

1. Multiple Choice:

Consider a state machine with:

- inputs: 0, 1, 2
- states: 0, 1, 2, 3
- outputs: 0, 1, 2, 3
- initial state: 0
- transition function:

	input	input	input
	0	1	2
old state: 0	1	3	0
old state: 1	2	0	0
old state: 2	3	1	0
old state: 3	0	2	0

- output function: same as transition function

It maybe helpful for you to draw a state diagram of this machine, to visualize its operation.

What is the best description of this machine?

- it counts forward and backward mod 4, and has a reset input.
- it counts how many more 1's than 0's have been input, and has a reset input.

2. Fill in Multiple Blanks:

Consider a state machine with:

- inputs: 0, 1, 2
- states: 0, 1, 2, 3
- outputs: 0, 1, 2, 3
- initial state: 0
- transition function:

	input	input	input
	0	1	2
old state: 0	1	3	0
old state: 1	2	0	0
old state: 2	3	1	0
old state: 3	0	2	0

- output function: same as transition function

It maybe helpful for you to draw a state diagram of this machine, to visualize its operation.

If you feed this string of inputs into the machine, what would the string of outputs be? Enter the output produced by the indicated input in the sequence below.

input 0, output:	[1]
input 0, output:	[2]
input 2, output:	[3]
input 0, output:	[4]
input 0, output:	[5]
input 0, output:	[6]
input 1, output:	[7]
input 1, output:	[8]
input 1,output:	[9]

3. Fill in Multiple Blanks:

Consider a state machine with:

- inputs: 0, 1, 2
- states: 0, 1, 2, 3
- outputs: 0, 1, 2, 3
- initial state: 0
- transition function:

	input	input	input
	0	1	2
old state: 0	1	3	0
old state: 1	2	0	0
old state: 2	3	1	0
old state: 3	0	2	0

- output function: same as transition function

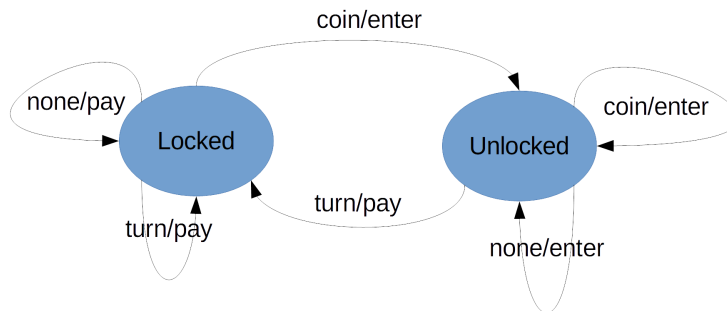
It maybe helpful for you to draw a state diagram of this machine, to visualize its operation.

What string of inputs could you feed in, in order to get this string of outputs? Enter the input that would produce the indicated output in the sequence below.

input: [1]	output: 3
input: [2]	output: 2
input: [3]	output: 0
input: [4]	output: 1
input: [5]	output: 2
input: [6]	output: 3

4. Fill in Multiple Blanks:

Here is a state transition diagram for a turnstile.



- it has states 'locked' and 'unlocked'
- it has inputs 'coin', 'none', and 'turn'
- it has outputs 'enter' and 'pay'

The idea that if it is locked and someone puts in a coin, then it becomes unlocked and turns on a sign that says 'enter'. If it is unlocked and someone turns the turnstile, then it becomes locked and turns on a sign that says 'pay'.

We can describe this machine as an instance of the SM class, as follows:

```
class Turnstile(sm.SM):
    start_state = 'locked'
    def get_next_values(self, state, inp):
        if state == 'locked':
            if inp == 'coin':
                return (Q1, Q2)
            else:
                return (Q3, Q4)
        else:
            if inp == 'turn':
                return (Q5, Q6)
            else:
                return (Q7, Q8)
```

1. What string should Q1 be? [1]
2. What string should Q2 be? [2]
3. What string should Q3 be? [3]
4. What string should Q4 be? [4]
5. What string should Q5 be? [5]
6. What string should Q6 be? [6]
7. What string should Q7 be? [7]
8. What string should Q8 be? [8]