Exercise 1.

- Consider the following regular expression : $(a|b)^*ab^*$
  - **a)** Construct a nondeterministic finite automaton (NFA) for the above regular expression using Thompson's construction.
  - **b)** Convert this nondeterministic finite automaton into a deterministic finite automaton (DFA).
  - **c)** Minimize the number of states in the resulting automaton.

Exercise 2.

- The lexical entities of a mini programming language are as follows:

Keywords	begin , end , if , then , else
Identifiers	Strings consisting of a single letter, or a letter followed by several letters or digits
Constants	Strings consisting of a single digit, or several digits
Operators	< , <= , = , < > , > , >= , + , -

  - **a)** Provide the regular expressions that describe these lexical entities.
  - **b)** Construct a finite-state automaton for these expressions.

Exercise 3.

The Thompson construction rules transform a regular expression $\mathbf{R_1}$ into a nondeterministic finite automaton $\mathbf{N_1}$. Propose analogous construction rules for nondeterministic finite automata for the following operators:

- $\mathbf{R_1^+}$
- $\mathbf{R_1?}$ (whose meaning is $\mathbf{R_1 \mid \varepsilon}$)

We modify Thompson's construction rule for the expression **R*** by **not** adding a new initial state and a new final state. Instead, we add two ε-labeled transitions: one from the final state of R's automaton back to its initial state, and the other from the initial state to the final state.

Is this modified rule always valid for the expression **R***? In the general case (composition of rules), does the proposed modification affect the validity of the constructions? Give a concise example to justify your answer.