ and alphabet $\Sigma = \{a, b\}$. Create an automaton M that accepts L ($L = L(M)$).

Exercise 12

Let language $L = \{a^{2n} b^{2m+1} | n, m \geq 0\}$ and alphabet $\Sigma = \{a, b\}$. Create an automaton M that accepts L ($L = L(M)$).

Exercise 13

Let language $L = \{a^n b a^m | n, m \geq 0, (n+m)\%2 = 0\}$ **and alphabet** $\Sigma = \{a, b\}$**. Create an automaton** M **that accepts** L **(**$L = L(M)$**).**