Notes for the Feb 3:
Finite Automata: Simplest Nontrivial Example.

At end, the machine reads an "silent" stop character, blank or unix $\$.

Keep focus on checking the integrity of the 9-bit parity checked byte.

Start is both the initial state and the desired final state.
A final state is also called an accepting state, others are rejecting.

9-bit byte: 011010100 $\in \{L, 011110100 \mid L$.
$L = \{8$-bit strings having even parity \} (DFA)

A (deterministic) finite automaton is a 5-tuple $M = (\mathbb{Q}, \Sigma, \delta, s, F)$ where:

$\mathbb{Q}$ is a finite set of states
$\Sigma$ is a finite input alphabet
$s$ is a member of $\mathbb{Q}$, is the start state
$F$, a subset of $\mathbb{Q}$, is the set of accepting states
$\delta$, the transition function, is a function from $\mathbb{Q} \times \Sigma \rightarrow \mathbb{Q}$

$\delta$ can be written:

$\delta = \{(5, 0/5), (8, 0/9), (5, 1/5), (9, 1/9)\}$
A DFA is a 5-tuple \( M = (Q, \Sigma, S, s, F) \) where

```java
class DFA {
    set(State) Q;
    set(Char) Sigma;
    State s;
    set(State) F;

    State delta(State p, char c);
    set(triple(State, char, State)) >
    delta;

    state execDelta(State p, char c); // return state
}

DFA(Sigma, Q, S, s, F) {
    empty, spear, dead
    set(Char)
    set(Sigma)
    set(State)
}
```

\[ \delta = \{(empty, 0, empty), (empty, 0, spear), (empty, D, dead), (spear, 0, spear), (spear, 1, spear), (spear, D, emph), (dead, 0, dead), (dead, 1, dead), (dead, D, dead)\} \]

<table>
<thead>
<tr>
<th>Q</th>
<th>0</th>
<th>1</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty</td>
<td>empty</td>
<td>spear</td>
<td>dead</td>
</tr>
<tr>
<td>spear</td>
<td>spear</td>
<td>spear</td>
<td>emph</td>
</tr>
<tr>
<td>dead</td>
<td>dead</td>
<td>dead</td>
<td>dead</td>
</tr>
</tbody>
</table>