Let $L$ be an arbitrary regular language over the alphabet $\Sigma = \{0,1\}$. Prove that the following languages are also regular. (You probably won't get to all of these during the lab session.)

1. $\text{FlipOdds}(L) := \{\text{flipOdds}(w) \mid w \in L\}$, where the function $\text{flipOdds}$ inverts every odd-indexed bit in $w$. For example:

   \[
   \text{flipOdds}(0001111010100) = 10100111111110
   \]

**Solution:** Let $M = (Q,s,A,\delta)$ be an arbitrary DFA that accepts $L$. We construct a new DFA $M' = (Q',s',A',\delta')$ that accepts $\text{FlipOdds}(L)$ as follows.

Intuitively, $M'$ receives some string $\text{flipOdds}(w)$ as input, restores every other bit to obtain $w$, and simulates $M$ on the restored string $w$.

Each state $(q,\text{flip})$ of $M'$ indicates that $M$ is in state $q$, and we need to flip the next input bit if $\text{flip} = \text{TRUE}$.

\[
\begin{align*}
Q' &= Q \times \{\text{TRUE},\text{FALSE}\} \\
 s' &= (s,\text{TRUE}) \\
 A' &= \\
\delta'((q,\text{flip}),a) &=
\end{align*}
\]
2. **UNFLIPODD1S(L)** := \{w ∈ Σ* | flipOdd1s(w) ∈ L\}, where the function **flipOdd1** inverts every other 1 bit of its input string, starting with the first 1. For example:

\[
\text{flipOdd1s}(00001110010101) = 00000101000100
\]

**Solution:** Let \(M = (Q,s,A,δ)\) be an arbitrary DFA that accepts \(L\). We construct a new DFA \(M' = (Q',s',A',δ')\) that accepts **UNFLIPODD1S(L)** as follows.

Intuitively, \(M'\) receives some string \(w\) as input, flips every other 1 bit, and then simulates \(M\) on the transformed string.

Each state \((q,\text{flip})\) of \(M'\) indicates that \(M\) is in state \(q\), and we need to flip the next 1 bit if and only if \(\text{flip} = \text{True}\).

\[
Q' = Q \times \{\text{True}, \text{False}\}
\]
\[
s' = (s, \text{True})
\]
\[
A' = \quad
\]
\[
δ'((q,\text{flip}), a) = \quad
\]
3. \( \text{FlipOdd}_1s(L) := \{ \text{flipOdd}_1s(w) \mid w \in L \} \), where the function \( \text{flipOdd}_1 \) is defined as in the previous problem.

Solution: Let \( M = (Q, s, A, \delta) \) be an arbitrary DFA that accepts \( L \). We construct a new NFA \( M' = (Q', s', A', \delta') \) that accepts \( \text{FlipOdd}_1s(L) \) as follows.

Intuitively, \( M' \) receives some string \( \text{flipOdd}_1s(w) \) as input, guesses which 0 bits to restore to 1s, and simulates \( M \) on the restored string \( w \). No string in \( \text{FlipOdd}_1s(L) \) has two 1s in a row, so if \( M' \) ever sees 11, it must reject.

Each state \((q, \text{flip})\) of \( M' \) indicates that \( M \) is in state \( q \), and we need to flip some 0 bit before the next 1 bit if \( \text{flip} = \text{True} \).

\[
Q' = Q \times \{ \text{True}, \text{False} \} \\
s' = (s, \text{True}) \\
A' = \\
\delta'((q, \text{flip}), a) =
\]
4. Faro($L$) := \{faro($w, x$) $|$ $w, x \in L$ and $|w| = |x|$\}, where the function faro is defined recursively as follows:

\[
\text{faro}(w, x) := \begin{cases} 
  x & \text{if } w = \epsilon \\
  a \cdot \text{faro}(x, y) & \text{if } w = ay \text{ for some } a \in \Sigma \text{ and some } y \in \Sigma^*
\end{cases}
\]

For example, faro(0001101, 1111001) = 0101011100011. (A “faro shuffle” splits a deck of cards into two equal piles and then perfectly interleaves the two piles.)

\begin{quote}
\textbf{Solution: } Let $M = (Q, s, A, \delta)$ be an arbitrary DFA that accepts $L$. We construct a new DFA $M' = (Q', s', A', \delta')$ that accepts Faro($L$) as follows.

Intuitively, $M'$ reads the string $\text{faro}(w, x)$ as input, splits the string into the subsequences $w$ and $x$, and passes those strings to independent copies of $M$. Let $M_1$ denote the copy that processes the first string $w$, and let $M_2$ denote the copy that processes the second string $x$.

Each state $(q_1, q_2, \text{next})$ indicates that machine $M_1$ is in state $q_1$, machine $M_2$ is in state $q_2$, and $\text{next}$ indicates whether $M_1$ or $M_2$ receives the next input bit.

\[
\begin{align*}
Q' &= Q \times Q \times \{1, 2\} \\
s' &= (s, s, 1) \\
A' &= \\
\delta'((q_1, q_2, \text{next}), a) &= 
\end{align*}
\]
\end{quote}