This assignment consists of two parts: a written homework and a programming portion.

**Part 1: Written homework**

1. Sipser 1.27: Read the informal description of the finite state transducer in exercise 1.24. Draw the state diagram of the FST on the alphabet $\Sigma = \{a, b\}$ which flips the symbols of the input for every odd position, but leaves the even positions unchanged. For example, on input $aabb$ the output should be $abba$.

2. Sipser 1.31: For any string $w = w_1w_2\cdots w_n$, the reverse of $w$, $w^R$ is the string $w$ in reverse order, e.g., $w^R = w_n\cdots w_2w_1$. For a language $L$, define $L^R = \{w^R|w \in L\}$. Show that if $L$ is a regular language, then $L^R$ is also regular. This implies regular languages are closed under reversal.

3. Sipser 1.41: For languages $A$ and $B$, let the perfect shuffle of $A$ and $B$ be the language $\{w|w = a_1b_1\cdots a_kb_k$, where $a_1\cdots a_k \in A$ and $b_1\cdots b_k \in B$, and each $a_i, b_i \in \Sigma\}$. Show that regular languages are closed under perfect shuffles.

4. Prove or disprove the following “identities” regarding regular expressions.

   (a) $(R^*S^*)^* = (R \cup S)^*$
   (b) $(R \cup S)^* = R^* \cup S^*$
   (c) $(R \cup S)^*S = (R^*S)^*$

**Part 2: Programming Exercise**

Implement a generic NFA simulator in the programming language of your choice. I recommend python and the examples here will use python. Your code should accept two files as input; a machine description, and a list of input strings to test. A full writeup appears on the course website.