Problem 1 (2 Points)
In the lecture an automaton was defined that recognises binary numbers that have a remainder of 0 when divided by 3.

![Automaton for division by 3 problem.](image)

Define a set of rules that construct exactly those words the automaton recognises. The rules may use types.

Problem 2 (2 Points)
Consider the alphabet \( \{a, b\} \). Construct an automaton that recognises all words that start and end with the same two letters. Note that there is no limit to the number of letters in the middle. For example the words \( abaababab \) and \( aabaa \) are in the language, but these are not: \( abaaba \), \( aababb \).

a) Give a non-deterministic automaton that recognises this language

b) Give a deterministic automaton that recognises the language

Problem 3 (6 Points)
Consider the language of Roman numerals. The goal of this exercise is to give a syntax and semantics for the Roman numerals

a) There are several variations of the exact definition of the Roman numerals. Choose one definition and write down which one you chose (link is sufficient).

b) Give a set of rules that construct exactly the valid Roman numerals.

c) Give a semantics for the Roman numerals (= translate to decimal numbers).
d) Give an intuition (as in write a few sentences) why your set of rules is correct.

*Hint:* It is not unusual to end up with dozens of rules to solve this problem.

*Important:* The presentation of the rules must be clean and understandable (group them in a reasonable way). Together with the intuition it should be easy to understand.

**Bonus problem (3 Bonus Points)**
Give for the Roman numerals an automaton that recognises the language. Give an intuition why your automaton is correct.