Solve problems 1 through 4. Problem 5 is optional.

Problem 1: Regular Expressions (10 points)
Write down regular expressions for each of the following languages. Assume that $\Sigma = \{0, 1\}$.

(a) The language containing strings where 0s and 1s alternate. (3 points)

(b) The language containing strings in which the number of 1s is divisible by 4. (3 points)

(c) The language containing strings in which the number of 0s between every pair of consecutive 1s is even. (4 points)

Problem 2: Constructing a DFA (10 points)
Consider the following language.

$$L = \{w \in \{0,1\}^* : w = 0^m 1^n \text{ where } m \text{ and } n \text{ are either both even or both odd}\}$$

(a) Write down the strings $w \in L$ such that the length of w is six. (2 points)

(b) Consider the following pair of languages.

$$L_1 = \{w \in \{0,1\}^* : w = 0^m 1^n \text{ where } m \text{ and } n \text{ are both even}\},$$

$$L_2 = \{w \in \{0,1\}^* : w = 0^m 1^n \text{ where } m \text{ and } n \text{ are both odd}\}.$$

Notice that $L = L_1 \cup L_2$. So, one way of designing a DFA for L would be to construct DFA for L_1 and L_2 and combine them using the "cross-product" construction shown in class.

Construct a DFA for L_1. (5 points)

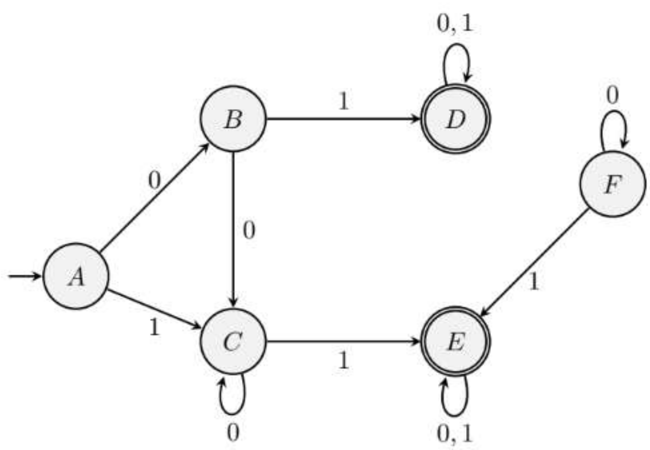
(c) If you were to construct a DFA for L using the method described in (b), how many states would it have? Your answer should only be a number. (1 point)

(d) However, there is a DFA for L using at most seven states. Find that DFA. (2 points)

Problem 3 : Thompson's Construction Method (10 points)
Convert the following regular expression into an ϵ-NFA using Thompson's construction method.

$$\mathtt{ab^*c} + \left((\mathtt{ab} + \mathtt{bc})^* + (\mathtt{a} + \mathtt{b})^+\right)\mathtt{bc} + \mathtt{a}(\mathtt{bc})^*$$

Problem 4: Applying Hopcroft's Algorithm (10 points)



(a) Apply Hopcroft's Algorithm on the DFA above and show step-by-step simulation. (7 points)

(b) Draw the new minimized DFA that you got from (a). (3 points)

Problem 5: (Bonus) Another DFA (5 points)

(Note that this is a bonus problem. Attempt it only after you are done with everything else. Even if you do not attempt it, you can get a perfect score. So, do not worry if you find it too hard!)

In a string over $\{0, 1\}$, call a 0 *special* if it is not immediately preceded by a 1. For example, 100100 contains two special zeros. The first one is in position three while the second one is in position six. The 0s in positions two and five are not special since they are immediately preceded by a 1.

(a) How many special 0s does 01000 contain? (0 points)

(b) Let $L = \{w \in \{0, 1\}^* : w \text{ contains an even number of special 0s}\}$. Draw a DFA with four states that recognizes L. (5 points)