Turing machine:

Write a program to simulate a universal Turing machine:

\[ M = (Q, \Sigma, \Gamma, \delta, q_0, F, B) \]

- \( Q \): states
- \( \Sigma \subseteq \Gamma \): input symbols
- \( \Gamma \): tape symbols
- \( \delta = Q \times \Gamma \vdash Q \times \Gamma \times \{Left, Right\} \): transition function
- \( q_0 \in Q \): start state
- \( F \subseteq Q \): set of halt states
- \( B \): blank (whitespace)

With the following additional specifications:

- \( \Sigma = \{0, 1, \ldots, 9, q, L, R\} \) are the input symbols.
- \( \Upsilon \subseteq \Gamma \) such that \( \Upsilon \cap \Sigma = \emptyset \); tape symbols not in input alphabet.
- \( D = \{0, 1\} \): data alphabet - computation will use these symbols.
- \( d = D \cup \Upsilon \): data item.
- \( m = \{L, R\} \): move specification.
- \( S = ddqn^+qn^+m \): a statement to be interpreted by the program reading part of the machine, where \( d \) is a data item, \( qn^+ \) is a state specified as \( q \) followed by an integer, and \( m \) is a move specification. This is interpreted as symbol read/written, present state+result state, move.
- \( P = S^+ \): a program for a Turing machine - that is, a sequence of statements ending with a period.

A universal Turing machine reads a program that simulates any other Turing machine. To simplify the programming task, we will write the program reading function of the machine in some external language, rather than programming the machine itself to read its program.

We will consider \( \Gamma \) to be an extended alphabet for the program read function; the machine(s) we are going to simulate have the input alphabet \( D \cup B \) and the tape alphabet \( D \cup B \cup \Upsilon \). You are free to pick symbols already in \( \Sigma \) and include them in \( \Upsilon \) if you wish. Note that if you use symbols not in \( \Sigma \), then \( \Gamma = \Sigma \cup \Upsilon \). Note also that the Turing machine you are required to simulate has a transition function \( \delta^t = Q \times (D \cup \Upsilon) \vdash Q \times (D \cup \Upsilon) \times \{L, R\} \), \( \delta^t \subseteq \delta \).

Your machine should be able to read a tape of the form \( Pw \) where \( P \) is a program and \( w \) is an input string for the machine being simulated. You are free to specify the number and type of tapes for your machine.
546 requirements:
Demonstrate the ability of your machine to read and execute a program that adds integers, similar to that given in class.

646 requirements:
In addition to 546 requirements, demonstrate the ability to recognize a language to be specified.