Problem 0. Document how much time you spend on each of the following problems and cite any resources you received help from.

Problem 1. Prove that the following languages are regular by giving an example of a DFA or an NFA that recognizes them along with a short argument that your machine recognizes the given language.

(a) $L_1 = \{x0y \mid x, y \in \{0,1\}^*\}$,

(b) $L_2 = \{w11 \mid w \in \{0,1\}^*\}$,

(c) $L_3 = L_1 \cap L_2$, where L_1 and L_2 are defined above.

Problem 2. Let $\Sigma = \{\texttt{a},\texttt{b}\} \cup \{\texttt{N},\texttt{/},\texttt{*}\}$ be the alphabet for this problem where `N` represents the *newline* character.

In Java, there are three different types of comments:

1. `/*...*/`
2. `//...N`
3. `/**...*/`

The third type is commonly called a *Javadoc* comment.

(a) Construct an NFA that recognizes the language of all such comments. Note that the entire input string needs to be a comment, therefore your NFA should accept strings like `//abbaN` and reject strings like `ab//ababN` and even `//aaab` (since the terminating newline is missing). Note that `/*aba/*aaa*/` is a valid string of the first kind, but `/*aba*/aaa*/` is NOT (because the first `*/` terminates the comment).

(b) For syntax highlighting purposes, IDEs often need to distinguish between normal comments and Javadoc comments.

Construct an NFA that only recognizes strings of the first and second kind (i.e. it rejects Javadoc comments).

Note that `/**/` is a valid comment of the first kind but `/***/` is a comment of the third kind.

Problem 3. Let $\Sigma = \{\texttt{0},\texttt{1},\texttt{2},\texttt{3},\texttt{4},\texttt{5},\texttt{6},\texttt{7},\texttt{8},\texttt{9}\} \cup \{+, \times, -, \div\}$ be an alphabet.

A string $w \in \Sigma^*$ is called a *simple numeric expression* if it is in one of the following two forms.

1. w consists of one or more digits with no leading zeros.
2. $w = w_1 \otimes w_2$ where w_1 and w_2 are also simple numeric expressions and $\otimes \in \{+, \times, -, \div\}$.

Therefore `1074` is a simple numeric expression of the first kind; `50 + 7` is a simple numeric expression of the second kind; and `2 − 700 × 8 ÷ 10` is also a simple numeric expression because of its recursive definition.

Let $S \subseteq \Sigma^*$ be the set of all such simple numeric expressions. We also note that `0` $\in S$ because `0` is a digit without any *leading* zeros.

(a) Give a DFA or an NFA that recognizes S.

(b) Give a regular expression that recognizes S.

Bonus Problem (Extra Credit)**.** In class, we discussed how to encode the relation $x + y = z$ into a language. Here we will show an alternative approach of encoding this relation. Consider the alphabet defined by:

$$\Sigma_3 = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}.$$

Note that the alphabet Σ_3 contains all the size 3 collumns of 0s and 1s. Therefore a string in Σ_3^* yields three rows of 0s and 1s. Consider each of these three rows to be a binary number and let

$$\text{ADD}_2 = \{w \in \Sigma_3^* \mid \text{the bottom row of } w \text{ is the sum of the top two rows}\}.$$

For example, the following string is in the language

$$\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} \in \text{ADD}_2,$$

since $001 + 011 = 100$. However, the following string is not

$$\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix} \notin \text{ADD}_2,$$

since $01 + 00 \neq 11$.

Prove that ADD_2 is regular by giving a DFA, NFA, or regular expression that recognizes it. You must also include an argument of why it recognizes the appropriate language.