## Finite State Machine (FSM) Challenge

A Finite State Machine (FSM) is a mathematical model of a computation algorithm. It is made of a finite (fixed) number of states, usually represented as circles, and transitions, usually represented as arrows. The machine is in only one state at a time, and the state changes based on some input, based on the transitions.

It's probably best to see an example. Here is a state diagram of an FSM whose inputs are a sequence of 0s and 1s. It figures out whether that sequence of zeros and ones contain a subsequence of three 1s in a row:

The FSM "reads" a sequence of 1s and 0s. The FSM starts at State 1 (S1). If the first input is a 0, it "transitions" back to S1. If the first input is a 1, it transitions to S2. Then it reads



the next input and makes a similar transition. If, at the end of the sequence, the FSM is on S4, it has "accepted" the input and we know that there is a sequence of three 1s somewhere in the input.

For example, let's try it with a sequence of 101110. Note that the FSM is assumed to start in State 1 (S1).



Since that is the end of the input, and the FSM is in S4, the FSM has accepted the input and we know there are three 1s in a row.

Consider what would happen if the input was 01101: The FSM would start on S1, move to S2, then move to S3, then back to S1, then to S2. Since it didn't end on S4, it did not accept the input and we know that there aren't three 1s in a row.

When you represent an FSM using circles and arrows, it is called a *state diagram*. You can also describe a FSM using a *state and transition table*. Here is the state and transition table for the FSM we just described:

Current State	Input	Next State
51	0	51
	1	52
52	0	51
	1	53
53	0	54
	1	51
S4 (Accepts)	0 or 1	54

For each challenge, you must make a READABLE FSM state diagram, and show that you tested the FSM with a few different inputs by showing the input and the states that result. For example, if the problem was to make a FSM that recognized a sequence of three 1s, you would draw the state diagram and show your inputs and states as follows:

Input string	Accept or reject?	States
0101100	Reject	51 (initial) 51 52 51 52 52 51 51
1110110	Accept	51 (initial) 52 53 54 54 54 54 54
0110	Reject	51 (initial) 51 52 53 51

CHALLENGES	(it	helps	to	think	about	what	each	state	means	when	you	design	your	FSN	۱)
------------	-----	-------	----	-------	-------	------	------	-------	-------	------	-----	--------	------	-----	----

Challenge	Input characters	Accept if	PTS
Any 1?	1s and Os	Any of the input is a 1	1 2
All 1s	1s and Os	All of the input is 1	1 2
Even 1s?	1s and Os	There are an even number of 1s.	1 2
Even 1s and 0s	1s and Os	There are an even number of 1s and 0s	1
One three ones	1s and Os	Exactly one sequence of three 1s; no more than three 1s. In other words,	1 <del>1</del>
		0 1 1 1 1 would fail, as would	
		0 1 1 1 0 1 1 1 (make sure to test with those inputs).	
Remainder of 3	1s and Os	The remainder that you get when you divide the number of 0s by 3 is equal to the remainder you get when you divide the number of 1s by 3.	1

Extra challenge: It is impossible to make a FSM that can tell if the number of 1s is equal to the number of 0s. Explain why it is impossible. (1 pt)